International Journal on Advances in Security

2025 vol. 18 nr. 1&2

The *International Journal on Advances in Security* is published by IARIA.
ISSN: 1942-2636
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Security
Volume 18, Number 1 & 2, 2025

Pedro Sousa, University of Minho, Braga, Portugal
Miroslav Velev, Aries Design Automation, USA
Cong-Cong Xing, Nicholls State University, USA

International Journal on Advances in Security
Volume 18, Numbers 1 & 2, 2025

CONTENTS

Automated Social Engineering Tools Revisited - An Extended Overview and Comparison with Respect to Capabilities and Detectability

Dominik Dana
St. Pölten UAS
St. Pölten, Austria
email: is191805@fhstp.ac.at

Timea Pahi
St. Pölten UAS
St. Pölten, Austria
email: timea.pahi@fhstp.ac.at

Sebastian Schrittwieser
CD-Labor AsTra, University of Vienna
Vienna, Austria
email: Sebastian.Schrittwieser@univie.ac.at

Simon Tjoa
St. Pölten UAS
St. Pölten, Austria
email: simon.tjoa@fhstp.ac.at

Peter Kieseberg
St. Pölten UAS
St. Pölten, Austria
email: Peter.Kieseberg@fhstp.ac.at

***Abstract*—The manual effort required by social engineers to obtain information about people and organizations that are in their focus can be extremely high in case of targeted attacks. Attackers, therefore, strive to automate processes as much as possible. With a few menu entries and selections, it is already possible to export email addresses from social media profiles, as well as to send friend requests and phishing messages to a large number of people. In this paper, we analyze the most popular frameworks for modeling Social Engineering attacks and generate a simplified and generalized meta-model. Based on this model, it was analyzed which parts of Social Engineering attacks can be automated using state-of-the-art tools that are readily available. The capabilities of these tools were thoroughly evaluated, including ready-to-use system environments. This work is an extended version of our work conducted presented at ICCGI 2024.**

***Keywords-Automated Social Engineering*; *Social Engineering Frameworks*; *Social Engineering Models*; *Technical Social Engineering*.**

I. Introduction

This paper is an extension of our work [1] published in the *Nineteenth International Multi-Conference on Computing in the Global Information Technology ICCGI 2024* and expands the original text. Major new parts, aside modernization and changes throughout the text, include a more conclusive and comprehensive analysis of related attacker models, the inclusion of OSINT (Open Source INTelligence) Link Lists for searching for user data, as well as the discussion of ready-to-use system environments in the reconnaissance phase. Furthermore, key aspects were updated to the current state of the art and the tool selection was expanded.

Social Engineering (SE) is an emerging threat that has evolved along with networking and social media and has attracted increasing attention in recent years. While fraud existed long before, the widespread use of social media and cyberspace provides fertile ground for traditional fraud, as more and more personal information is shared but little awareness and measures are in place to protect it [2]. Especially the widespread and constantly available Social Networking Sites (SNS), are a playground to carry out various forms of phishing attacks [3]. There are advanced phishing attacks that spread through sharing SNS posts that can lead to information leakage [3], but also targeted attacks, where users working for a specific company are identified and contacted through SNSs and their confidential information is stolen, e.g., via direct messages [4]. Last but not least, habituation effects also lead to various links being clicked, posts being copied, liked, shared and pasted, which ultimately promotes Social Engineering [3]. However, Social Engineering requires a great deal of time spent cultivating relationships, building trust, and then exploiting users to obtain classified information [5]. The tools used for this purpose are, in terms of basic information retrieval, mostly located in the Open Source Intelligence (OSINT) area and rely on a large collection of publicly available information on the Internet about people and organizations. From the social engineers' point of view, the attacks need to be automated, in order to reach many victims and they should behave human-like, so that more victims fall for them [6]. Automation is especially interesting in the reconnaissance phase, as e.g., in the context of an initial information gathering phase, known users would have to be searched for manually for hours on various platforms and social media channels. this task can already be performed by proprietary search engines, across hundreds of platforms, with just a few mouse clicks. It is a similar story with creating phishing messages, or phishing sites. Instead of designing websites yourself that are used for water-holing or phishing attacks, or instead of sending out a high number of phishing messages via email yourself, a few menu selections or clicks in the respective tools are enough.

This paper describes current automation possibilities which can be used for Social Engineering. The structure of this paper, after a brief introduction and analysis of related work in Section II, it is divided into three main sections, where relevant legal and ethical aspects for the work are considered (Section III), a comparative analysis of Social Engineering phase models and frameworks (Section IV), and the application of the Social Engineering tools themselves (Section V) is conducted. Section VI provides a conclusion and suggestions for future work, including answers to these research questions:

- RQ1: To what extent are freely available Social Engineering supporting tools already automated and what does this mean in terms of Social Engineering?
- RQ2: Which phases of Social Engineering can be handled with the tools?
- RQ3: How do the different tools interact with each other, are there tool suites that start and accompany a complete Social Engineering process?
- RQ4: How reliable are the results of the tools?

II. Related Work

In this section, we provide an overview on the most important techniques, tools, advanced attacks, as well as trust factors and alternative frameworks.

A. Techniques and tools

In addition to the literature by Mitnick [2] and Hadnagy [3], publications by Talamantes [6] and Kim [7] were analyzed, in which the first tools from the OSINT domain and the first automated tools, including the Social Engineering Toolkit (SET) and Maltego, were already mentioned. Handnagy additionally describes in [8] the Social Engineering pyramid as another Social Engineering phase model. An important distinction into the attack categories "Computer Based" and "Human Based" within Social Engineering, is made by Wang et al. in [9], similarly in Aldawood and Skinner's work [10]. In their paper, Wang et al. also state that technical attacks are becoming increasingly difficult and therefore Social Engineering attacks are on the rise. Furthermore, they assumed the most important attack media to be e-mail, websites and the telephone. Banire et al. also describe in [11] that these also represent the most common attack methods from which phishing, vishing and smishing attacks result. In [10], it is also concluded that virtual communities, after personal data is often stored in these platforms, are the largest source of Social Engineering attacks, as little technological know-how is needed once trust has been established with the victims (see also the study from Kenya [12]). Other techniques and tools, especially from the OSINT domain and people-search engines, are described in [13]. However, their main area of application extends to the USA, as application within the EU, due to the General Data Protection Regulation (GDPR), is not allowed as the GDPR requires operators of the tools to ask for consent when collecting personal data.

B. Advanced attacks and automation

A definition of automation is simplistically and naively made in [14] as systems that take over the execution of tasks from humans and thereby simply reduce the amount of work, or attention, that humans need to devote to these tasks. Wang et al. state in [15] that the wide adoption and availability of SNSs, the Internet of Things (IoT), industrial Internet, and mobile devices, have created greater attack surfaces for Social Engineering. The reason behind this is that due to huge amounts of data generated by their use and that people in today's world share more information about their own personal identities, activities, relationships, locations, and personal interests, as well as their work and work environments on social media combined with the availability of Social Engineering tools, facilitates large-scale Social Engineering attacks. Automated tools, mentioned by Wang et al. in [15], in addition to ways to bypass phishing and deep learning detection, include the automated chat bots of Huber (ASE bot) [16], Lauinger et al. (Honeybot) [17], amongst others. According to their own statements, compared to the ASE bot, Honeybot moves one step further, by not having humans communicate directly with a bot, but instead initiating a conversation between two real people, with Honeybot acting as a "Bot in the Middle", interposed in between. The behavior of Honeybot by changing, replacing, or deleting parts of messages, is individually controllable and the chance, for example, to click on links, which are inserted, or changed by Honeybot, is greatly increased, compared to other chat bots. The project "Social Network Automated Phishing with Reconnaissance" (SNAP_R) [18] on the other hand, interacts with users on the Twitter platform and sends a machine-generated tweet to its targets, which mostly contains a shortlink. Broken English and shortlinks are accepted on Twitter due to the character limit, which is why the authors see SNAP_R as an extension to SET to automatically distribute phishing messages to a larger target group. The ASE bot, Honeybot and additionally the Koobface bot, spreading as malware through the Facebook social media platform, are also cited as automated Social Engineering tools in a study by Kaul and Sharma [19].

C. Trust factors as the basis for automation functionality

The trust factors that enable Social Engineering to be successful, are described by Kano and Nakajima after an experiment [20]. The fact that people are more likely to open suspicious links in messages from Facebook friends than from, e.g., their bank is also addressed by Stern at Kaspersky [21]. The latter go on to state that it is also widespread to clone unrestricted Facebook profiles and send friend requests to friends of this original profile. The goal is to use the cloned profile to send convincing phishing messages or to get the Facebook friends to click on phishing links.

D. Alternative Frameworks

In addition to the classical frameworks and Social Engineering models, presented in a subsequent section, models such as the one described by Tong Wu et al. in [4], consisting of Social Engineering Sessions (SES) and Social Engineering Dialogues (SED) and the models in [22], which are still in early stages of development represent alternative approaches for new Social Engineering models.

III. Legal and ethical aspects

When compiling and searching for information in the context of Social Engineering, data and information from and about specific individuals are used. This also holds true for the experiments conducted in this study. While malicious attackers will not care about legal or ethical issues regarding private data retrieval, this had, of course, been an issue during our research. Data and information that can be traced back to individuals is considered as personal data in the current version of the General Data Protection Regulation (GDPR), under Article 4 [23], the processing of which is considered to be lawful if there is consent for processing for one or more specific purposes and these are processed appropriately for the purpose and in accordance with the principle of data minimization [23] and appropriate protective measures have also been taken by the processor for the required storage period. Even if information about individuals and institutions can be found freely on the Internet, from an ethical point of view, it cannot and should not be assumed that this information is also freely available for use. However, information can also be interpreted differently in the wrong circumstances, leading to unintended and unfavorable outcomes for the individuals concerned. Another dilemma is that the OSINT sample is minimized or selected depending on the needs of the collector [13]. Thus, important sources might indeed be intentionally neglected in order to achieve a particular result. The handling of legal and ethical aspects is quite different in the related work. This ranges from permissions and questionnaires requested in advance, to simply conducting experiments. Debriefing with participants is rarely held. In order not to unknowingly turn participants into experimental subjects, which has already raised serious ethical concerns [24], own outdated and already known leaked data was searched for first tests with the tools. When processing the data and information found, an attempt was made, despite automation, to take into account the principle of data minimization and purpose limitation as far as possible. Attention was paid to emerging and possibly disadvantageous combinations of the results. The search and test results were not saved after the application of the different tools. In some cases, the tools automatically created log files that contained the results of the search queries. These log files were also deleted at the end of the tests.

New regulations will also result in new ethical and legal requirements, especially when dealing with personal information. Regarding the utilization of automation for Social Engineering this is especially important, as SE touches two very important aspects: Privacy, as already outlined in this section, but increasingly also the use of Artificial Intelligence (AI) methods. This is especially important with respect to regulations like the AI Act [25] and the Data Act [26], which are first attempts to regulate the use of information in AI. While these are currently limited to the European Union, these regulations could be exemplary for other legal regimes as well. Of course, real attackers will not care about the legality of their tool utilization, the topic is far more important for white hat social engineers that use the tools for enhancing SE security in companies: Since modern machine learning techniques require training with quite large amounts of high quality data, the question of the availability of legal training data needs to be solved. This also includes issues like membership inference attacks, where attackers can try to infer the existences of certain persons in the training data of a trained model, which could, again, pose a privacy problem. Further challenges result from the lack of explainability of modern Machine Learning (ML) tools [27], i.e., it is currently impossible to explain, why a specific model arrives at a specific solution, even in full knowledge of model, training and processing data. While this is certainly no problem in case of criminal use of the tools, it becomes a problem when white hat social engineers need to be able to fully determine the inner workings of attack tools in order to find countermeasures. In addition, even the white hat use of certain tools could pose potential legal problems, which has to be decided in the near future by the respective courts.

IV. Social Engineering models and frameworks

A standardized formulation of a Social Engineering attack, as well as the sequence and temporal events, allows researchers to compare different Social Engineering attacks with each other. Next, we will compare the following most common phase models and frameworks that divide Social Engineering attacks into phases: The *Cyber Kill Chain (M1)* [28], the *Social Engineering Cycle (M2)* [2], the *Social Engineering Lifecycle (M3)* [29], the *Social Engineering Pyramid (M4)* [8], the *Social Engineering Attack Framework (M5)* [30], the *Cycle of Deception (M6)* [31], the *Social Engineering Attack Spiral (M7)* [32], the *Session and Dialogue Based Framework (M8)* [4], and the *Phase based and Source based Model (M9)* [33].

Following, we give a short overview on the most important models.

A. The Cyber Kill Chain

Originally developed by Lockheed Martin [28], the Cyber Kill Chain is one of the oldest and best known models that saw some extensions and changes since 2011, e.g., by IBM Security [34]. It consists of the following phases:

1) *Reconnaissance*: In the reconnaissance phase, targets (persons, institutions or specific persons in institutions) are selected and as much information as possible is obtained about them. Any information, no matter how small and seemingly unimportant, can be of significance for the further course of the attack.
2) *Weaponization*: In this phase, an attack is prepared based on the information previously obtained. On the one hand, a pretext suitable for the attack target is drafted and on the other hand, usable tools are compiled.
3) *Delivery*: In the delivery phase, the execution of an attack is started. Prepared phishing messages are sent to selected targets, prepared data carriers are deposited or water-holing pages are activated.
4) *Exploitation*: In the exploitation phase, security gaps and vulnerabilities of the attack target are exploited. This is also where vishing calls take place, which can persuade the attack target to co-operate and help.
5) *Installation*: In this phase, malware is installed unnoticed on the devices of the targets. This can happen via the

previously prepared data carriers or via one of the activated water-holing pages.

6) *Command and control*: In this phase of the Cyber Kill Chain, the previously installed malware is used to obtain data, further personal information or access data.
7) *Action on Objectives*: In the final phase of the Cyber Kill Chain, the attacks are concretised, systems are compromised and data and access data obtained are exploited to complete the attack.

A major criticism of the kill chain is its focus on malware, as well as on the pure attacker perspective, a criticism that it shares with many of the other models [35]. Furthermore, it is neither cyclic in nature, nor does it allow for the repetition of intermediate phases in the original version, which makes in rather cumbersome to model realistic targeted attacks with attackers moving inside a system and gradually taking it over. This is especially problematic in the light of Advanced Persistent Threats (APT), where attackers are highly persistent and probe the system in many ways [35]. Due to its acyclic nature. the Cyber Kill Chain focuses on a single intrusion attempts, which does not reflect attacker behavior in the case of APTs. Due to the popularity of the Cyber Kill Chain, several enhancements have been proposed, e.g., by providing a holistic model that also includes legal aspects and policy making [36].

B. Social Engineering Cycle and similar approaches

In contrast, the Social Engineering Cycle by Mitnick and Simon [2] has a non-technical focus, which can be seen in the four phases that, again in contrast to the Cyber Kill Chain, are defined as a cyclic approach: (i) Research, (ii) Developing rapport and trust, (iii) Exploiting trust and (iv) Utilization of information.

An attack begins with the *Research phase*, in which information is gathered and research is carried out on the respective target. This can be done via all possible channels (e.g. public sources, annual reports, marketing documents, newspaper articles, websites, content from social media). With more detailed information and insider information, identities are assumed and references are made to people known to the victim. In the next phase, *relationships and trust are developed*, which are then exploited in the subsequent phase. In the *exploitation of trust* phase, the victim is asked for favours and actions. A special form of "reverse sting" also occurs here, in which the victim asks the attacking side for help. In the final phase, the *information gathered is utilised*. If it turns out in this phase that something is still missing to finally achieve the goal, it is possible to return to an earlier phase of the cycle. This continues until the attacking side has achieved its goal.

When searching for social engineering life cycles or phase models, the *Social Engineering Lifecycle* of the internationally active IT security company Imperva [29] needs to be mentioned. Imperva also uses a 4-phase model to illustrate the life cycle of social engineering attacks, similar to Mitnick's model, but with different phases and names: (i) *Investigation*, where the foundations for an attack are prepared. The victims of the attack are selected, background information about them is gathered, and suitable attack methods are chosen. (ii) *Hook*, where the aim is to deceive the victims of the attack and gain a foothold with them. Contact is made with the target, they are deceived with an invented story and control is taken over interactions. (iii) *Play*, which revolves around information that is retrieved over a certain period of time. The implantation from the previous phase is deepened, attacks are carried out, business processes are disrupted and/or data is siphoned off. Finally, (iv) the *Exit* phase, where the attack is completed, ideally without arousing suspicion. To this end, all traces are covered, malware is removed and the pretext, the story that was invented in the hook phase, is brought to a natural conclusion. This is rather different to the model of Mitnick and Simon which does not explicitly tie up lose ends and go for a safe exit.

The *Social Engineering Pyramide* by Hadnagy [3] is also very similar to the Social Engineering Cycle, with the notable deception that it is linear instead of cyclic. Furthermore, it is the only one of the models analyzed in this work that has an explicit reporting step included, which was especially included by Hadnagy, as he used this approach for penetration tests for customers, thus reporting was of the utmost importance.

Another approach derived from the works of Mitnick and Simon is the *Social Engineering Attack Framework* by Mouton [30], which was explicitly stated to be an extension in order to cover shortcomings in the original cycle. In comparison, the social engineering attack framework generally consists of several more phases and is more detailed, especially at the beginning, as the target of the attack cannot yet be clearly defined at the start and it is not yet clear which target persons could possibly help to achieve the desired goal. For this reason, Mouton et al. introduced an additional "Attack Formulation" phase. Furthermore, the "Information Gathering" phase is more detailed in terms of the evaluation of the information gathered, as this is of great importance for the further course of the attack and the subsequent trust relationships to be established are heavily dependent on the quality of the information obtained from this phase. Another important and additional phase, "Preparation", in which data is prepared and attack vectors are selected, is found before the "Develop Relationship" phase, which is very similar but differs in the entry point. The "Exploitation Relationship" phase is also described in more detail in this framework. Finally, there is the additional debriefing phase in which the target persons are to be put back into a normal emotional state (maintenance process). The idea here is to make the target person feel good so that they do not feel as if they have been attacked, in order to counteract feelings of guilt from (unauthorised) disclosure of information and thus avoid unforeseen consequences. In the transition process within the final phase, a decision is made as to whether the target of the attack has been achieved or whether it is necessary to return to an earlier stage (e.g. to obtain more information). As this approach is far more complex when compared to the others, the original figure from the original paper [30] is provided as Figure 1.

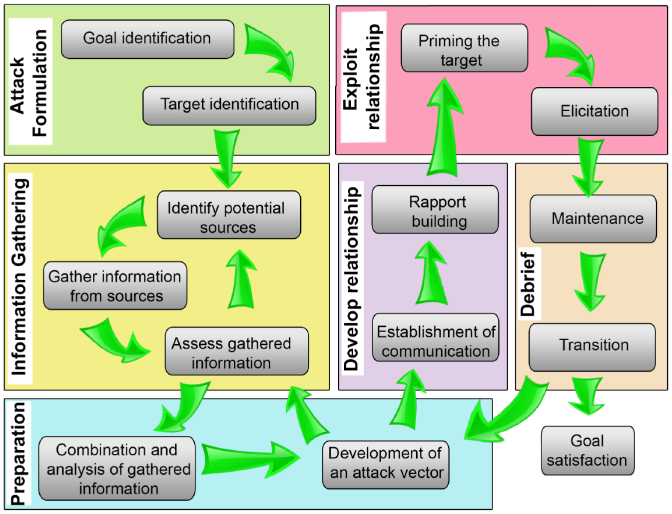


Figure 1. Social Engineering Attack Framework by Mouton et al. [30].

C. Cycle of Deception

The *Cycle of Deception* [31] is a social engineering framework that not only includes the phases from the perspective of the attackers, but also those from the perspective of the attack victims and their defenders. The model was developed because the frameworks available at the time were considered too simple and at the same time too opaque. According to the authors, it is intended, among other things, as an aid for training purposes, but also as a model for a holistic protection strategy against social engineering. The framework is typically depicted in the form of three concentric circular cycles, with the outermost being the *Attack Cycle*, the next the *Defense Cycle* and the innermost the *Victim Cycle*. Each cycle consists of 5 steps that not only work in circular order, but also relate to their counterparts in the other cycles.

a) Attack Cycle: The Attack Cycle is dedicated to the behaviour and actions of the attackers with its included phase: (i) *Goal & Plan* that includes the aim, purpose and justification of the attack, (ii) *Map & Bond*, where attackers use various search techniques to gather information about the attack targets, (iii) *Execute*, where the attackers carry out an unauthorised or punishable act, (iv) *Recruit & Cloak*, which refers to all activities to conceal traces after an attack has taken place and (v) *Evolve/Regress*, where he attackers learn from the process and create an internal justification for what happened.

b) Defense Cycle: In the direction of the center, the attack target, is the next defense cycle, which is dedicated in phases to the options available to the defenders. In some cases, the role of the defenders can be played either by the victims themselves or by IT professionals: (i) *Deter*, providing a deterrent effect through appropriate guidelines and perceptions of good reporting lines in the event of incidents, (ii) *Protect*, providing a small amount of sensitive data, training measures for employees and an appropriate policy provide protection in this phase, (iii) *Detect* describes the detection of attacks by attentive employees or by technical equipment, (iv) *Respond* by creating ways to easily report social engineering attacks or attempts to do so and (v) *Recover* that includes knowledge of the value of your own data, good existing policies and well-documented, reported attacks in order to learn from them.

c) Victim Cycle: The Victim Cycle is placed directly around the attack target and focuses on the behaviour of the individual victims, to whom the authors believe too little attention is paid when analysing attacks: (i) *Advertise*, the victim (knowingly or unknowingly) possesses something of value that makes them a target, (ii) *Socialize & Expose*, where by interacting with the attackers, the victim can be deceived into giving up their valuables or access to them, (iii) *Submit*, the release of e.g., secret information, (iv) *Accept & Ignore*, referring to the behaviour of the victim after an attack has taken place, in that it was accepted, ignored or not noticed at all and (v) *Evolve/Regress*, describing the development of the attack target into the role of the learner, or into the role of the victim.

D. Comparison and Technical Social Engineering model (TSE)

These models differ most clearly in the area of representation. With M1, the M4, M8, and M9 represent in successive process steps, the M2, M3, M5, M6, and M7, respectively, represent in circuits. The fact that the majority of the researched frameworks use a circular structure to describe Social Engineering attacks, which mostly includes the phases of information gathering, trust exploitation, attack development, and target fulfillment, is also already described in [4]. The circular form provides the possibility of representing the repetition of previous phases when more information is needed, or the goal is not achieved in a single phase [2]. M6 does not provide the opportunity to return to a single previous phase, but provides a sequence of several cycles spherically on top of each other, which makes this framework seem to be very complex at first sight, especially in combination with the inclusion of risks as a three-dimensional component. The models and frameworks also differ in terms of the number of phases. Apart from two models, all other models were designed with fewer than eight phases. M1 is only to a limited extent suitable for Social Engineering attacks, since these types of attacks do not necessarily have to pass through all phases of the framework. Also, the complete section, in which relationships and trust are established, as well as exploited, is completely missing. M4 shows five phases and is the only model that includes reporting as the final step, for traceability and documentation of the process and results. The model M3, as well as model M2, are limited to a total of only four phases with similar names. M2 is seen as a good basis in comparison with M5, but too simplistic, according to [30], as it leaves too much room for interpretation and does not include a debriefing phase, which is intended in M5 to bring the target person back to a normal emotional state. No matter how many phases the respective models and frameworks have, a phase for thorough information gathering is required at the beginning of every successful Social Engineering attack, since the quality of the information obtained contributes significantly to the success of the subsequent phases. Based on the compared models and frameworks, the Technical Social Engineering model (TSE) was designed, shown in Figure 2, which was reduced to only three common phases, within which automation with tool support is possible.

Figure 2. The Technical Social Engineering model (TSE).

A corresponding assignment of the phases of the previously described phase models and frameworks to the phases of the reduced model can be seen in Table I.

V. Tool-supported automation for Social Engineering

While we tackled a lot of different tools during our analysis, we will only be able to give a short outline on the findings in this section, grouping the tools according to the previously defined TSE model.

The tools in the information gathering phase are used to obtain all kinds of information about a (potential) target. Included in this phase are also tools used in reconnaissance and OSINT, as well as Social Media Intelligence (SOCMINT). Still, as this is not an analysis of OSINT tools, we did not further dive into the extreme amount of apps there. We divided the tools into (i) web-based and (ii) locally installed tools.

A. Web based tools for Information Gathering

1) Searching for user data: Google Dorks are pre-defined searches that can be executed using the Google Programmable Search Engine for automation as Custom Search Engines (CSEs). This allows for fine-tuning and exchange of fine-tuned searches, which can be accesses through catalogs. One of the most prominent of these catalogs is the Exploit-DB [37]. At the time of the research, the then current status of the exploit database was 7,341 Google Dorks. Using the Google Programmable Search Engine [38], it is also possible to save search queries online to Custom Search Engines (CSE). These CSEs are also publicly accessible and usable for the general public. A CSE by Brijesh Singh that is specially tailored to social media platforms is available at [39], while Stefanie Proto lists over 130 other available and directly usable CSEs in the compilations [40] and [41] at the time of research.

Another important source for tool gathering are *OSINT link lists*. During the research on automated social engineering tools, links to lists with hundreds of links to web applications were often provided in relevant forums, which are suitable for OSINT purposes, but which can also support the information gathering process within social engineering. Bellingcat [42], a Dutch-based group of investigative journalists specializing in OSINT investigations, provides a compilation of useful web applications for use at [43] and [44]. Similar information can also be found on the homepage of the OSINT researcher with the pseudonym "Technisette" [45], as well as in the other sources listed below:

- *Technisette Tools [45]*: Web-based OSINT tools and web applications to support online searches, links to other partner platforms, social media online search engines.
- *Bellingcat's Online Investigation Toolkit [43], [44]*: Compilation of several hundred web-based tools to support information gathering, grouped according to application areas (e.g. image search engines, social media, people search and much more).
- *OSINT for Journalists [46]*: Media map and link list with links to various OSINT online search engines, tool collections, links to other extensive link lists, web applications and databases.
- *Search Social Media [47]*: Numerous online search engines grouped according to social media platforms (Twitter, Reddit, Periscope, Tumblr, Facebook, Instagram, YouTube, LinkedIn, TikTok, Telegram, Snapchat, Pinterest). Links to other link lists and search engines for information on people, user names, telephone numbers and email addresses.
- *Ph055a GitHub Repository [48]*: GitHub repositories `Domains_OSINT_Collection` and `OSINT_Collection` contain link lists with numerous links to OSINT resources available online, such as search engines for users across several hundred social media platforms, search engines for information about companies, search engines for searching leaks, but also links to online resources to investigate domains and IoT products, such as subdomain enumerators and crawlers, link checkers, DNS info, similar site search and much more.
- *OSINT Framework [49]*: An animated OSINT tool link collection that offers freely available search engines and web applications for searching and enumerating user names, domains, e-mail addresses and archives as well as documentation and training material.

The number of links in these lists is so extensive that it was not possible to carry out a precise review as part of this work. Random checks showed that not all links were functional and not all tools worked automatically. It also turned out that links to similar pages are included, which in turn contain a large number of tool links. It also turned out that the listed tools, search engines and browser plugins are very often similar.

Regarding Social Media platforms, the web application *CheckUsernames* [50] allows the parallel search of over 300 platforms for user-names and linked profiles. Still, the search is very limited, only allowing for exact (partial) matches without additional intelligence. *ReconTool* [51] provides several additional features, like e.g., mindmapping information for dynamic interaction with the search engine. Even more extended functionality is provided by *HOPain Tools* [52], [53], as it also allows searching for pics, videos, detailed content

TABLE I
PHASE ASSIGNMENT

Model	Information Gathering	Attack Preparation	Attack Execution
M1	Reconnaissance	Weaponization, Delivery	Exploitation, Installation, Command & Control, Action on Objectives
M2	Research	Developing Rapport and Trust, Exploiting Trust	Utilize Information
M3	Investigation	Hook	Play
M4	Information Gathering	Attack Planning	Perform Attacks
M5	Information Gathering	Preparation	Exploit Relationship
M6	Map & Bond	Execution	
M7	Recon	Relationship Building, Attack Scenario Building	Execution, Action on Objectives
M8	Attack Preparation		Attack Implementation
M9	Using suitable gates of SNSs to gather information about victim	Using suitable gates of SNSs to reach the victim	Attack

like postings (also allowing filtering like time frames, location or number of likes), as well as bitcoin addresses. Social media platforms can be searched individually or in groups, for many platforms require a respective account.

2) Technology checks: In order to expand the possibilities of pretexts and impersonations for Social Engineering in organisations, it can be helpful to examine existing websites for the technologies used and possible vulnerabilities. The following tools can be used as an alternative to considerably more expensive systems due to higher licence and operating costs. The result of a scan with *BuiltWith* [54] shows the technologies, plugins and hosting provider used for a website, but also other websites that use the same hosting provider, as well as the duration and the respective public IP address under which they were accessible. However, the results can only be viewed to a limited extent in the free version, but are sufficient for searching for *Common Vulnerabilities and Exposures* (CVE) entries and for developing pretexts. Technological information, telephone numbers, email addresses, CVE vulnerabilities with the corresponding CVE number, public IP addresses used, open ports, domain names, cybersquatting domains and much more to determine further attack surfaces and risks of a website can also be found out very conveniently with *SpiderFoot* [55]. The *SpiderFoot HX* [56] version offers an even greater scope and an intuitive, graphical interface that can display all this information in the form of a node graph, where each node can be selected individually. The scan results were surprisingly comprehensive and consistently correct in the short time available and in view of the basic version used. Regarding the analysis of industrial (IoT) devices, Shodan [57], ZoomEye [58], Spyse [59] and Chaos [60] seem to be the most popular. Shodan provides many filter options and requires a familiarisation period in order to achieve useful results. The search results depend on the time in which Shodan has scanned the target system, but contain a high level of detail about the scanned target system. Despite language barriers, ZoomEye could be used with translation software at the time of the research and the presentation of the search results was very similar to Shodan. Surprisingly, Spyse was only able to deliver a few results during the application and using identical target systems and is therefore not very suitable for Social Engineering purposes. Chaos was still at an early stage of development at the time of the research. On the other hand, SynapsInt [61] is a freely available tool that also fits into this categorisation. It provides search results for domains, IP addresses, SSL certificates, email addresses, telephone numbers and Twitter accounts, as well as searching for ransom bitcoin addresses and CVE numbers. The results of a scan with the same inputs as before quickly delivered correct results, a current screenshot of the page, a VirusTotal analysis, the last available entry in the Internet archive Wayback Machine, open ports and information on the hosting provider used. In addition, all domains that can be reached under the same IP address, all subdomains, internal links and related social media links are listed and checked to see whether it is included in various blocklists. The blacklist check also works with entered email addresses. The leak check and the Twitter account check did not work with a private email address that has already been leaked many times.

3) Generate valid email formats: In order to generate the formats for E-Mail addresses of targets, we had a look at the search engines *Email-Format* [62] and*Hunter.io* [63]. Hunter, as well as Email-Format, derive patterns for corresponding email address formats from a large number of email addresses collected via web scans. Of the target domains entered for testing, around a third did not return any search results. The email address formats derived in both web applications appear correct, and sample data is also displayed freely in both applications, although it is not always up to date. Email address format offers, in addition to the identified conventions, a larger list of representative email addresses, as well as (depending on the payment plan) the option of downloading them. In comparison to Email Format, Hunter tends to limit the output, but in addition to more up-to-date data records, it also shows the occurrence of the representative email addresses, which are used to derive the logics for the email addresses.

4) Data breaches and data leaks: Regarding searching data breaches and data leaks, the IntelligenceX platform [64] retrieves results from Dataleaks, Wikileaks, paste sites and even the darknet for search queries, such as email, Bitcoin, MAC and IP addresses, domains, URLs, telephone numbers, credit card numbers and much more. IntelligenceX offers a so-called ”Third Party Search”, in which the search scope can be extended again to several search engines (simultaneously via pop-ups) and, for example, Vehicle Identification Numbers (VIN) can also be searched for. There are separate search functions for social media channels, links to OSINT link lists, as well as file and encoding tools. The test searches carried out delivered surprisingly accurate results. A privately used, knowingly leaked email address that was no longer in use was found, including the password used at the time of use. For another, still privately used email address, it was possible to find out in which data breach the email address appeared and which platform was affected by the breach. Valid access data was also found for other email addresses in the private sphere; Reverse image searches from the third-party search category with randomly uploaded images from private collections and quick Google searches, mostly referred to Adobe stock images, however; three out of ten uploaded images were found. The VIN search was also tested with two different VIN numbers from our own stock, but the search yielded no results.

5) Detecting online times: Online times of targets are especially interesting for targeted attacks. The tool *SleepingTime* [65] was analysed for the SNS platform Twitter and successfully used with several Twitter accounts. SleepingTime analyses the last 1000 tweets of a Twitter account and derives an estimated ”sleep schedule” from the time stamps of the respective tweets, in which the account is least active and in use. *WhatsApp Monitor* [66] is a similar tool that works with browser notifications when a specific WhatsApp contact is available online. The use of the tool sounded very interesting during the research, but could not be used at the time of the tests, as the website was not accessible at the time of the tests.

6) Searching for personal information: Regarding searching for personal information. Suche nach Personendaten, *Webmii* [67] compiles publicly available information about people on the Internet and uses it to generate an online score that is intended to show the availability of the person. Webmii usually lists the results in four sections. (i) the results list, containing the names of people who have interacted with the target person on social media channels, (ii) search results from various newspaper articles, (iii) results from various social media channels and (iv) search results obtained via a Google CSE. At first glance, *IDCrawl* [68] offers a wider range of functions, as it can be used to search not only for people’s names, but also for user names across 17 SNSs. A reverse phone search is also offered. IDCrawl offers the option of an ”opt-out”, where you can exclude yourself from search results. During the test and the search for own findable information, IDCrawl was only able to verify one search result as correct, but the topicality of the result was doubtful, as in this specific case the user profile picture did not match and had already been replaced some time ago. However, the accuracy of the data is not guaranteed in large quantities at Webmii either, as only parts of the information could be considered correct as well. The majority of the search results were not usable, and in some cases links to results could not be opened at all.

B. Locally installed tools for Information Gathering

1) Maltego and alternatives: The data mining tool Maltego [69] is one of the best-known tool suites in the OSINT environment and is almost unique in its range of functions. Depending on the licence and the added plugins, the scope anc capability of the software change. For the tests and the tool comparison with a similar tool, the registered, free Community Edition with eight free plugins was used, which provides a certain number of credits depending on the query used. With six out of one hundred available credits, it was already possible to find domain information, whois entries, company owner data, email addresses, telephone numbers, public IP addresses, all plugins used on the website, as well as archived versions of these since 2009. Audit reports from American companies in the same business sector were also found in the Maltego document cloud. However, these were not related to the exemplary target company. As part of the research, a comparable alternative, or supplement, to Maltego could be found, which, despite critical voices [70], was implemented, licensed and tested for comparison: Lampyre [71], which is only available on Windows platforms and offers a similar overview to Maltego’s Transformation Hub in the so-called ”List of requests”. The advantage of the software is that the plugins do not have to be installed individually; a selection (and like Maltego, the entry of a corresponding API key) of the modules to be used, the underlying and desired tasks, as well as the required parameters, is sufficient for the start.

In direct comparison, Maltego is clearer and more structured to use. Lampyre is simpler in terms of usability, the results are mostly displayed in tabular form and graphical dependencies are only possible in isolated cases. Furthermore, it is partially unstable, e.g., during the application tests, various result tabs suddenly stopped responding and could no longer be selected, meaning that the results could no longer be viewed.

Of the plugins already included, Lampyre offers a selection of search criteria that could not yet be found in Maltego and vice versa. These included, for example, the search for IMEI numbers, WLAN SSIDs or Vehicle Identification Numbers (VIN) in Lampyre, while Maltego offers the Wayback Machine, Movie Database, Blockchain.info or Google Maps Geocoding, which are regularly updated and expanded in both applications. Within Maltego, the origins of the search results and the use of the search providers are traceable. At first glance, it is not possible to recognise where Lampyre obtains the results of the transformations if the search provider is not described in the tasks. In the transformations to the same target organisation, more search results could be achieved with Maltego with less known data. The reliability of the data was also higher in Maltego; for example, the public company Facebook account could be found with Maltego, whereas Lampyre returned error messages for these transformations.

2) Searching for user and personal data: Regarding searching of account or personal data, *CrossLinked* [72] allows for automated searches in LinkedIn by filtering external search engine results, so-called *Search Engine Scraping*, thus not requiring account data for searching. When verifying the results, it was found that although they were plausible (by randomly comparing the results with the online employee directory), but the results also included every person who had specified St. Pölten UAS in their LinkedIn profile, not only employees. When searching for another organisation without results, it turned out that links from search engines were also counted as results. The tools UserReCon [73] and Userrecon-py [74], Nexfil [75], Sherlock [76], Us3R-F1nD3R [77] and Thorndyke [78] promise similar functionalities with search scopes spanning several hundred social media platforms. From the own descriptions and command references of these tools, it is clear that Sherlock is the only application that can process several search entries as well as prepared lists in one search run. The tools are very similar in their use and appearance, as are the results. In addition to existing social media accounts, the Instagram test account `@dominikhatkeininsta` could also be found as a registered user on several platforms according to the search results. As the test account was only created for Instagram, it can be assumed that the search results are not valid, except for the Instagram platform. This was confirmed when checking the search results for the Twitter and Reddit platforms. Buster [79] can also find users on social media platforms, but the search scope is extended to the generation of email addresses, which are provided from possible data breaches, pastes and reverse-whois queries. Buster also shows the sources of results, as the services of Hunter.io, among others, are used in the background.

3) Technology checks: Regarding checking for technology, *TheHarvester* [80] is already pre-installed under Kali Linux and offers searches for domain information and Google dorks in 38 different search engines. Corresponding API keys are required for use, and the search results can be limited in scope. In the test, the search engines did not work properly under version 4.0.3, despite reinstalling the tool; under version 3.2.2, search results could at least be obtained via Google, although most of them were not valid. Raccoon [81] is basically an extension of nmap. The tool is still in the development stage and the focus is on simplicity. The convenience of using Raccoon lies in the fact that the parameterisation of the nmap scans is already predefined by the tool. In addition to the possibilities of nmap scans and subdomain enumeration, Raccoon should also be able to search cookies, recognise web application firewalls and provide information on CMS, web servers and Whois queries. However, this did not work in the test (without nmap scan). A coherent subdomain enumeration could be carried out using three different domains, including that of the St. Pölten University of Applied Sciences, with Sublist3r [82], Sn0int [83] and Frogy [84], whereby Frogy also uses Sublister in the enumerations. Sublister also offers the option of a port scan and a brute force scan, which were not performed. Under Sn0Int, the subdomain enumeration is only a small part of the functionalities. Frogy was still under development at the time of research and testing. In addition to finding IPs, domains and subdomains, it is also designed to find live websites and login portals. What is particularly interesting about this tool is that it can access the Chaos-database. Another tool suggested in the information retrieval communities is ReconSpider [85], which is a tool for the automated scanning of IP and e-mail addresses, websites, telephone numbers, DNS and domain information, but also for searching data breaches. ReconSpider was able to consistently return correct data in the test entries, but occasionally crashed with Python errors when making entries in the menus for whois and domain queries.

4) Export data from social media: Regarding the export of data from social media profiles, ReconSpider can display information of Facebook, Twitter and Instagram accounts, but this is limited to the name, number of followers and profile description and cannot be exported. The tool OSINTGram [86] on the other hand requires a valid Instagram account to be usable. For export, optionally in *.txt and *.json file formats, all addresses that can be read from posted image material, all texts and comments that have been added to posted images, the number of followers of the target account, as well as the number of accounts that the target account follows, account information, as well as the number of all likes, hashtags, a list of all links of the target account and a list of all accounts that have commented on posts of the target account at any time are available. The ”fwersemail”, ”fwingsemail”, ”fwersnumber” and ”fwingsnumber” functions are particularly interesting features for Social Engineering purposes, each of which creates a list of telephone numbers and email addresses (if specified in the respective accounts) of the followers and followings. In the test application with the Instagram account of the St. Pölten University of Applied Sciences, several thousand pieces of data were found. With a private test account, the consistently correct information could be provided in lists within a short time. Sterra [87] also exports follower and following accounts, including their account ID, user name, specified name, biography, number of posts and links to the respective account in CSV files. Within the application, it is also possible to compare follower lists with each other and filter them for similarities or differences. As Sterra works directly with Instagram’s API, the reliability of the data is guaranteed. List comparisons can also be carried out with the Python tool Insta-Extract [88] and these are simpler in the application than within Sterra, but not as extensive. What works well on the social media platform Instagram in the test applications also works with two other applications on the Twitter platform. Twi1tter0s1nt [89], also known as TWINT and twosint, offers pretty much the same functions on the command line that TinfoLeak [90] also offers in a GUI. These include general searches for user names, searches for geocoded tweets (if the geolocation data in the tweets can be read), tweets in a specific time window, filtering for specific terms, but also exporting the number of followers. In addition to exports in several file formats, TWINT also offers to translate tweets directly into other languages using Google Translate. A time limit between individual scrapes can also be set for scraping tweets using the ”min-wait-time” parameter. TinfoLeak is easier to use with the graphical user interface, where the desired operations are simply ticked and provided with the corresponding values or data.

C. Ready-to-use system environments

The information procurement phase is very extensive due to the large number of applications available. Automation is largely attempted to be created within an application in order to automate and positively influence time and effort through recurring activities and queries (for example, the same searches for different user names on social media platforms). Applications such as Maltego and Lampyre use plugins from various manufacturers and developers to offer automation with various and different search queries within their own application. During the research for social engineering tools with automation and possibilities for this, two Linux distributions could also be found, with which no complete automation can be created in the process of information retrieval, but the effort is greatly simplified by the convenient operation.

1) Tsurugi-Linux: Similar to the Linux distributions Kali and BlackArch, the ready-to-use distribution of the Tsurugi Linux project [91] is structured in a similar way. The distribution is completely free and includes a variety of tools that can be used for the purposes of digital forensics and malware analysis. The distribution is based on Ubuntu and is available for download in three versions. Two of the three versions are available as a live system, while the third version can be downloaded as a ready-to-use image for Oracle VirtualBox. Tsurugi is a double-bladed sword used by Japanese monks. The metaphor of the double-bladed sword has also been transferred to the distribution: there is a profile switcher that switches from the digital forensics environment to the OSINT environment, making numerous tools for information gathering and reconnaissance purposes conveniently available in the start menu with just a few mouse clicks. A list of pre-installed tools can be viewed at [91], some of which were also discussed in this paper independently of this distribution. Similar to Kali and BlackArch, the tools must be started manually, but the ease of use is increased by the profile switcher and thus simplifies the process of information retrieval.

2) CSI-Linux: The Linux distribution CSI-Linux [92] is also designed for digital forensics. CSI-Linux optimises the time and effort involved in the process of obtaining information by using several tools, which have also already been discussed in this paper, to enable the pre-parameterised starting of applications with a so-called "case management" and to store the search results clearly in a corresponding folder structure. Each new investigation process starts with the creation of a new case file, after which the desired type of investigation is selected. This can be ”Social Media Intelligence (SOCMINT)”, for example. The respective launcher is kept so simple, even when selecting a different investigation (e.g. ”Domain and Website OSINT”) that you only need to select what you want to search for. Special knowledge of and in programmes and applications, as well as the parameters required for use, is therefore not necessary. The handling of API keys, some of which are subject to a charge, is also kept simple and clear with this workflow-like user interface. Keys can be added, exchanged or removed conveniently with just a few mouse clicks. CSI-Linux is also available as a ready-to-import image for Oracle VirtualBox. In addition, it is also offered as a bootable image in the form of a forensic RAW image. For support, there are also instruction videos and walkthroughs for various application purposes at [92].

D. Tools for the attack preparation phase

The attack preparation phase includes those tools that, depending on the selected attack scenario, are useful for preparing attacks, e.g., for preparing payloads or phishing messages.

1) Preparing Payloads: To prepare suitable payloads, already generated and available versions [93] can be used, or new ones can be generated. In addition to one of the best-known tools, the Social Engineering Toolkit (SET) [94], the PowerShell script [95] designed by Matt Nelson and Matt Robinson is also suitable for this, which creates an Excel document after the run that creates a Meterpreter shell when called on the target system. It also persists in the Windows registry and in the user directory so that it can be executed again when the system is restarted. A connection to the infected system can be established via Meterpreter Reverse HTTP and HTTPS. The MacroPack tool from Emeric Nasi [96] is more up-to-date and has an extended range of functions compared to the PowerShell script and requires a functioning and registered Office installation on the system on which the payload is to be integrated into an Office file. The tool also offers the service of code obfuscation so that the malicious code in the Office markers is not so easily recognisable and it supports all Microsoft Office document versions and shortcut files in the community version. The Pro version offers an even wider range of functions and can be used on existing Office files. During the tests, the generation of payloads with the PowerShell script did not work, despite changes in the execution guidelines, which originally prevented the execution of the script. For the execution and use of MacroPack, it is recommended to adjust the Windows security settings, as these prevent execution and classify the tool as a serious threat. The tool Social_X, which was supposed to be able to generate Trojans with its own reverse shell and in the form of an *.exe file, unexpectedly failed to install correctly and terminated after several start attempts. Documentation for the tool was not available at the time of testing and a linked YouTube video was no longer available. Social_X is therefore only mentioned as another possibility, as the last commit on GitHub was only a few months old and the error could possibly be fixed soon.

SET, which is included in every current installation of Kali-Linux, offers the option of automatically manipulating data carriers, so that malicious code can be automatically executed on removable media via the autorun function. This can be done via an executable file, which is executed via the autorun.inf file contained on the removable storage device, or via a file format exploit to bypass any security warnings. TrustSec also provides detailed documentation on SET. SET worked out of the box and, with the TrustSec documentation, was simple and reliable.

2) Recognising tone and emotions in texts: In order to test messages for the effect of emotions, the Tone Analyser [97] from IBM was tested during the research into automated Social Engineering tools. The Tone Analyzer can be freely tested online in a web form and recognises the emotions and tones of voice contained in an entered text via machine learning analysis. The Node.js version of the Tone Analyser [98] offers free analyses and support for several languages and files directly for the first 1000 API calls per month after registration in the IBM Developer Cloud. To quickly test the analysis, the following sample texts were entered for analysis:

- *Positive emotion: "Dominik likes doing his master thesis all night long :-)"*
- *Negative emotion: "Dominik does not like doing his master thesis all night long :-("*

Tone Analyzer carried out the analyses with respect to the emotions "Confident", "Joy" and "Sadness" and classified the strength of the expressions in the messages with different colours. In further tests, with different text fragments, Tone Analyser also classified in the direction of "Analytical" and "Tentative". We did not conduct any further tests, as this work is not focusing on the capabilities of emotion detection, but on the general usability of the tools.

3) Bot preparation: Parts of a Social Engineering attack can also be carried out by bots, depending on the target and attack scenario selected. Implementations of Twitter bots, modelled on Realboy [99] or SNAP_R [100], for example, can be used in the attack execution phase for the automated distribution of phishing links. In the attack preparation phase, corresponding Twitter accounts can be created, filled with content and equipped with a network of followers and followings to make them more credible. Both bots, Realboy and SNAP_R, were not tested and evaluated in this work, as there exists ample recent work analyzing bot preparation for Social Engineering.

E. Tools for the attack execution phase

The attack execution phase includes all those tools that can directly execute a Social Engineering attack. While researching the relevant tools, it emerged that the automation of attack tools is described almost exclusively in terms of phishing with website cloning, mass emails and occasionally the use of bots.

1) Phishing with website cloning: SET [101] offers the possibility to clone any website into a website with phishing or hosting multiple attack methods. The cloned page is ready for use as soon as it is entered, and the user data entered is displayed in colour directly on the command line. Zphisher [102] works in a similar way, also with regard to website cloning. Unlike SET, however, Zphisher only offers ready-made templates for phishing pages and does not clone individual pages. This is also the case with phishEye [103], although it is the only tool listed that also offers the option of cloning websites for mobile devices. During the application tests, it was found that although Blackeye [104] provides a number of templates for social media platforms, these could not be tested directly as an error occurred when generating the phishing links and no links were generated or output for use. SocialFish [105] could also not be fully tested and evaluated, as module error messages occurred within the main application when the application was started, despite all installed requirements and dependencies. The documentation for the app is very brief and rudimentary, so the error could not be rectified. Cloning the GitHub repository again did not help either. StormBreaker [106] extends the list of phishing tools mentioned in this subsection with a tool that cannot clone websites like the others mentioned so far, but instead generates pages and links with the help of Ngrok with a maximum of two inputs, which enable access to the camera, microphone and location data of the end devices. The location data is returned with a Google Maps link. StormBreaker also offers an "OS Password Grabber" function, which is designed to transfer the passwords entered. During the tests, there were difficulties with this part of the function, as either the links to be sent were not generated or the application did not respond to inputs. However, the functionality of accessing the microphone, camera and location data of the potential target's device is only possible if all phishing warnings displayed by the current browser generations are ignored when the page is accessed and authorisation to access the microphone, camera or location is granted accordingly.

2) Mass mailer: In addition to individual (spear) phishing messages, the Social Engineering toolkit SET [101] can also be used to set up the sending of mass emails. The email addresses of the recipients can be provided via a separate text file, and a separate mail server or sending via Google Mail (gmail) can be selected for sending. The message content is accepted in both HTML and plain text formatting. A test mailing with SET was carried out using our own mail server. As expected, the e-mail message was classified as SPAM and filtered accordingly. In many cases it is not clear before sending a message whether it will be blocked by a mail server or whether it will be delivered without any problems. In order to check the behaviour of mail servers when a message is received, a check can be carried out in advance using Phishious [107]. According to its own information, Phishious is the only tool to date that makes it possible to scan phishing attacks via email. Phishious analyses the header data of undeliverable messages and can therefore predict whether a message will be delivered or classified as spam or junk mail. Another mass mailer tool can be seen in Catero [108]. In addition to the option of cloning websites, Catero offers various ways of sending automated messages and can be controlled entirely via the Command Line Interface (CLI). Catero supports sending messages via Twillo accounts for sending SMS messages, sending via LinkedIn accounts and WebMail services, Google Voice and iMessage.

3) Bot utilization: Another type of automation of Social Engineering using bots is the preparation for the use of SMSRanger[109], which is based on a Telegram bot. SMSRanger sends automated messages to people, in each case on behalf of a bank, and asks them to enter OTP codes (One Time Password) in corresponding websites or in an automated call via a voice bot using the telephone keypad. The service contains daily updates, is available in various languages and is subject to a charge. At the time of research, calls from and to various countries, including German-speaking countries, were also included for USD 425 per month. SMSRanger is controlled via a Telegram chat. This bot was also not activated for security, legal and ethical reasons. With Honeybot [17], Tobias Lauinger et al. have already shown that conversations between two people can be started and influenced and controlled by the bot-in-the-middle, which can also be used to carry out attacks. The *Honeybot* tool is only mentioned in this section and was not tested or evaluated in this paper, as this has already been done in related work.

VI. Conclusion and Future Work

In this section, we provide some conclusion, but also references for future work in this fast evolving topic.

A. Conclusion

In order to better understand automation in the area of Social Engineering and to be able to search for suitable tools and tool suites, but also to be able to classify automation in different phases of Social Engineering, various Social Engineering frameworks were analyzed and compared with each other. It was found that the various models often differ in the number of phases and that classifying automated tools into individual phases in this way is not purposeful. Therefore, a compression to common phases of all models was carried out and from this, the *technical Social Engineering model* was derived. Furthermore, the individual phases of the described frameworks from other works were assigned to the phases of the technical Social Engineering model, using phase mapping. A similar and comparable abstract model could not be found by the time of writing this paper. For the listing and clustering of the automation-supported Social Engineering tools within Section V, the individual phases of the technical Social Engineering model were used. The clustering of the corresponding tools shows that in the information gathering phase there exists a lot of diversity and a large number of tools allowing for the most automation possibilities, as there is a large community of interested parties and contributors from the OSINT area. This was shown not only in the short intervals, in which tools and updates to existing tools are published, but also in the linguistic diversity in which the applications are written. The short intervals make it impossible to list and test all of the available tools. A selection of over 140 tools, written in German or English language, were subjected to a practical application and comparison, where it was found that information retrieval within the European Union has become more difficult since the introduction of the General Data Protection Regulation, and that web applications for information retrieval in particular largely only provide results in the states of the USA. There are, in the applications that are available free of charge, often query limits implemented that only allow a small number of queries within a certain period of time. Registering to receive an API key, shifts the query limits, depending on the chosen tariff and tool, but also the up-to-dateness, as well as the amount of data provided. Within this work, only freely available tools and API keys free of charge were used. Furthermore, it became apparent that results must be manually checked for plausibility and validity before further use, since the results of automated tools, with the exception of those that read information directly from social media platforms, are not necessarily correct or appropriate. When using the tools to gather information from social media platforms, most of the platforms require a registered account. When using the tools to prepare for attacks, it has been shown that automation can be summarized to the preparatory generation and creation of payloads and bots, as well as support in the formulation of texts. When using the tools in the attack execution phase, the researched and mentioned tools could be summarized into the categories ”phishing with website cloning”, ”mass mailers” and the ”use of bots”. A completely end-to-end automated software that can map a complete Social Engineering attack in all of its phases could not be found. The two tools Maltego and SET are, after completion of the tests and comparisons, the most functional and reliable tools.

B. Answering the research questions

The research questions posed at the beginning of the paper can thus be answered as follows.

a) RQ1: The freely available Social Engineering tools are automated in the sense that recurring query and search work can be performed automatically, thus significantly reducing manual effort. Searches can be performed via web applications, but also locally installed tools. Web applications shine with simpler operation and fast availability. The automation possibilities are greater when using the APIs of the search providers and platforms, since the results can be processed further in an automated manner if the appropriate output is available. A completely automated solution could not be found and is correspondingly difficult to develop, since Social Engineering can be very dynamic and the validation and decision as to, whether data and information fit a current target and scenario, must be made manually by the social engineers themselves. Automation is also already available in the execution of attacks and in the corresponding preparation, and the corresponding tools are already very easy to use. During the application and writing of the paper, it has become evident that the selection and availability of automated tools for the purpose of information retrieval is the largest. One justification of this can be the availability of a large community from the OSINT domain. Another reason can be seen in the greater availability of these tools, among other things for awareness-raising measures. With regard to quality, it was stated in the paper that the scope of the search and the number of permitted searches are subject to certain limitations, depending on the platform and are only increased with paid subscriptions. This also affects the reliability of the search results. Regarding availability, interesting tools could be collected during the research phase, but during the testing and application phase a few weeks later, they were no longer available and applicable. The free availability of automated Social Engineering tools means that these tools are available to any person, can be used by any person, and thus any person can easily use Social Engineering techniques, without much effort or in-depth knowledge. Due to the availability of ready-to-use system environments, pre-configured systems are provided, which, with a simplified graphical user interface, can deliver usable results within a short period of time, even for beginners.

b) RQ2: The various frameworks and phase models differ in terms of the number of phases, as well as the processes within the phases themselves. Generally speaking, the phases of reconnaissance and the phases, within which attacks take place, are best served and supported by automation. Due to the number of differences between the various Social Engineering models, it was not possible to map the automated tools to all models, which is why the abstract technical Social Engineering model was derived from the other analyzed frameworks.

c) RQ3: Records must be manually selected, validated, and formatted for the next tool. Toolsuites, which offer multiple options and whose functionalities can be extended with plugins, such as the mentioned tools Maltego, Lampyre, or also Spiderfoot HX, can transfer results into new searches most easily. These tools cannot guide a complete Social Engineering process, but they accompany a large part of it very reliably.

d) RQ4: The results of the tools depend very well on the respective mode of operation itself. While some of the tools, in order to deliver search results, make use of searching in archive databases or searching crawled and scanned websites, some tools access live data directly. In free program versions, live data was only analyzed by tools that search across social media platforms, for example Tinfoleak or OSINTGram, and required a corresponding user account. Searching crawled pages affects the reliability and the up-to-dateness of the results.

C. Future Work

As an extending future work, paid API keys of the applications, offering higher-value subscriptions, can be purchased and the results compared between the premium versions. Under appropriate legal and ethical coverage, extended use of the tools, including for awareness and training purposes, is conceivable. In the light of the increasing number of phishing messages, the comparison and use of professional Social Engineering tools, such as CanIPhish, GoPhish and SET, in the corporate context is a possibility. From this, organizational countermeasures, suitable for the respective organization, can be derived and an anti-Social Engineering framework can be designed. In the analysis of free tools, it was found that search platforms, including Hunter.io, Shodan.io, as well as _IntelX, were used in common by some tools. In the context of a future work, the comparison of which and how many search engines and databases are used in the background, together and whether the results, despite use of same sources, differ. Also, the development of an automated Social Engineering application, which can link the applications and results of different Social Engineering tools together, can be initiated.

Acknowledgments

Part of this work was funded by the Christian Doppler Laboratory for Assurance and Transparency in Software Protection, Research Group Security & Privacy, Faculty of Computer Science, University of Vienna. The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged. Part of this work was funded by the COIN project "Secure Supply Chains for Critical Systems" (SSCCS, 883977). The financial support by the Austrian Research Promotion Agency (FFG) is gratefully acknowledged.

References

[1] D. Dana, S. Schrittwieser, and P. Kieseberg, “Automated social engineering tools-overview and comparison with respect to capabilities and detectabilitys”, in *Proceedings of the Nineteenth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2024)*, IARIA, 2024.

[2] K. D. Mitnick and W. L. Simon, *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, 2003.

[3] C. Hadnagy, *Social Engineering: The Art of Human Hacking*. John Wiley & Sons, 2010.

[4] K. Zheng, T. Wu, X. Wang, B. Wu, and C. Wu, “A session and dialogue-based social engineering framework”, *IEEE Access*, vol. 7, pp. 67 781–67 794, 2019.

[5] E. D. Frauenstein and S. V. Flowerday, “Social network phishing: Becoming habituated to clicks and ignorant to threats?”, in *2016 Information Security for South Africa (ISSA)*, IEEE, 2016, pp. 98–105.

[6] J. Talamantes, *The Social Engineer’s Playbook: A Practical Guide to Pretexting*. Hexcode Publishing, 2014.

[7] P. Kim, *The Hacker Playbook 2: Practical Guide to Penetration Testing*. Secure Planet, LLC, 2015.

[8] C. Hadnagy, *The Science of Human Hacking*. Wiley Publishing Inc., 2018.

[9] Z. Wang, H. Zhu, P. Liu, and L. Sun, “Social engineering in cybersecurity: A domain ontology and knowledge graph application examples”, *Cybersecurity*, vol. 4, pp. 1–21, 2021.

[10] H. Aldawood and G. Skinner, “An advanced taxonomy for social engineering attacks”, *International Journal of Computer Applications*, vol. 177, no. 30, pp. 1–11, 2020.

[11] B. Banire, D. Al Thani, and Y. Yang, “Investigating the experience of social engineering victims: Exploratory and user testing study”, *Electronics*, vol. 10, no. 21, p. 2709, 2021.

[12] J. Obuhuma and S. Zivuku, “Social engineering based cyber-attacks in kenya”, in *2020 IST-Africa Conference (IST-Africa)*, IEEE, 2020, pp. 1–9.

[13] N. A. Hassan and R. Hijazi, *Open Source Intelligence Methods and Tools*. Springer, 2018.

[14] C. P. Janssen, S. F. Donker, D. P. Brumby, and A. L. Kun, “History and future of human-automation interaction”, *International Journal of Human-Computer Studies*, vol. 131, pp. 99–107, 2019.

[15] Z. Wang, L. Sun, and H. Zhu, “Defining social engineering in cybersecurity”, *IEEE Access*, vol. 8, pp. 85 094–85 115, 2020.

[16] M. Huber, “Automated social engineering, proof of concept”, *Royal Institute of Technology Stockholm*, 2009.

[17] T. Lauinger, V. Pankakoski, D. Balzarotti, and E. Kirda, “Honeybot, your man in the middle for automated social engineering.”, in *LEET*, 2010, pp. 1–8.

[18] J. Seymour and P. Tully, “Weaponizing data science for social engineering: Automated e2e spear phishing on twitter”, *Black Hat USA*, vol. 37, pp. 1–39, 2016.

[19] P. Kaul and D. Sharma, "Study of automated social engineering, its vulnerabilities, threats and suggested countermeasures", *International Journal of Computer Applications*, vol. 67, no. 7, pp. 13–16, 2013.
[20] Y. Kano and T. Nakajima, "Trust factors of social engineering attacks on social networking services", in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, IEEE, 2021, pp. 25–28.
[21] A. Stern, "Social networkers beware: Facebook is a major phishing portal", *Kaspersky Lab*, vol. 23, 2014.
[22] K. Kikerpill and A. Siibak, "Mazephishing: The covid-19 pandemic as credible social context for social engineering attacks", *Trames: A Journal of the Humanities and Social Sciences*, vol. 25, no. 4, pp. 371–393, 2021.
[23] EUR-Lex, *Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)*, 2016.
[24] M. Huber, S. Kowalski, M. Nohlberg, and S. Tjoa, "Towards automating social engineering using social networking sites", in *2009 International Conference on Computational Science and Engineering*, IEEE, vol. 3, 2009, pp. 117–124.
[25] EUR-Lex, *Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act)*, 2024.
[26] EUR-Lex, *Regulation (eu) 2023/2854 of the european parliament and of the council of 13 december 2023 on harmonised rules on fair access to and use of data and amending regulation (eu) 2017/2394 and directive (eu) 2020/1828 (data act)*, 2023.
[27] R. Goebel *et al.*, "Explainable ai: The new 42?", in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, 2018, pp. 295–303.
[28] Lockheed Martin Corporation, "The cyber kill chain", [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html (visited on 03/17/2025).
[29] imperva, "What is social engineering", [Online]. Available: https://www.imperva.com/learn/application-security/social-engineering-attack/ (visited on 03/17/2025).
[30] F. Mouton, M. Malan, L. Leenen, and H. S. Venter, "Social engineering attack framework", in *2014 Information Security for South Africa*, IEEE, 2014, pp. 1–9.
[31] M. Nohlberg and S. Kowalski, "The cycle of deception: A model of social engineering attacks, defenses and victims", in *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA)*, University of Plymouth, 2008.
[32] A. Cullen and L. Armitage, "The social engineering attack spiral (seas)", in *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*, IEEE, 2016, pp. 1–6.
[33] A. Algarni, Y. Xu, and T. Chan, "Social engineering in social networking sites: The art of impersonation", in *2014 IEEE International Conference on Services Computing*, IEEE, 2014, pp. 797–804.
[34] P. Pathak, "Is your soc overwhelmed? artificial intelligence and mitre att&ck can help lighten the load", [Online]. Available: https://securityintelligence.com/is-your-soc-overwhelmed-artificial-intelligence-and-mitre-attck-can-help-lighten-the-load/ (visited on 03/17/2025).
[35] M. S. Khan, S. Siddiqui, and K. Ferens, "A cognitive and concurrent cyber kill chain model", *Computer and Network Security Essentials*, pp. 585–602, 2018.
[36] J. Happa and G. Fairclough, "A model to facilitate discussions about cyber attacks", *Ethics and Policies for Cyber Operations: A NATO Cooperative Cyber Defence Centre of Excellence Initiative*, pp. 169–185, 2017.
[37] Offensive Security, "Google hacking database", [Online]. Available: https://support.google.com/websearch/answer/2466433 (visited on 03/17/2025).
[38] Google, "Programmable search engine", [Online]. Available: https://programmablesearchengine.google.com/about/ (visited on 03/17/2025).
[39] B. Singh, "One-stop cse for social media", [Online]. Available: https://cse.google.com/cse?cx=73dda67fd05b4405f#gsc.tab=0 (visited on 03/17/2025).
[40] S. Proto, "Stefanie proto's custom search engines", [Online]. Available: https://docs.google.com/spreadsheets/d/1fBPz6KHsFXryhu6JNrj1l-Rl04bEKLfinyCgCIqTyzU/edit?fbclid=IwAR3niqdKoD6Zx1DL4ZPvM9yXMUo8Hhr6zK2MpELKvyvegQ2Ea7xWqLELiM0#gid=436019663 (visited on 03/17/2025).
[41] S. Proto, "Sprp77's osint resources", [Online]. Available: https://drive.google.com/drive/folders/1CBcemFdorkAqJ-Sthsh67OVHgH4FQF05 (visited on 03/17/2025).
[42] Bellingcat, "Bellingcat - the home of online investigations", [Online]. Available: https://www.bellingcat.com/ (visited on 03/17/2025).
[43] Bellingcat, "Bellingcat's online investigation toolkit", [Online]. Available: https://docs.google.com/spreadsheets/d/18rtqh8EG2q1xBo2cLNyhIDuK9jrPGwYr9DI2UncoqJQ/edit?fbclid=IwAR2gnqA0CUujpYiS4Kg6Jvwch0Sg-tR1g9_s9gfofwRyIz75ioy-PzdQRto#gid=1919065780 (visited on 03/17/2025).
[44] Bellingcat, "Bellingcat osint landscape - start.me", [Online]. Available: https://start.me/p/ELXoK8/bellingcat-osint-landscape (visited on 03/17/2025).
[45] Technisette, "Tools - technisette website", [Online]. Available: https://technisette.com/p/tools (visited on 03/17/2025).
[46] 16.OSINT-IO, "16.osint-io", [Online]. Available: https://start.me/p/1kOJ9N/16osint-io (visited on 03/17/2025).
[47] Sourcesource, "Search social media", [Online]. Available: https://start.me/p/RMKeQv/search-social-media (visited on 03/17/2025).
[48] Ph055a, "Ph055a (ph055a) / repositories", [Online]. Available: https://github.com/Ph055a?tab=repositories (visited on 03/17/2025).
[49] J. Nordine, "Osint framework", [Online]. Available: https://osintframework.com/ (visited on 03/17/2025).
[50] KnowEm, "Checkusernames", [Online]. Available: https://checkusernames.com/ (visited on 03/17/2025).
[51] R. Tool, "Osint recon tool", [Online]. Available: https://recontool.org/%5C#mindmap (visited on 03/17/2025).
[52] HOPain, "Hopain osint search tools", [Online]. Available: https://osint.hopain.cyou/ (visited on 03/17/2025).
[53] HOPain, "Github - hopain complex osint search tools", [Online]. Available: https://github.com/HOPain/OSINT-Search-Tools (visited on 03/17/2025).
[54] BuiltWith Pty Ltd, "Builtwith technology lookup", [Online]. Available: https://builtwith.com/ (visited on 03/17/2025).
[55] S. Micallef, "Spiderfoot", [Online]. Available: https://www.spiderfoot.net/about/ (visited on 03/17/2025).
[56] S. Micallef, "Spiderfoot hx", [Online]. Available: https://sf-c824cc8.hx.spiderfoot.net/scaninfo?id=48f74883e9c61198ca1d8356ad0d38e9cc42584e208317542c21b56afbbbb890 (visited on 03/17/2025).
[57] Shodan Search Engine, "The shodan search engine", [Online]. Available: https://www.shodan.io (visited on 03/17/2025).

[58] KnownSec, “Zoomeye”, [Online]. Available: https://www.zoomeye.org (visited on 03/17/2025).
[59] SPYSE, “Spyse - internet assets search engine”, [Online]. Available: https://spyse.com (visited on 03/17/2025).
[60] projectdiscovery.io, “Projectdiscovery.io | chaos”, [Online]. Available: https://chaos.projectdiscovery.io (visited on 03/17/2025).
[61] M. Garciaguirre, “Synapsint”, [Online]. Available: https://synapsint.com/index.php (visited on 03/17/2025).
[62] Email-Format, “Email-format”, [Online]. Available: https://www.email-format.com (visited on 03/17/2025).
[63] Hunter Web Services, Inc., “Find email addresses in seconds - hunter”, [Online]. Available: https://hunter.io (visited on 03/17/2025).
[64] Kleissner Investments s.r.o., “Intelligencex”, [Online]. Available: https://intelx.io/ (visited on 03/17/2025).
[65] A. Agarwal, “Sleeping time”, [Online]. Available: http://sleepingtime.org/ (visited on 02/08/2024).
[66] R. Ahmad, “Whatsapp monitor - whatsapp contact online monitoring tool”, [Online]. Available: https://github.com/rizwansoaib/whatsapp-monitor (visited on 03/17/2025).
[67] Webmii, “Webmii people search engine”, [Online]. Available: https://webmii.com/ (visited on 03/17/2025).
[68] IDCrawl, “Idcrawl free people search”, [Online]. Available: https://www.idcrawl.com/ (visited on 03/17/2025).
[69] Maltego Technologies, “Maltego”, [Online]. Available: https://www.maltego.com (visited on 03/17/2025).
[70] M. Krueger, “Be careful what you osint with”, [Online]. Available: https://keyfindings.blog/2020/03/23/be-careful-what-you-osint-with/ (visited on 03/17/2025).
[71] DATA TOWER Kft., “Lampyre: Data analysis & osint tool for everyone”, [Online]. Available: https://lampyre.io (visited on 03/17/2025).
[72] m8r0wn, “Crosslinked: Linkedin enumeration tool to extract valid employee names from an organization through search engine scraping”, [Online]. Available: https://github.com/m8r0wn/crosslinked (visited on 03/17/2025).
[73] vijaysahuofficial, rly0nheart and HanslettTheDev, “Userrecon: This is a simple username recognition tool.”, [Online]. Available: https://github.com/vijaysahuofficial/UserReCon?%5C%5Cfbclid=IwAR0NAexz0KEyNDvJSOfSyOzsw9Z0Hc9j7AtB38ZK5AsI-5vupj46Dh95o-o (visited on 03/17/2025).
[74] [lucmski], “Recognition usernames in 187 social networks”, [Online]. Available: https://github.com/lucmski/userrecon-py (visited on 03/17/2025).
[75] thewhiteh4t, “Nexfil: Osint tool for finding profiles by username”, [Online]. Available: https://github.com/thewhiteh4t/nexfil?%5C%5Cfbclid=IwAR0NAexz0KEyNDvJSOfSyOzsw9Z0Hc9j7AtB38ZK5AsI-5vupj46Dh95o-o (visited on 03/17/2025).
[76] sherlock-project, “Hunt down social media accounts by username across social networks”, [Online]. Available: https://sherlock-project.github.io (visited on 03/17/2025).
[77] machine1337, “Userfinder: An osint tool to find user’s all over the internet including social media platforms”, [Online]. Available: https://github.com/machine1337/userfinder?%5C%5Cfbclid=IwAR3sCrgnkLvCUuLHP5VT6X8pVUvfyb8W0DZPenHVDA-VTIq3Et3zwMldWL0 (visited on 03/17/2025).
[78] rly0nheart, “Thorndyke: Lightweight username enumeration tool”, [Online]. Available: https://github.com/rly0nheart/thorndyke?%5C%5Cfbclid=IwAR1qnLkHJOC0a-OdlRXk1svN8ypAo6BvuQTrA8L5E4VY%5C%5CxbgI4UzVXLUz6PE (visited on 03/17/2025).
[79] sham00n, “Buster: An advanced tool for email reconnaissance”, [Online]. Available: https://github.com/sham00n/buster (visited on 03/17/2025).
[80] c. Martorella, “Theharvester | kali linux tools”, [Online]. Available: Https://www.kali.org/tools/theharvester/ (visited on 03/17/2025).
[81] E. Meged, “Raccoon: A high performance offensive security tool for reconnaissance and vulnerability scanning”, [Online]. Available: ttps://github.com/evyatarmeged/Raccoon (visited on 03/17/2025).
[82] aboul3la, “Sublist3r: Fast subdomains enumeration tool for penetration testers”, [Online]. Available: https://github.com/aboul3la/Sublist3r (visited on 03/17/2025).
[83] kpcyrd, “Sn0int: Semi-automatic osint framework and package manager”, [Online]. Available: https://github.com/kpcyrd/sn0int (visited on 03/17/2025).
[84] Iamthefroggy, “Frogy: Subdomain enumeration script”, [Online]. Available: https://github.com/iamthefrogy/frogy (visited on 03/17/2025).
[85] bhavsec, “Most advanced open source intelligence (osint) framework for scanning ip address, emails, websites, organizations”, [Online]. Available: https://github.com/bhavsec/reconspider (visited on 03/17/2025).
[86] G. Criscione, “Osintgram is a osint tool on instagram. it offers an interactive shell to perform analysis on instagram account of any users by its nickname”, [Online]. Available: https://github.com/Datalux/Osintgram (visited on 03/17/2025).
[87] Novitae, “Instagram osint tool to export and analyse followers | following with their details”, [Online]. Available: https://github.com/novitae/sterraxcyl (visited on 03/17/2025).
[88] J. Sánchez, “Instagram osint tool to scraping user information”, [Online]. Available: https://github.com/JavideSs/insta-extract (visited on 03/17/2025).
[89] L. Zaccagnini and falkensmz, “Tw1tter0s1nt: Python tool that automates the process of twitter osint investigation using twint”, [Online]. Available: https://github.com/falkensmz/tw1tter0s1nt (visited on 03/17/2025).
[90] V. Diaz, “The most complete open-source tool for twitter intelligence analysis”, [Online]. Available: https://github.com/vaguileradiaz/tinfoleak (visited on 03/17/2025).
[91] G. Rattaro *et al.*, “Tsurugi linux”, [Online]. Available: https://tsurugi-linux.org/index.php (visited on 03/17/2025).
[92] J. Martin *et al.*, “Csi linux”, [Online]. Available: https://csilinux.com/ (visited on 03/17/2025).
[93] D. Kitchen, “Usb-rubber-ducky wiki”, [Online]. Available: https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads (visited on 03/17/2025).
[94] D. Kennedy, “The social-engineerin toolkit (set)”, [Online]. Available: https://www.trustedsec.com/tools/the-social-engineer-toolkit-set (visited on 03/17/2025).
[95] M. Robinson and M. Nelson, “Generate-macro: This powershell script will generate a malicious microsoft office document with a specified payload and persistence method”, [Online]. Available: https://github.com/enigma0x3/Generate-Macro (visited on 03/17/2025).
[96] E. Nasi, G. Michel, and J. Goldberg, “Tool used to automatize obfuscation and generation of office documents, vb scripts, shortcuts, and other formats for pentest”, [Online]. Available: https://github.com/sevagas/macro%5C_pac (visited on 03/17/2025).
[97] IBM Watson Developer Cloud, “Tone analyzer demo”, [Online]. Available: https://tone-analyzer-demo.ng.bluemix.net/ (visited on 03/17/2025).
[98] IBM Watson Developer Cloud, “Sample node.js application for the ibm tone analyzer service”, [Online]. Available: https://github.com/watson-developer-cloud/tone-analyzer-nodejs (visited on 03/17/2025).
[99] Z. Coburn and G. Marra, “Realboy - believeable twitter bots”, [Online]. Available: http://ca.olin.edu/2008/realboy (visited on 03/17/2025).

[100] J. Seymour and P. Tully, “A machine learning based social media pen-testing tool”, [Online]. Available: https://github.com/zerofox-oss/SNAP%5C_R (visited on 03/17/2025).

[101] D. Kennedy, “The social-engineerin toolkit (set)”, [Online]. Available: https://www.trustedsec.com/tools/the-social-engineer-toolkit-set/ (visited on 03/17/2025).

[102] T. Rayat, “Zphisher: An automated phishing tool with 30+ templates.”, [Online]. Available: https://github.com/htr-tech/zphisher (visited on 03/17/2025).

[103] A. Kumar, “Phisheye is an ultimate phishing tool in python.”, [Online]. Available: https://github.com/sky9262/phishEye?%5C%5Cfbclid=IwAR1hdh%5C_rgxK24YB4gi%5C_2FYtY4D7Qrxt05WPwU2ZKGa1g%5C%5CXCh7ln7MF0RfmyI (visited on 03/17/2025).

[104] An0nUD4Y, “Blackeye: The ultimate phishing tool with 38 websites available”, [Online]. Available: https://github.com/An0nUD4Y/blackeye (visited on 03/17/2025).

[105] UndeadSec, “Socialfish: Phishing tool & information collector”, [Online]. Available: https://github.com/UndeadSec/SocialFish (visited on 03/17/2025).

[106] A. Moghaddas, “Storm-breaker: Tool social engineering with ngrok”, [Online]. Available: https://github.com/ultrasecurity/Storm-Breaker?%5C%5Cfbclid=IwAR2HX8B5RRQ2f-yRlWndAjxZM1PKfZxVZq-GM-9C%5C_f317IFWGjdAVhcRHaY (visited on 03/17/2025).

[107] Reis, “Phishious: Ein open-source-evaluierungs-toolkit für secure email gateway (seg)”, [Online]. Available: https://github.com/Rices/Phishious?%5C%5Cfbclid=IwAR2OhR2kRNkAyyGS7skSzOwlRPEWDcxzFwzohAFuj%5C_coi%5C%5CQFlMdq7t9wlh%5C_k (visited on 03/17/2025).

[108] Section9Labs, “Cartero - social engineering framework”, [Online]. Available: https://github.com/Section9Labs/Cartero (visited on 03/17/2025).

[109] smsranger.io, “Smsranger is the most advanced sms capture bot on the market”, [Online]. Available: https://smsranger.io/ (visited on 03/17/2025).

Graceful Degradation of Control Device Operation Under Attack

Rainer Falk, Christian Feist, and Steffen Fries
Siemens AG
Foundational Technologies
Munich, Germany
e-mail: {rainer.falk|christian.feist|steffen.fries}@siemens.com

***Abstract*—Cybersecurity includes preventing, detecting, and reacting to cyber-security attacks. Cyber resilience goes one step further and aims to maintain essential functions even during ongoing attacks, allowing to deliver an intended service or to operate a technical process, and to recover quickly back to regular operation. During an ongoing attack, the impact on the overall system operation is limited if the attacked system stays operational, maybe with degraded performance or functionality. Control devices of a cyber physical system monitor and control a technical process. This paper describes a concept for a control device that reduces its operation depending on the current threat landscape, maintaining its basic and essential functionalities. If attacks have been detected, or if relevant vulnerabilities have been identified, the functionality is increasingly limited, thereby reducing the attack surface in risky situations, while allowing the device and the cyber physical system to stay operational.**

Keywords–cyber resilience; cyber physical system; industrial security; cybersecurity.

I. INTRODUCTION

A Cyber Physical System (CPS), e.g., an industrial automation and control system, contains control devices that interact with the real, physical world using sensors and actuators. They implement the functionality to control and monitor the operations in the physical world, e.g., a production system or a power automation system. A control device can be a physical device, e.g., an industrial Internet of Things (IoT) device, an electronic control unit, a Programmable Logic Controller (PLC), or a virtualized control device, e.g., a container or virtual machine executed on a compute platform. Control devices communicate via data networks to exchange control commands and to monitor the CPS operation to realize different automation use cases. These use cases may comprise predictive maintenance or the reconfiguration of control devices for flexible automation and for optimizing operational systems (Industry 4.0), or specific line protection features in power system operation. The connectivity of control devices is thereby increasingly extended towards enterprise networks and towards cloud-based services, increasing the exposure towards attacks originating from external networks or the Internet [2].

Being resilient means to be able to withstand or recover quickly from difficult conditions [3][4]. The Cybersecurity puts the focus on preventing, detecting, and reacting to cyber-security attacks. With cyber resilience, the scope is extended to the aspect to continue to deliver an intended outcome despite an ongoing cyber attack, and also to recover quickly back to regular operation. When an attack is carried out, the impact on the overall system operation is limited if the attacked system stays operational, even with degraded performance or functionality. Even during attacks, intended services can still be provided, at least in a limited way.

This paper, as an extended version of [1], describes a concept for a control device that can adapt to a changing threat landscape by adapting and limiting its provided functionality. If attacks have been detected, or if relevant vulnerabilities have been identified, devices can limit their functionality increasingly towards only basic and essential functions, thereby reducing their attack surface in risky situations. Basic and essential functions refer to the main functionality of a device that contribute to the intended operational use case and the embedding operational environment. This paper extends [1] by giving an overview on industrial CPS and their cybersecurity, and by describing the concept of a resilience engine, an isolated execution environment ensuring that the resilience functionality is executed in a trustworthy way even if the main functionality of the control device has been manipulated. Furthermore, the evaluation section has been extended.

The remainder of the paper is structured as follows: Section II gives an overview on related work, Section III on industrial CPS, and Section IV on their integrity protection. Section V describes the concept of graceful degradation under attack, and Section VII presents a usage example in industrial automation systems. Section VIII provides an evaluation of the presented approach from different perspectives relevant for an industrial application. Section IX concludes the paper and gives an outlook towards future work.

II. RELATED WORK

Cybersecurity requirements for Industrial Automation and Control Systems (IACS) are defined in the standard series IEC62443 [5]. This series provides a holistic security framework as a set of standards defining security requirements for the development process and the operation of IACS, as well as technical cybersecurity requirements on automation systems and the used components. IEC62443 requires that the IACS security measures do not cause a loss of essential services and functions, i.e., essential functions

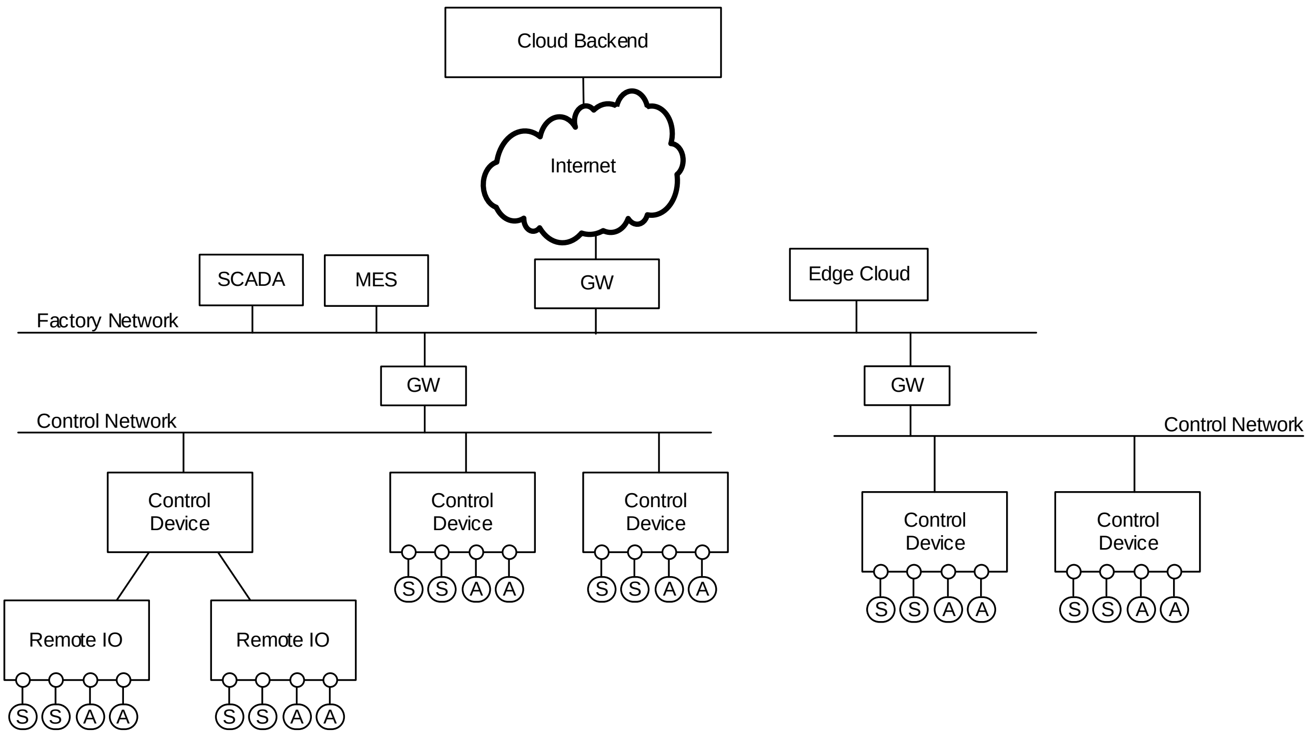


Figure 1. Industrial Cyber Physical System.

have to be kept operational in a degraded operation mode. A main objective is that deployed cybersecurity measures do not negatively impact the availability of the IACS operations. An essential function is defined as a "function or capability that is required to maintain health, safety, the environment and availability for the equipment under control". Essential functions have to be maintained also during denial-of-service attacks, or if a zone boundary protection, e.g., a network firewall, activates an island mode with limited or no connectivity. Further requirements address backups of the configuration of IACS devices, allowing to restore configurations, and the recovery and reconstitution to a known secure state after an incident.

Cyber resilience in the broader meaning to keep systems operational under attack and to recover quickly gets increasing attention, as can be seen by recent security standards and the regulation of the European Cyber Resilience Act (CRA) [6] and the Delegated Regulation for the Radio Equipment Directive (RED) [7]. The regulation of the Cyber Resilience Act (CRA) [6] includes in Annex I the requirement to maintain essential and basic functions under attack ("protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks"). The development of corresponding standards addressing CRA regulative requirements has just started. The NIST Cybersecurity Framework (CSF) 2.0 [11] gives general guidance on managing risk, addresses resilience for normal and adverse situations. The standard NIST SP800-193 [8] describes technology-independent guidelines for resilience of platform firmware. Resilience-specific roots of trust are defined for update of platform firmware, for detection of a corrupted firmware, and for recovery from a compromised platform state. England et al. give a high-level overview of the Cyber Resilient Platforms Program (CyReP) [10], describing hardware and software components addressing NIST SP800-193 requirements. A working group on "cyber resilient technologies" of the Trusted Computing Group (TCG) is working on technologies to enhance cyber resilience of connected systems. Here, different building blocks for cyber resilient platforms have been described that allow to recover from a malfunction reliably back into a well-defined operational state [9]. Such building blocks support cyber resilience as they allow to recover quickly and with reasonable effort from a manipulated state. Basic building blocks are a secure execution environment for the resilience engine on a device, protection latches to protect access to persistent storage of the resilience engine even of a compromised device, and watchdog timers to ensure that the resilience engine can in fact perform a recovery. A further standard, ETSI EN 303 645 [12], describes specific security requirements for the consumer IoT device domain, addressing also resilience by the requirement to "remain operating and locally functional in the case of a loss of network access".

III. Industrial Cyber Physical Systems

An industrial CPS, i.e., an IACS, monitors and controls a technical system. Examples are process automation, factory automation, production machines, building automation, energy automation, and cloud robotics. Figure 1 shows an example of an IACS, comprising different control networks connected to a factory network and a cloud backend system. Sensors (S) and actuators (A) of a technical system are connected with control devices directly or via remote input/output (IO) modules. The technical process is controlled by measuring its current state using the sensors, and by determining the corresponding actuator signals. Separation of the network by gateways (GW) is used to realize distinct control networks with strict real-time requirements for the interaction between sensors and actuators of a production cell, or to enforce a specific security policy within a production cell. A Supervisory Control and Data Acquisition (SCADA) system allows operators to monitor and influence the technical operation, and a Manufacturing Execution System (MES) can be used to plan, track, and document manufacturing steps.

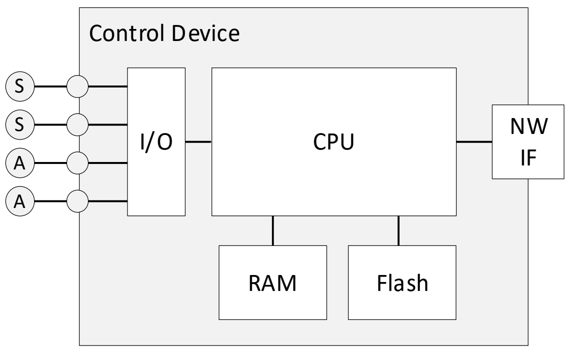


Figure 2. Control Device.

Figure 2 shows the typical structure of automation components that monitor and control the physical world using sensors and actuators. The monitoring and control functionality is defined by its firmware/software that is executed on a central processing unit (CPU) and the corresponding configuration data, both stored in non-volatile memory (Flash). A network interface (NW IF) allows communication with other devices, e.g., via Ethernet or via wireless communications as wireless local area network (WLAN) or a private 5th generation (5G) mobile communication system.

In a CPS, the impact of an attack in the OT system may not only affect data and data processing as in classical IT, but it may have an effect also on the physical world. For example, production equipment could be damaged, or the physical process may operate outside the designed physical boundaries, so that the produced goods may not have the expected quality, or even safety-related requirements could be affected.

IV. CPS CYBERSECURITY

Cybersecurity mechanisms have been known for many years and are applied in smart devices (Internet of Things, Cyber Physical Systems, industrial and energy automation systems, operation technology). Such mechanisms target source authentication, system and communication integrity, and confidentiality of data in transit or at rest.

A. *Industrial Security*

Protecting IACS against intentional attacks is demanded by operators to ensure a reliable operation, and also by regulation. The main relevant industrial security standard that describes security from a holistic perspective is IEC 62443 [3]. Security requirements defined by the industrial security standard IEC 62443 range from security processes during development and operation of devices and systems, personal and physical security, device security, network security, and application security, addressing the device manufacturer, the integrator as well as the operator of the IACS.

Industrial security is also called Operation Technology (OT) security, to distinguish it from general IT security. Industrial systems have different security priorities and requirements compared to common IT systems. Typically, availability and integrity of an automation system have higher priority than confidentiality. Specific requirements and side conditions of industrial automation systems like high availability, planned configuration (engineering info), scheduled maintenance windows, long life cycles, unattended operation, real-time operation, and communication, as well as safety requirements have to be considered when designing an OT security solution.

B. *Control Device Integrity*

The objective of device integrity is to ensure that a single device is not manipulated in an unauthorized way, ensuring that it operates as genuine device. Device integrity is highly relevant for industrial control devices to ensure their reliable operation.

Integrity protection includes the integrity of the device firmware, the integrity of the device configuration, but also its physical integrity. The main technologies to protect device integrity are:

- Secure boot: A device loads at start-up only unmodified, authorized firmware. Typically, a device verifies the digital signature of loaded firmware before executing it.
- Measured boot: The loaded software modules are checked at the time they are loaded. Usually, a cryptographic hash value is recorded in a platform configuration register of a hardware or firmware Trusted Platform Module (TPM). The configuration information can be used to grant access to keys, or it can be attested towards third parties.
- Protected firmware update: When the firmware of a device is updated, the integrity and authenticity of the firmware update is checked. The firmware update image can be digitally signed.
- Application whitelisting: Only allowed, known applications can be started on a device. A whitelist defines which application binaries can be started.
- Runtime integrity checks: During operation, the device performs a self-test of security functionality and integrity checks to verify whether it is operating as expected. Integrity checks can verify the integrity of files, configuration data, software modules, and runtime data

as the process list, i.e., the list of currently executed processes.

- Process isolation, kernel-based Mandatory Access Control (MAC): Hypervisors, OS-level virtualization such as containers, or kernel-based MAC systems can be used to isolate different classes of software (security domains). An attack or malfunction of one security domain does not affect other security domains on the same device.
- Tamper evidence, tamper protection: The physical integrity of a device can be protected, e.g., by security seals or by tamper sensors that detect opening or manipulation of the housing.
- Device integrity self-test: A device performs a self-test to detect failures. The self-test is performed typically during startup and is repeated regularly during operation.
- Operation integrity checks: Measurements on the device can be compared with the expected behavior in the operative environment. An example is the measurement of connection attempts to/from the device, based on parameters of a Management Information Base (MIB).

These technologies protect the device integrity, ensuring that the device's control functionality operates as designed, and to detect manipulations. Device resilience technologies are needed on top to support a reliable operation during attacks and to recover quickly.

C. Cyber Physical System Integrity Monitoring

Integrity does not only affect single devices, but also the overall system level comprising a set of interconnected devices. The main approaches to protect system integrity are collecting and analyzing information at system level:

- Centralized Logging: Devices provide log data, e.g., using Open Platform Communication Unified Architecture (OPC UA) protocol, Simple Network Management Protocol (SNMP), or syslog protocol, to a centralized logging system for further analysis. This may be done in a Security Information and Event Management (SIEM) System and lead to reactions on identified cybersecurity events.
- Runtime device integrity measurements: A device integrity agent provides information gathered during the operation of the device (see also subsection B above). It collects integrity information on the device and provides it for further analysis. Basic integrity information includes the results of a device self-test, and information on the current device configuration (firmware version, patches, installed applications, configuration). Furthermore, runtime information can be gathered and provided for analysis (e.g., process list, file system integrity check values, partial copy of memory).
- Network monitoring: The network communication is intercepted, e.g., using a network tap or a mirror port of a network switch.

The captured integrity information can be used for system runtime integrity monitoring to detect integrity violations in in a timely manner. Operators can be informed, or actions can be triggered automatically. Furthermore, the information is archived for later investigations. This allows that integrity violations can be detected also later with a high probability, so that corresponding countermeasures can be initiated (e.g., plan for an additional quality check of produced goods).

An intelligent analysis platform performs data analysis (e.g., statistical analysis, big data analysis, artificial intelligence) and triggers suitable respondence actions (e.g., alarm, remote wipe of a device, revocation of a device, stop of a production site, planning for additional test of manufactured goods).

D. Resilience Under Attack

In a cyber physical environment, a main objective is that the CPS stays operational and that its integrity is ensured. In the context of an industrial automation and control system, that means that intended actions of the system in the physical world continue to take place even when the automation and control system of the CPS is attacked successfully. Risk management, the established approach to cyber security, identifies threats and determines the risk depending on probability and impact of a potential attack. The objective is to put the focus of defined security measures on the most relevant risks, reducing the probability that a successful attack takes place, and reducing the impact of successful attacks, e.g., by detecting successful attacks by security monitoring allowing to react, e.g., by shutting down a CPS.

Resilience, however, puts the focus on a reduction of the impact of successful attacks, where the system can stay operational with a degraded performance or functionality, and to recover quickly from a successful attack.

Being resilient means to be able to withstand or recover quickly from difficult conditions [12]. It shifts the focus of "classical" IT and OT security, which put the focus on preventing, detecting, and reacting to cyber-security attacks, to the aspect to continue to deliver an intended outcome despite an adverse cyber attack taking place, and to recover quickly back to regular operation. More specifically, resilience of a system is the property to be resistant to a range of threats and withstand the effects of a partial loss of capability, and to recover and resume its provision of service with the minimum reasonable loss of performance.

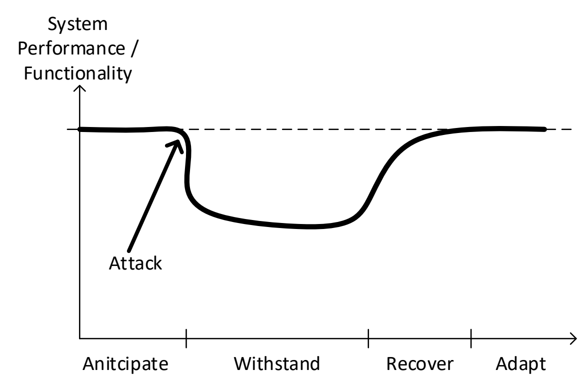


Figure 3. Concept of Cyber Resilience.

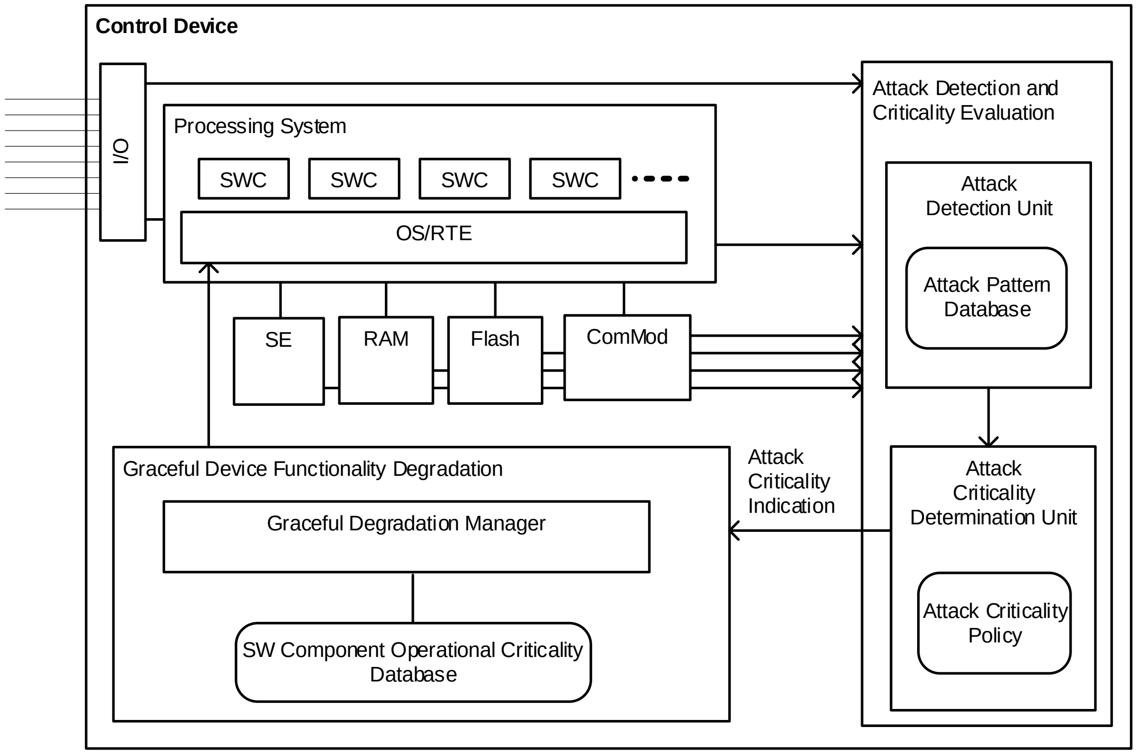


Figure 4. Control Device with graceful degradation under attack.

Figure 3 illustrates the concept of cyber resilience: When an attack is ongoing, the impact on the CPS operation is limited as basic and essential functionality is maintained in a reliable way. The effects of an attack on the CPS operation are "absorbed", so that the CPS can stay operational, but with limited performance or functionality. In particular, it can be avoided that the CPS has to be shut down completely. A recovery takes place to bring the system up to the regular operation in a fast and trustworthy way.

V. Control Device with graceful degradation under attack

Control devices of a cyber physical system monitor and control a technical process via sensors and actuators. The proposed enhanced control device can adapt to a changing threat landscape by adapting and limiting its functionality depending on the current threat landscape. If attacks have been detected, or if relevant vulnerabilities have been identified, the functionality of the device is increasingly limited towards essential functions. This graceful degradation under attack reduces the attack surface in risky situations, while maintaining essential functions of the device. This allows the cyber physical system, in which the control device is deployed, to stay operational even during attack.

Figure 4 shows the concept of a control device that is designed for graceful degradation under attack. The main functionality of the device is realized on its processing system by multiple SoftWare Components (SWC) that are executed by an Operating System (OS) and/or an app RunTime Environment (RTE). Software components may, e.g., implement the control function and diagnostic functions. The components interact with the physical world via sensors and actuators that are connected via an Input/Output (I/O) interface. The processing system uses a Secure Element (SE) for secure key storage and cryptographic operations, a Random Access Memory (RAM), a flash memory, and a Communication Module (ComMod).

An attack detection and criticality evaluation module monitors the operation of these device components to detect unexpected device behavior, here by matching the detected monitoring events with an attack pattern database. It would also be possible to check the device monitoring data against reference states providing the expected behavior. Such a check could be done against static reference data, but could also be done in conjunction with a digital twin, providing a simulation of the ongoing process. If a suspicious device behavior is detected, a criticality is determined, and depending on that, the functionality of the device is adapted by the Graceful device functionality Degradation Manager (GDM). For example, a SWC implementing a simplified control function with reduced functionality can be activated instead of the regular control function, reducing the threat exposure.

This example shows a self-contained realization in which the attack detection and graceful degradation functionality is realized as part of the device. A distributed implementation involving also device-external components would be possible as well, but would require tight protection of all external interfaces to ensure a reliable operation even during ongoing attacks.

In industrial automation, the control functionality is usually not fixed, but is commissioned by the automation system operator, a machine builder, or an integrator. For this application domain, the need is therefore foreseen to allow also commissioning of the graceful degradation functionality of a control devices, allowing to define the device resilience behavior under attack. This specifically relates to the definition of essential functions, depending on the application use case.

VI. Resilience Engine

An isolated execution environment, a resilience engine, is needed to ensure that resilience functionality is executed in a trustworthy way even if the main functionality of the control device is manipulated.

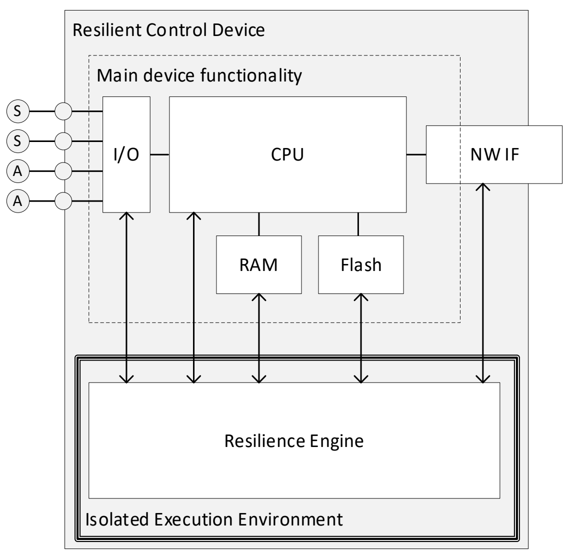


Figure 5. Control Device with Resilience Engine

Figure 5 shows a resilient control device that includes a resilience engine for the resilience functionality. The resilience engine can monitor and restrict the control device operation, in particular the processing unit (CPU), the IO operations, and the network communications. The resilience engine is isolated from the regular control function of the control device to ensure that is in a trustworthy state even if the main device functionality has been attacked successfully. Various realization options can be followed:

- Separate security chip or crypto controller with tamper protection.
- Integrated circuit (system on chip) with separate security core. The security core may implement specific tamper protection measures.
- Isolation on a regular processing core using hardware-support, e.g., by a trusted execution environment (TEE).
- Isolation using a software-based hypervisor executed below the operating system on the main processor.
- Isolation using operating systems means.

These approaches differ concerning the robustness of isolation, but also concerning their implementation overhead. It is a design decision, based on threat and risk analysis, to balance implementation robustness with implementation effort. Besides isolation, the resilience engine has to be protected by cybersecurity measures, e.g., secure firmware update and remote integrity attestation. Dedicated cryptographic keys for protecting the resilience engine can be used, to ensure that the cryptographic protection measures of the resilience engine are independent of the protection measures of the main device functionality.

VII. Usage Example

This section describes the usage in an exemplary way, distinguishing software components of varying criticality from the perspective of maintaining the CPS operation under attack.

Figure 6 shows example software components that are grouped according to the operational criticality. The graceful degradation manager activates the software components of the respective functionality group depending on the current attack scenario. In this example, three sets of software components are defined, defining the software components that are active in full, reduced, and in minimum functionality mode.

To ensure cyber resilience, the functionality is reduced to a limited control functionality that can be less optimized and lead to reduced CPS performance, and to keep limited remote access. In more critical attack scenarios, a fail-safe operation mode is activated, i.e., if even the reduced functionality operation cannot be ensured reliably.

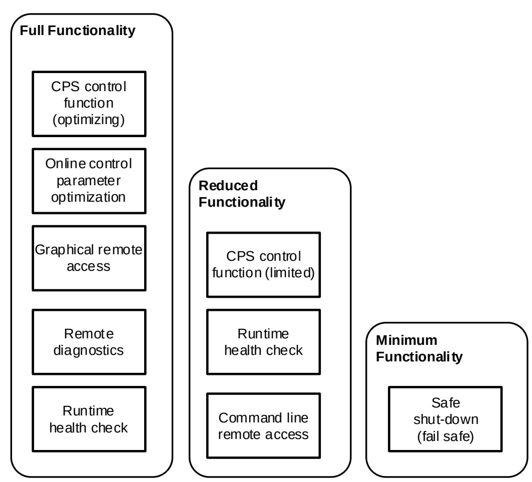


Figure 6. Software components with different operational criticality.

As an example from an industrial application use case, a protection device of a substation of an energy automation system may be considered as control device. Protection devices are applied within electric power systems to detect abnormal and intolerable electrical conditions and to initiate appropriate corrective actions, e.g., to interrupt a power line. The software executed on the protection device that implements the control functionality could be attacked via the network interface. In the extreme case, the network interface may be switched off for a limited time by the GDM, keeping the protection functionality based on local sensor readings and connected actuators. That way, the protection device will not communicate its measurements to other substation devices in the substation anymore, but it would retain the local protection functionality and thus the safety of the connected power line.

VIII. Evaluation

This section gives a preliminary evaluation of the presented concept from different perspectives.

CPS availability perspective: Availability and the flexibility to adapt to changing production requirements are important requirements for OT operators [6]. The proposed approach allows to maintain CPS operation in a limited way even under ongoing attacks or in specific failure situations. A reliable CPS operation can be maintained, avoiding the need to shutdown the CPS operation completely. This is considered to be the main advantage of enhanced control device resiliency with graceful degradation under attack, as the availability of the CPS is improved.

CPS operational performance perspective: The limited function mode may lead to a reduced productivity and less efficiency of the CPS. The exact impact depends on the limitations of the limited control operation functionality.

CPS management perspective: The operator of the CPS has to be aware about the resilience functionality supported by the CPS control devices. The CPS operator has to be made aware if some control devices have activated a restricted resilience operation mode, so that the overall CPS operation and the production planning can be adapted accordingly. The CPS' operation concept has to be defined accordingly to address restricted resilience operation modes, and the operating personnel has to be trained for the resilience functionality.

Implementation perspective: Control devices have to implement the functionality for attack detection and resilience management / graceful degradation in a highly protected execution environment that can be relied upon even if the main processing system of the control device should be attacked. The overhead depends on the specific technical implementation approach, e.g., requiring an additional protected hardware component, e.g., a secure microcontroller or a secured Field Programmable Logic Controller (FPGA). Both development effort and hardware costs are increased, which would have an impact in particular for cost-optimized control devices. Also, SCADA and MES systems have likely to be extended to allow operational personnel and production processes to be adapted if a control device activates a restricted resilience mode.

Engineering perspective: The graceful degradation functionality (attack criticality determination, as well as the definition of use case specific essential functions) has to be planned and defined so that it can be commissioned on the control device, leading to additional commissioning effort. It may be required that the same functionality has to be realized in different versions, e.g., in fully flexible, optimized operation mode and a limited operation mode. These modes have to be tested and validated, e.g., using simulations. Blueprints that give practice-proven engineering examples can limit the required additional engineering effort.

Testing perspective: The graceful degradation functionality has to be tested carefully to ensure that relevant attack scenarios are reliably detected, and also to validate that the limited control operation mode is reliably activated and performs reliably even under the detected attack scenarios. Testing has to be performed both on device-level for a single control device, as well as on system level for a CPS that uses multiple control devices, where some may be enhanced with graceful degradation under attack. As testing attack scenarios in real-world operational systems is often not possible, simulation tools are essential that allow simulating the CPS operation realistically under various attack scenarios when the engineered graceful degradation functionality is in place. Testing can be performed not only during the planning and engineering phase, but also during regular CPS operation to test the impact of recent attacks. Simulation may be useful also for training operational personnel.

Overall, implementing, engineering, and testing graceful degradation under attack implies additional effort that has to be justified by the increased availability of the CPS. The benefit depends on the attacks observed in real-world operations. Simulation tools (like digital twins) can be used also for this purpose to determine key performance indicators of the real-world CPS for which resilience under attack is protected with control devices implementing the engineered graceful degradation functionality and comparing it with a simulated CPS using control devices *not* implementing the engineered graceful degradation functionality.

IX. Conclusion and Future Work

The proposed concept for cyber resilient control devices can enhance CPS availability even under ongoing attack scenarios. However, it comes with relevant additional effort for implementation, engineering, testing, training, and with overhead for the trusted execution environment required for resilience functionality that requires besides hardware support also specific security-focused implementation effort. However, cyber resilience requirements and technologies are increasingly defined in cybersecurity standards and regulations, and are adopted in real-world solutions, e.g., for server systems in data centers [13]. The specific robustness properties and the implementation effort of different technical approaches to implement a resilience engine on embedded control devices have still to be investigated.

The additional effort needed for implementing cyber resilience for control devices has to be justified by the positive impact on CPS operation, allowing to maintain a reliable CPS operation during ongoing attacks. The CPS operation may relate to a business model focusing on providing a continuous service like energy provisioning or may focus on the preservation of a safety function, like the availability of a protection system. Simulation tools for CPS and their control devices allow investigating cyber resilience for CPS in both the planning and operation phases, reducing in particular the testing effort, and allowing to analyze the effectiveness for different types of attack. A further direction addresses robustness under attack that tries to keep the CPS operational under attack with minimal or even no reduction of the systems operational performance, i.e., to withstand attacks.

REFERENCES

[1] R. Falk, C. P. Feist, and S. Fries, “Graceful Degradation under Attack: Adapting Control Device Operation Depending on the Current Threat Exposure”, International Conference on Cyber-Technologies and Cyber-Systems, CYBER2024, September 29, 2024 to October 3, 2024, Venice, Italy, pp. 9-12, 2024. [Online]. Available from: https://www.thinkmind.org/library/CYBER/CYBER_2024/cyber_2024_1_20_80023.html 2025.05.05

[2] Platform Industrie 4.0, “Resilience in the Context of Industrie 4.0”, Whitepaper, April 2022. [Online]. Available from: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Resilience.html 2025.05.05

[3] R. Falk and S. Fries, “Enhanced Attack Resilience within Cyber Physical Systems”, Journal on Advances in Security, vol 16, no 1&2, pp. 1-11, 2023. [Online]. Available from: https://www.iariajournals.org/security/sec_v16_n12_2023_paged.pdf 2025.05.05

[4] R. Falk and S. Fries, “System Integrity Monitoring for Industrial Cyber Physical Systems”, Journal on Advances in Security, vol 11, no 1&2, July 2018, pp. 170-179. [Online]. Available from: www.iariajournals.org/security/sec_v11_n12_2018_paged.pdf 2025.05.05

[5] IEC 62443, “Industrial Automation and Control System Security” (formerly ISA99). [Online]. Available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx 2025.05.05

[6] “Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act), Document 32024R2847, November 2024. [Online]. Available from: http://data.europa.eu/eli/reg/2024/2847/oj 2025.05.05

[7] “Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC Text with EEA relevance”, 10/2023. [Online]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0053 2025.05.05

[8] A. Regenscheid, “Platform Firmware Resiliency Guidelines”, NIST SP 800-193, May, 2018. [Online]. Available from: https://csrc.nist.gov/publications/detail/sp/800-193/final 2025.05.05

[9] TCG, “Cyber Resilient Module and Building Block Requirements”, V1.0, October 19, 2021. [Online]. Available from: https://trustedcomputinggroup.org/wp-content/uploads/TCG_CyRes_CRMBBReqs_v1_r08_13jan2021.pdf 2025.05.05

[10] P. England et al., “Cyber resilient platforms”, Microsoft Technical Report MSR-TR-2017-40, September, 2017. [Online]. Available from: https://www.microsoft.com/en-us/research/publication/cyber-resilient-platforms-overview/ 2025.05.05

[11] NIST CSF, “The NIST Cybersecurity Framework (CSF) 2.0”, February, 2024. [Online]. Available from: https://doi.org/10.6028/NIST.CSWP.29 2025.05.05

[12] EN 303 645, “Cyber Security for Consumer Internet of Things: Baseline Requirements”, ETSI, V3.1.3 (2024-09), September, 2024. [Online]. Available from: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/03.01.03_60/en_303645v030103p.pdf 2025.05.05

[13] Intel Data Center Block with Firmware Resilience, Solution Brief. [Online]. Available from: https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/firmware-resilience-blocks-solution-brief.pdf 2025.05.05

[14] D. Bodeau and R. Graubart, “Cyber resiliency design principles”, MITRE Technical Report, January 2017, [Online]. Available from: https://www.mitre.org/sites/default/files/publications/PR%2017-0103%20Cyber%20Resiliency%20Design%20Principles%20MTR17001.pdf 2025.05.05

Post-Quantum Cryptography - An Overview of Standards, Protocols, and Practical Applications

Jörn-Marc Schmidt
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: joern-marc.schmidt@iu.org

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: alexander.lawall@iu.org

***Abstract*—In cryptographic security, quantum computing poses a significant challenge to traditional cryptographic protocols. This study investigates the landscape of Post-Quantum Cryptography (PQC), focusing on the transition from theoretical underpinnings, over standardization efforts to practical implementations. The primary research question that guides this contribution is: What mechanisms can be implemented to safeguard applications and what efforts are under way by application providers and technology platforms? This question is answered by the current state of standards supporting PQC and the ongoing preparation efforts. Thereby, not only the standards for cryptographic algorithms, but also the protocols relying on them are considered. Furthermore, the status of (open-source) implementations is reviewed and roadmaps from companies / technology providers are discussed. Hence, this paper does not only discuss what a company can do to protect their applications but also takes the viewpoint of an end-user regarding the support of applications.**

Keywords-Post Quantum Cryptography (PQC); PQC Standards; PQC Implementations; PQC Libraries; Technology Roadmaps.

I. Introduction

This work is an extended version of *Theoretical and Practical Aspects in Identifying Gaps and Preparing for Post-Quantum Cryptography*, published at SECURWARE 2024 [1].

Quantum computers will influence many fields. They will improve biological and chemical simulations, can be applied for risk modeling, and improve solving of optimization problems. In addition to those constructive improvements, they have the potential to impact the security of cryptographic algorithms. Especially, asymmetric algorithms that rely on the hardness of factorization or the discrete logarithm problem cannot be considered secure when a Cryptographic Relevant Quantum Computer (CRQC) is available. Hence, use cases relying on such algorithms will be impacted by CRQCs. Moreover, even data transmitted today can be endangered by attackers recording the transmission and decrypting it as soon as CRQCs are available. This is referred to as harvest now and decrypt later attack.

Quantum computers also have the potential to impact symmetric cryptography. However, it is assumed that such attacks can be addressed by using larger keys for symmetric encryption [2]. For example, the National Security Agency (NSA) states that the Advanced Encryption Standard (AES) can be considered secure when used with 256-bit keys [3]. Hence, the remainder of this paper will focus on asymmetric cryptography.

This paper is organized as follows: It starts with a brief discussion of the current state of the art in Section II. Section III discusses the general preparation process and security protocols. Section IV summarizes the status of the standardization of new cryptographic algorithms, while Section V looks into the status of protocol standards. Libraries that support Post-Quantum Cryptography (PQC) algorithms, as a foundation for implementations, are presented in Section VI. Section VII discusses applications that are available for endusers and Section VIII discusses the possibilities to use quantum mechanisms to improve security. Finally, conclusions are drawn in Section IX.

II. State of the Art

The challenge of ensuring Post-Quantum (PQ) security is already picked up by security researchers, developers, several government agencies, and companies. In order to drive the readiness of post-quantum cryptographic algorithms and their adoption in standard applications forward, many activities are underway. They include various working groups, like the Internet Engineering Task Force (IETF) working group *Post-Quantum Use In Protocols* [4], and the European Telecommunications Standards Institute (ETSI) *Quantum-Safe Cryptography (QSC)* working group [5]. Further activities are driven by various companies like Google [6], IBM [7], Microsoft [8], and Utimaco [9].

This paper provides an overview of those activities. Thereby, its focus is on use cases for asymmetric cryptography due to the expected high impact of CRQC on this type of algorithm. The paper highlights the status of standardization processes and the production-readiness of implementations. As such, it demonstrates what is currently done by different organizations, and gives guidance on what can be done today to protect own applications and data.

III. Use Cases of Cryptography

The transition to post-quantum cryptography, given the widespread use of the algorithms, is a huge undertaking. As a first step, it is important to understand where susceptible algorithms are employed and how valuable the protected data is. Hence, for a company to prepare, a risk assessment of its application portfolio is required. The first step in such an endeavor is creating a cryptographic inventory, providing insights on where algorithms and protocols are used, together with related parameters. Various tools can help creating an inventory [10].

Afterwards, a sound risk model that integrates into the company's risk management procedures is required. For the financial industry, for example, the Financial Services Information Sharing and Analysis Center (FS-ISAC) provides a white paper on modeling the risk [11]. This helps to create a profound strategy and to decide where the highest risks and the biggest benefits are expected. Finally, a maturity index helps judging and comparing where a company is on its journey to post-quantum security [12][13].

Generally speaking, data requires protection at rest, in transit, and in use.

Data at rest commonly relies on symmetric cryptography, where limited impact of quantum computers is expected. Solutions that employ asymmetric cryptography can make use of Key Encapsulation Mechanisms (KEMs) discussed in Section IV.

Encryption of data in use is not yet widely used. An available possibility is to rely on processor extensions like Intel Software Guard Extensions (SGX) [14] / Trust Domain Extensions (TDX) [15] or AMD Secure Encrypted Virtualization (SEV) [16]. Especially the attestation, i.e., proving that the protected environment is in a trustworthy state, relies on asymmetric cryptography. Solutions are discussed in [17].

In particular, when focusing on harvest now and decrypt later attack scenarios, security of encryption in transit against attacks with quantum computers is the most pressing scenario. In order to protect data in transit, it is possible to

- protect the underlying infrastructure by ensuring that the communication is PQ-secure. While this has large impact, it is restricted to endpoints that are in direct control; protecting the connections to end-users might not be possible. Commonly, protocols like IPsec and Media Access Control security (MACsec) are employed in such scenarios.
- ensure that the communication protocols are PQ-secure. Common protocols are Transport Layer Security (TLS) and Secure Shell (SSH). Both protocols allow to negotiate the used ciphers with a handshake. This enables using PQC whenever both parties support it without preventing non-PQC-secure communication in case one endpoint is not able to use such a cipher.
- encrypt the transferred message in a quantum-secure way. By using a method that ensures that the data is encapsulated with post-quantum cryptography, sound protection against adversaries can be achieved. This can be done either via standards suitable to the application, like Secure/Multipurpose Internet Mail Extensions (S/MIME) for emails/webpages, JavaScript Object (JSON) Signing and Encryption (JOSE)/Concise Binary Object Representation (CBOR) Object Signing and Encryption (COSE) for messages between applications and Pretty Good Privacy (PGP) for encrypting arbitrary data including files. Another option is to rely on self-defined, custom protocols, e.g., by employing implementations discussed in Section VI directly.
- rely on plattforms and services that use PQC for protecting data or at least have a clear roadmap regarding PQC-migration. Especially in cloud environments, it is not always required that platform-users implement security mechanisms themselves, but it is possible to rely on services delivered by the could provider.

A common requirement that is independent of the layer where data protection is applied, is ensuring as sound authentication of the communication partners and the authenticity of the data. Related methods are required as soon as a CRQC is available. Collecting data today, as in the harvest now and decrypt later scenario, does not represent a current threat. However, a lack of being ready in time will have devastating consequences as well, as an adversary can impersonate every identity that is not protected and forge any non-PQC signature. While details of different protocols on how to achieve a secure authentication vary, many make use of certificates issued by a Public Key Infrastructure (PKI). In essence, issuing a PQ-secure certificate requires a PQ-secure signature algorithm. However, there are many processes around a secure PKI and different ways of integrating a PQ-secure signature into a certificate. As it can be the foundation for critical processes like TLS authentication, for re-signing documents, including contracts, and for secure authentication of devices, it is a critical aspect of the PQ-migration as well.

The different options that are discussed in Section V are shown in Figure 1. In addition, a few examples for end-user applications are discussed in Section VII.

IV. The Quest for new Cryptographic Algorithms

The basis of all protocols and building blocks is quantum-secure algorithms. Hence, it is essential to develop and standardize new (asymmetric) cryptographic algorithms to replace the current ones.

A key activity in this regard was launched by National Institute of Standards and Technology (NIST) end of 2016. The NIST issued a call for papers for new post-quantum cryptographic algorithms [18]. Out of 69 initial submissions, three were selected to become Federal Information Processing Standards (FIPS). The following documents have recently (at the time writing this paper) been finalized:

- FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), based on Cryptographic Suite for Algebraic Lattices (CRYSTALS)-Kyber [19]
- FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA), based on CRYSTALS-Dilithium [20]
- FIPS 205, Stateless Hash-Based Digital Signature Standard (SLH-DSA), based on SPHINCS+ (for practical stateless hash-based signatures) [21]

Moreover, the process is continuing with a fourth round. The remaining candidates are the Key-Encapsulation Mechanisms (KEMs) Bit Flipping Key Encapsulation (BIKE), Classic McEliece, Hamming Quasi-Cyclic (HQC), and Supersingular Isogeny Key Encapsulation (SIKE). As there is no algorithm for digital signatures left from the initial submissions, NIST

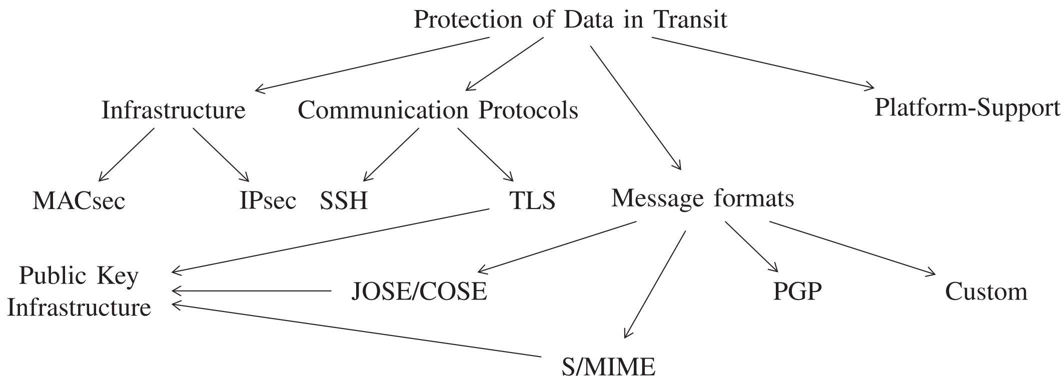


Figure 1. Overview of protocols used in different scenarios to protect data in transit.

launched another Call-for-Proposals on *Post-Quantum Cryptography: Digital Signature Schemes*, which is currently in the second round [22]. Hence, despite there are NIST standards already finalized, further algorithms are under consideration.

Naturally, the NIST process and its contributions from researchers all over the world are closely followed by government agencies from other nations.

The British National Cyber Security Center (NCSC) published a white paper recommending the use of the NIST standards or the hash-based signatures Leighton-Micali Hash-Based Signatures (LMS) or eXtended Merkle Signature Scheme (XMSS) [23].

In terms of post-quantum algorithms, the German Federal Office for Information Security / Bundesamt für Sicherheit in der Informationstechnik (BSI) recommends in its technical policy TR-02102-1 Version 2025-01 using FrodoKEM, Classic McElice or ML-KEM as a post-quantum cryptographic algorithm for encryption/key-agreement [24]. It mentions FrodoKEM and Classic McElice as a more conservative choice compared to the ML-KEM that is standardized by NIST. While FrodoKEM is not planned to be part of a NIST standard, its specification was submitted to International Organization for Standardization (ISO) for standardization [25].

For digital signatures, the policy recommends Merkle-Signatures, in detail XMSS or LMS, including Multi-Tree-Variants as described in [26] in addition to the NIST algorithms ML-DSA and SLH-DSA. The NIST algorithms should be used in the *hedged* version.

In general, the policy recommends combining a PQC approach and a classical one. The combination needs to ensure staying secure, as long as one of the used schemes is secure. Hash-based signatures are an exception in case they are properly implemented, i.e., they do not require a hybrid approach.

In contrast to the German BSI, the French Cybersecurity Agency (ANSSI) states in their PQC position paper, that the *ANSSI traditionally does not provide any closed list of recommended algorithms in order to avoid proscribing innovative state-of-the-art algorithms that could be well-suited for some particular use cases* [27]. However, a list of post-quantum algorithms together with recommendations is given. For KEM, they include ML-KEM and FrodoKEM. The list of digital signature algorithms contains ML-DSA, Falcon (FN-DSA), XMSS/LMS and SLH-DSA. In terms of combining PQC and classical algorithms, the ANSSI states their alignment with the position of the BSI recommending a hybrid approach.

Overall, the process of standardization results in the publication of various recommendations and draft standards. The analysis, including research on secure implementations, is still ongoing, leading to new attacks, cf. [28]. Despite the NIST is driving the most prominent competition, the government bodies of UK, Germany, and France are basically in line with the recommendations and have not announced any plans for running another competition.

While there is a focus on the most frequently used primitives, i.e., key encapsulation and digital signatures, several other, not that widely used primitives require consideration regarding PQ-security as well. A collection of such primitives with their PQC-status is given in [29]. The collection includes, e.g., schemes for authentication, protecting backups, and ensuring privacy used by Android, Apple, and Chrome. This list shows that a majority of the schemes are not PQ-secure. In particular, many of them rely on different versions of a Password-Authenticated Key Exchange (PAKE) scheme. While common PAKE schemes used today are not PQ-ready, there is a proposal of a generic construction to derive a PAKE from a KEM [30]. However, it is criticized not being fully PQ-secure [31]. Note that most of those schemes do not rely on standards but were designed for their specific purpose. Hence, it is up to the company to come up with PQC versions of their solutions.

Concluding, the current state, especially in a hybrid setting with a classic algorithm, provides a solid foundation for building and implementing protocols and further post-quantum secure solutions. However, when it comes to non-standard solutions, there are still many open questions and various schemes to adapt to provide PQ-security.

V. PROTOCOLS

In addition to developing and standardizing quantum-secure algorithms, protocol standards need to be adapted. Table I given an overview of the infrastructure and communication protocols and their actual status regarding PQC support.

TABLE I
OVERVIEW OF INFRASTRUCTURE AND COMMUNICATION PROTOCOLS, THEIR PQC SUPPORT, AND RELATED IMPLEMENTATIONS.

Protocol	Standard	PQC Support	Implementation
MACsec	IEEE Standard [32]	relies on symmetric cryptographic primitives	
IPsec	RFC 6071 [33]	RFC 8784 [34], Internet Darft [35]	Cisco IOS XE [36], Junos OS [37], strongSwan [38]
TLS	RFC 8446 [39]	Internet Draft [40]	Botan [41], WolfSSL [42], rustls [43], Open Quantum Safe* [44] Applications: Google experiments [6], Cloudflare [45]
SSH	RFC 4251 [46]	Internet Draft [47]	OpenSSH [48], Amazon implementation [49], Open Quantum Safe* [44]

* Use in production is not recommended by the project/developers.

A. Infrastructure

Common communication protocols to connect hosts to networks in a secure fashion or to establish a secure connection between networks are MACsec [32] and IPsec [33].

1) MACsec: As MACsec relies only on symmetric algorithms during the key agreement, using a 256-bit key is sufficient for post-quantum security. In addition, it is important to ensure that the key distribution is quantum-secure. Especially, since the session keys do not provide forward secrecy, i.e., a compromise of the long-term key material affects past session keys [50].

2) IPsec: For Internet Protocol Security (IPsec), Request For Comments (RFC) 8784 [34] defines a method to use pre-shared keys to achieve post-quantum security. The RFC is already supported by several products, like Ciscos IOS XE [36], strongSwan [38] and Junos OS [37]. This provides a viable solution already today. Potential adoptions of PQC for the Internet Key Exchange Protocol Version 2 (IKEv2) are in draft status. For example, [35] specifies a Hybrid Key Exchange with ML-KEM.

B. Communication Protocols

Common communication protocols include Transport Layer Security (TLS) and Secure Shell (SSH).

1) Transport Layer Security (TLS): The Transport Layer Security (TLS) protocol allows a secure end-to-end connection between applications. Various research has been conducted on how to best integrate post-quantum cryptography in the actual version of the protocol, TLS 1.3, and related performance, e.g., [51][52][53][54][55]. For TLS 1.3, a draft specifies a hybrid use of algorithms [40]. This ensures that the connections remain secure even if used algorithms are broken.

Version 3.7.0 of the Botan library [41] enables a hybrid key exchange per default using x25519/ML-KEM-768 and adds support for ML-KEM key exchange in a non-hybrid mode. WolfCrypt, the underlying library of WolfSSL also supports ML-KEM, ML-DSA, SPHINCS+, and stateful hash-based signatures. They are integrated as Kyber and Elliptic Curve Cryptography (ECC)/Kyber hybrid codepoints, as well as Dilithium signature algorithms [42]. Since version 0.23.22, rustls supports a hybrid PQ key exchange, using ECC and ML-KEM [43].

Other implementations of the draft and non-hybrid PQC key exchange methods are provided by the Open Quantum Safe project [44] in form of an OpenSSLv3 provider and an integration into a BoringSSL fork. However, those two implementations should not be considered *production quality* according to the project.

Note that a recent IETF draft states that TLS 1.2 will not be further enhanced, which implies, it will not support PQC, despite TLS 1.2 is still widespread [56].

Further experiments on challenges when using PQC-TLS at a large scale were conducted by Google [6]. Their tests revealed incompatibilities in network products that will be fixed via firmware updates. Similar PQC-support is enabled by Cloudflare [45], targeting support of all outbound connections by March 2024. This can be used with browsers supporting the hybrid cipher suite consisting of X25519 and Kyber-768, like Chrome, where it has been enabled since version 116 [57]. Moreover, Cloudflare provides real-time data on the percentage of their Hypertext Transfer Protocol Secure (HTTPS) connections that utilize PQC [58]. At the of time writing this paper, around 33% of the HTTPS connections are using PQC.

Hence, a draft standard and implementations are available. Some widespread experiments have been conducted successfully and first rollouts are taking place. Standardized support of PQC for TLS 1.3 is expected to build on the released NIST standards.

2) Secure Shell (SSH): Secure Shell (SSH) is a protocol for secure execution of remote commands. A very prominent implementation is OpenSSH, which is part of many major Linux distributions. OpenSSH made a hybrid key exchange method that combines ML-KEM with an Elliptic-Curve Diffie–Hellman (ECDH) key exchange default in version 9.9 [48]. The implementation relies on an individual submission [59], which has been replaced by an Internet Draft [47]. The same mechanism is implemented and used by Amazon Web Services (AWS) [49]. Amazon states in the PQC roadmap to adopt ML-KEM for SSH as soon as *as standards bodies such as the IETF publish implementation guidance for those protocols* [60]. The Open

Quantum Safe project [44] also provides an experimental implementation supporting PQC based on an OpenSSH-Fork.

Overall, with OpenSSL, that uses a hybrid approach per default, and the AWS implementation, there are real-world possibilities for PQC key-agreement, despite there being no final standard yet.

C. Message Security

On the message layer, the application can choose to encrypt/sign the transferred data, depending on the use case. Potential solutions include JOSE/COSE for sharing data between applications, S/MIME for mail/web pages and PGP for arbitrary data, including file exchange.

1) JOSE/COSE: JSON and CBOR are formats for data exchange between applications. The related signing and encryption standards are JOSE and COSE. For COSE, hash-based signatures are defined in RFC 8778 [61]. An active IETF drafts exists to support Dilithium [62]. In addition to this working group draft, other individual drafts have been submitted to the IETF as well.

2) S/MIME: The S/MIME standard [63] mandates the support of RSA-based and ECC-based ciphers for signing and encryption. Preparing the standard for the quantum-age is part of the *Limited Additional Mechanisms for PKIX and SMIME (lamps)* working group charter [64]. Nevertheless, the possibility of integrating PQC-ciphers into the mail client Thunderbird is briefly discussed in [65], and a demo integration was done by the MTG AG [66].

3) PGP: The options for using post-quantum ciphers in PGP were analyzed by Wussler [67], leading to an IETF draft [68]. A former version of this draft was formally analyzed by Tran et al. [69].

While there is work underway for all three standards, there is still a lack of practical implementations and experiments that will lead to solutions that can be used in production environments.

D. Platform-Support

Especially with the increasing use of cloud computing, it is important to take a look at the security foundations of the cloud providers and service platforms in general. Google Cloud Platforms (GCP) protects its internal communication with a protocol called Application Layer Transport Security (ALTS), which already employs post-quantum cryptography to protect against harvest now and decrypt later attacks [70]. Similar to Google, Meta is using a hybrid PQC implementation for most internal communications, protecting it against attacks recording traffic today and decrypting it with quantum computers when possible [71].

Amazon published a roadmap for the transition of Amazon Web Services (AWS) towards PQC, mentioning that their libraries, including those used for HTTPS-based endpoints, support PQC and hence, also allow customers testing the impact of PQC [60].

Microsoft launched a project called Quantum Safe [72] and participate [73] in the Open Quantum Safe project [44]. Furthermore, the integration of PQC in Microsoft libraries is ongoing [74].

E. Public Key Infrastructures (PKIs) and Certificates

Public Key Infrastructures (PKIs) are essential for ensuring trust in the digital world. Ranging from communication protocols to digitally signed documents - a reliable PKI is required to ensure the identity of the counterpart. For trustworthy certificates in the presence of quantum computers, the whole chain, starting with the root certificate must be quantum-secure.

The draft [80] defines a composite certificate combining ML-DSA with traditional signature algorithms. This solution ensures that the certificate remains secure even in case one of the algorithms is broken. A similar approach is used for KEM solutions [81] in the context of PKI-related profiles and protocols like Cryptographic Message Syntax (CMS) [82] and Public Key Infrastructure for X.509 (PKIX).

Various drafts are already published to be ready to proceed now the NIST standards are finalized. They include certificates using ML-KEM [75], and Dilithium [76].

The draft [83] relies on the Stateful Hash-Based Signature Schemes (S-HBS) [26], Hierarchical Signature System (HSS), eXtended Merkle Signature Scheme (XMSS) [84], and XMSSˆMT, a multi-tree variant of XMSS and provides algorithm identifiers for X.509 PKIs. While their security is well understood, those signature schemes come with the drawback that they can only create a limited amount of signatures, and it is required to maintain a state to remain secure.

During the transition phase, it is important that also legacy systems that might not support post-quantum cryptography can verify a certificate with classic algorithms. The specifications above cannot be used in such a scenario, as they require the verifying system process PQC signatures. A possible approach in the transition scenario is using related certificates, as laid out in the draft specifications [78] and the individual submission [79]. The impact of hybrid certificates on current implementations was investigated in [85]. The authors concluded the certificates can be processed by the tested solutions without or with minor modifications.

Another option is specified by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [77], namely to include an alternative signature in a certificate. This allows clients that are not capable of processing PQC algorithms to ignore this signature, while others can benefit from it. However, the drawback of this approach is the increased certificate size for all consuming entities. Table II provides and overview of the different options.

When it comes to commercial products, PKI solution vendors are working towards addressing the upcoming challenges, preparing examples [86], offering experimental suites [87][9] or solutions [88].

Despite various activities that are underway, neither the majority of the standardization work nor the related implementations have been concluded yet. As especially the root

TABLE II
OVERVIEW DIFFERENT WAYS HOW CERTIFICATES CAN SUPPORT PQ ALGORITHMS.

Certificate Type	Standard	Purpose	Contains
PQC Certificates	Internet Drafts [75][76]	supports only PQC-enabled system	PQC keys and signatures
Hybrid Certificates	ITU-T Specification [77], Internet Drafts [78][79]	can be verified by PQC-enabled and legacy systems; allow a transition phase	PQC and non-PQC keys and signatures
Composite Certificates	Internet Draft [80]	verification requires PQC-enabled system; remain secure in cases the non-PQC or the PQC algorithm breaks	PQC and non-PQC keys and signatures

certificates are commonly valid for several years, it is important to plan their replacement together with a sound transition approach.

VI. Foundations and Libraries

Together with research and standardization of PQC algorithms, their implementation is progressing. A popular project to support *the transition to quantum-resistant cryptography* is Open Quantum Safe [44]. It is part of the Linux Foundation's Post-Quantum Cryptography Alliance. Its main working items are a C library for post-quantum algorithms, called liboqs, and prototype integration into protocols and applications.

Currently, liboqs supports the standards ML-KEM, ML-DSA, XMSS, LMS, as well as Falcon and SPHINCS+. Furthermore, the NIST round 4 candidates Classic McEliece, BIKE and HQC, as well as, FrodoKEM and NTRU-Prime. The project provides several language wrappers to allow using it for example in C++, JAVA, Go, and Python. However, the project page does recommend refraining from using the library in production environments, as it has not undergone a thorough audit/analysis process yet.

Another library is libpqcrypto [90], that provides C-reference implementations for 19 algorithms with different parameter sets and comes with a Python API. The project team warns that there might be security problems in the library, either due to issues of the cryptographic primitives themselves or due to software bugs.

PQClean [91][95] aims at providing standalone C implementations of the NIST PQC algorithms. The project states that the library is suited for research purposes and suggest an use-case-specific assessment for any other use of the library. The JavaScript library pqc.js makes use of PQClean.

The Cloudflare Interoperable, Reusable Cryptographic Library (CIRCL) [93] provides several Go implementations of cryptographic primitives, including PQC. It comes with the cautious note that it is provided as-is and parts are experimental.

A popular library that provides PQC support is Bouncy Castle for Java and C# [94]. Its implementation includes ML-KEM, ML-DSA, SLH-DSA, LMS, and XMSS algorithms. Further algorithms are implemented as well, but project states that those non-standardized algorithms can be used for experiments and hybrid scenarios, but not for long-term protection. Further libraries that have already been mentioned in the context of TLS are Botan [41] and wolfCrypt [42]. An overview of implementations is given in Table III.

Overall, there are two aspects to consider about using PQC algorithms today: (1) First standards have recently been finalized and the security research is ongoing. They also do not have the benefit of a long history of intensive security research that current standards possess. Therefore, the BSI, recommends using the current PQC algorithms in a hybrid mode. (2) In addition to the security of the algorithms, quality [95] and security of its implementations are important. This includes sufficient quality assurance and auditing to prevent vulnerabilities and security bugs as well as resistance against potential side-channel attacks like [96]–[98].

VII. Applications

In addition to developments on standards and implementations that are directly related to those standards, there are also products and solutions that move forward in supporting PQC.

A. Messengers

Messengers are widely used on mobile phones to exchange information. Very common apps are WhatsApp, Signal, Threema, Telegram, and Apples iMessage. Key functionality of all apps is, in addition to various different ways of exchanging messages, protecting the security and privacy of its users. Hence, they all provide end-to-end encryption for the exchanged messages [99]–[101][1]. Note that WhatsApp and Signal rely on the same protocol [102].

Recent versions of Signal support a post-quantum secure key agreement protocol, called PQXDH [103]. Apple's iMessage protocol PQ3 also relies on PQC [104]. In addition to PQXDH, PQ3 also performs PQC rekeying. However, both protocols do not provide a quantum-secure authentication method yet.

B. Blockchain

Blockchain technology enables a decentralized trust model with various applications. Those include cryptocurrencies such as Bitcoin [105], health applications [106], and blockchain-backed logistics [107]. Given the data stored in and protected by different blockchains, it is important to ensure that neither the data a chain contains can be manipulated using quantum attacks

[1]Note that for Telegram, end-to-end encryption is only enabled for secret chats, not per default.

TABLE III
OVERVIEW OF SELECTED LIBRARIES SUPPORTING POST-QUANTUM ALGORITHMS.

Name	**Language**	**Supported Algorithms**	**Experimental**
liboqs [89]	C	ML-KEM, ML-DSA, XMSS, LMS, Kyber, Dilithum, Falcon, SPHINCS+, Classic McEliece, BIKE, HQC, FrodoKEM, NTRU-Prime. CROSS, MAYO	Y
libpqcrypto [90]	Python, C	Classic McEliece, Dilithium, Kyber, FrodoKEM, NTRU Prime, SPHINCS+ (and more)	Y
PQClean [91]	C	Kyber, HQC, Classic MCEliece, Dilithium, Falcon, SPHINCS+	Y
pqc.js [92]	JavaScript	uses PQClean	Y
CIRCL [93]	Go	ML-KEM, X-Wing, Kyber, Frodo, CSIDH, ML-DSA, Dilithium	Y
Bouncy Caste [94]	JAVA, C#	ML-KEM, ML-DSA, SLH-DSA, LMS, XMSS*	
Botan [41]	C++	ML-DSA, SLH-DSA, HSS/LMS, XMSS	
wolfSSL/wolfCrypt [42]	C	ML-KEM, ML-DSA, SPHINCS+, LMS/HSS, XMSS	

* Further PQC-algorithms are implemented, but according to the documentation unsuitable for long-term use.

nor future transactions can be forged. The key primitives that a blockchain uses are hash functions and public-key cryptography. A good overview on the impact and potential solutions is given in [108]. In addition to the scientific analysis like [109]–[111], there are products that already focus on post-quantum secure blockchains using post-quantum cryptography [112] or combine PQC with quantum key distributions technologies [113]. In contrast, making the popular system Ethereum post-quantum secure, is on the roadmap, but is stated ongoing research [114]. For Bitcoin, an experimental branch, Bitcoin Post-Quantum (BPQ), exists [115].

VIII. Quantum Key Distribution

Quantum mechanisms cannot only be used to attack cryptographic primitives. It is also possible to use them for protecting digital communication. In essence, quantum mechanisms allow exchanging key material in a secure way. This is called Quantum Key Distribution (QKD). The shared keys allow establishing a secure data transmission channel, e.g., via symmetric algorithms.

In contrast to post-quantum cryptography, using QKD requires specific hardware, like encryptors provided by Fraunhofer [116] or ID Quantique [117]. As it is relying on quantum effects, there are expected to be very secure systems. However, neither the German BSI [118] nor the United States National Security Agency (NSA) [119] consider QKD a priority. The BSI mentions that the technology is not yet mature enough in terms of security and only suitable for some niche use cases. The NSA highlights cost efficiency and better maintainability of PQC compared to QKD. Both recommend focusing on PQC.

IX. Conclusion and Future Work

Quantum computers endanger the security of cryptographic algorithms. Especially asymmetric algorithms are affected. This requires new algorithms as well as updated standards to make use of those new algorithms. Various efforts from research over standardization to implementation are currently under way to address this challenge. This paper started by looking at possibilities to secure the underlying network infrastructure. As IPsec and MACsec can rely on secret-key cryptography, the remaining challenge is secure key management.

In order to achieve end-to-end security, SSH can be used with post-quantum security, e.g., via OpenSSH, first rollouts of PQC TLS implementations are taking place. Standards for message encryption are still at a comparably early stage. However, libraries, especially BouncyCastle for JAVA and C#, wolfCrypto for C and Botan for C++, provide algorithms that can already integrated into applications; given the required expert knowledge is available. Several applications like messengers, services like Blockchain and cloud platforms are moving or have already completed important steps towards supporting PQC. The use of quantum technologies to exchange key material, called QKD, is another area of ongoing research, especially suited for niche applications.

Overall, the transition will require thorough planning. This paper highlighted where first steps can be done already today. Depending on the use case, hybrid approaches can protect against quantum attacks while preventing risks due to attacks on comparably new PQC algorithms. Furthermore, becoming crypto-agile, in the sense that algorithms can be exchanged easily, will not only help in addressing the current PQC challenge, but also reduce the effort of future transitions of cryptographic algorithms.

References

[1] J. Schmidt and A. Lawall, “Theoretical and practical aspects in identifying gaps and preparing for post-quantum cryptography”, in *Proceedings of the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), 2024*, Nice, France: IARIA, Nov. 2024, pp. 36–42, ISBN: 978-1-68558-206-7.

[2] U.S. Department of Homeland Security, *Post-quantum cryptography - frequently asked questions*, https://www.dhs.gov/sites/default/files/publications/post_quantum_cryptography_faq_3_seals_october_2021_508.pdf, retrieved: May, 2025.

[3] National Security Agency, *The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ*, https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/1/CSI_CNSA_2.0_FAQ_.PDF, retrieved: May, 2025.

[4] P. E. Hoffman and S. Celi (WG chairs), *Post-Quantum Use In Protocols (pquip)*, https://datatracker.ietf.org/wg/pquip/about/, retrieved: May, 2025.

[5] ETSI, *Quantum-Safe Cryptography (QSC)*, https://www.etsi.org/technologies/quantum-safe-cryptography, retrieved: May, 2025.

[6] Google, *How Google is preparing for a post-quantum world*, https://cloud.google.com/blog/products/identity-security/how-google-is-preparing-for-a-post-quantum-world, retrieved: May, 2025.

[7] IBM, *Make the world quantum safe*, https://www.ibm.com/quantum/quantum-safe, retrieved: May, 2025.

[8] Microsoft, *Post-quantum cryptography*, https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/, retrieved: May, 2025.

[9] Utimaco, *Post Quantum Cryptography*, https://utimaco.com/solutions/applications/post-quantum-cryptography, retrieved: May, 2025.

[10] ETSI, *ETSI TR 103 619 V1.1.1 (2020-07) - CYBER; Migration strategies and recommendations to Quantum Safe schemes*, https://www.etsi.org/deliver/etsi_tr/103600_103699/103619/01.01.01_60/tr_103619v010101p.pdf.

[11] Post-Quantum Cryptography (PQC) Working Group, "Risk model technical paper", FS-ISAC, Tech. Rep., 2023, available at https://www.fsisac.com/hubfs/Knowledge/PQC/RiskModel.pdf, retrieved: May, 2025.

[12] T. Patterson, *Moving toward a Quantum Security Maturity Index*, Presentation at Post-Quantum Cryptography Conference 2023, available at https://pkic.org/events/2023/pqc-conference-amsterdam-nl/pkic-pqcc_tom-patterson_accenture_moving-toward-a-quantum-security-maturity-index.pdf, retrieved: May, 2025.

[13] DigiCert, *Post-Quantum Cryptography (PQC) Maturity Model*, https://www.digicert.com/resources/post-quantum-cryptography-maturity-model.pdf, retrieved: May, 2025.

[14] Intel, *Intel® Software Guard Extensions (Intel®SGX)*, https://www.intel.de/content/www/de/de/products/docs/accelerator-engines/software-guard-extensions.html, retrieved: May, 2025.

[15] Intel, *Intel® Trust Domain Extensions (Intel® TDX)*, https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/overview.html, retrieved: May, 2025.

[16] AMD, *AMD Secure Encrypted Virtualization (SEV)*, https://www.amd.com/de/developer/sev.html, retrieved: May, 2025.

[17] G. Caruso, "Post-quantum algorithms support in Trusted Execution Environment", Available at https://webthesis.biblio.polito.it/31076/, Ph.D. dissertation, Politecnico di Torino, 2024.

[18] NIST - Computer Security Resource Center, *Post-Quantum Cryptography PQC - Call for Proposals*, https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals, retrieved: May, 2025.

[19] NIST, "Module-lattice-based key-encapsulation mechanism standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 203, 2024. DOI: 10.6028/NIST.FIPS.203.

[20] NIST, "Module-lattice-based digital signature standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 204, 2024. DOI: 10.6028/NIST.FIPS.204.

[21] NIST, "Stateless hash-based digital signature standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 205, 2024. DOI: 10.6028/NIST.FIPS.205.

[22] NIST, *Post-Quantum Cryptography: Additional Digital Signature Schemes*, https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures, retrieved: May, 2025.

[23] National Cyber Security Center, *Next steps in preparing for post-quantum cryptography*, https://www.ncsc.gov.uk/whitepaper/next-steps-preparing-for-post-quantum-cryptography, retrieved: May, 2025.

[24] Federal Office for Information Security / Bundesamt für Sicherheit in der Informationstechnik (BSI), *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf, Technische Richtlinie, Bonn, Deutschland, 2025.

[25] E. Alkim et al., *FrodoKEM - Practical quantum-secure key encapsulation from generic lattices*, "https://frodokem.org/, retrieved: May, 2025.

[26] D. Cooper *et al.*, "Recommendation for Stateful Hash-Based Signature Schemes", Special Publication (NIST SP), National Institute of Standards and Technology, U.S. Department of Commerce, Tech. Rep., 2020. DOI: https://doi.org/10.6028/NIST.SP.800-208.

[27] ANSSI, *ANSSI views on the Post-Quantum Cryptography transition (2023 follow up)*, https://cyber.gouv.fr/sites/default/files/document/follow_up_position_paper_on_post_quantum_cryptography.pdf, retrieved: May, 2025.

[28] P. Ravi, D. Jap, S. Bhasin, and A. Chattopadhyay, "Invited paper: Machine learning based blind side-channel attacks on pqc-based kems - A case study of kyber KEM", in *IEEE/ACM International Conference on Computer Aided Design, ICCAD 2023, San Francisco, CA, USA, October 28 - Nov. 2, 2023*, IEEE, 2023, pp. 1–7. DOI: 10.1109/ICCAD57390.2023.10323721.

[29] B. Westerbaan et al., *Fancy cryptography in the wild*, https://github.com/fancy-cryptography/fancy-cryptography, retrieved: May, 2025.

[30] H. Beguinet, C. Chevalier, D. Pointcheval, T. Ricosset, and M. Rossi, "GeT a CAKE: Generic Transformations from Key Encaspulation Mechanisms to Password Authenticated Key Exchanges", in *Applied Cryptography and Network Security*, M. Tibouchi and X. Wang, Eds., Cham: Springer Nature Switzerland, 2023, pp. 516–538, ISBN: 978-3-031-33491-7.

[31] N. Alnahawi, J. Alperin-Sheriff, D. Apon, and A. Wiesmaier, *NICE-PAKE: On the security of KEM-based PAKE constructions without ideal ciphers*, Cryptology ePrint Archive, Paper 2024/1957, available at https://eprint.iacr.org/2024/1957, 2024.

[32] M. Seaman, *IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security*.

[33] S. Frankel and S. Krishnan, *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*, RFC 6071, Feb. 2011. DOI: 10.17487/RFC6071.

[34] S. Fluhrer, P. Kampanakis, D. McGrew, and V. Smyslov, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*, RFC 8784, Jun. 2020. DOI: 10.17487/RFC8784.

[35] P. Kampanakis and G. Ravago, "Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)", Internet Engineering Task Force, Internet-Draft draft-ietf-ipsecme-ikev2-mlkem-00, May 2025, Work in Progress, 10 pp.

[36] CISCO, *Security and VPN Configuration Guide, Cisco IOS XE 17.x*, https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m-sec-cfg-quantum-encryption-ppk.html, retrieved: May, 2025.

[37] Juniper Networks, *Release Notes: Junos OS Release 22.4R1*, https://www.juniper.net/documentation/us/en/software/junos/release-notes/22.4/junos-release-notes-22.4r1/index.html, retrieved: May, 2025.

[38] strongSwan, *IPsec and Related Standards*, https://docs.strongswan.org/docs/latest/features/ietf.html, retrieved: May, 2025.

[39] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. DOI: 10.17487/RFC8446.

[40] D. Stebila, S. Fluhrer, and S. Gueron, "Hybrid key exchange in TLS 1.3", Internet Engineering Task Force, Internet-Draft draft-ietf-tls-hybrid-design-12, Jan. 2025, Work in Progress, 24 pp.

[41] Botan, *Botan - Release Notes*, https://botan.randombit.net/news.html, retrieved: May, 2025.

[42] wolfSSL, *wolfSSL Support for Post-Quantum*, https://www.wolfssl.com/products/wolfcrypt-post-quantum/, retrieved: May, 2025.

[43] J. Birr-Pixton et al., *rustls releases*, https://github.com/rustls/rustls/releases, retrieved: May, 2025.

[44] *Open Quantum Safe Project*, https://openquantumsafe.org/, retrieved: May, 2025.

[45] W. Evans, B. Westerbaan, C. Patton, P. Wu, and V. Gonçalves, *Post-quantum cryptography goes GA*, https://blog.cloudflare.com/post-quantum-cryptography-ga/, retrieved: May, 2025.

[46] C. M. Lonvick and T. Ylonen, *The Secure Shell (SSH) Protocol Architecture*, RFC 4251, Jan. 2006. DOI: 10.17487/RFC4251.

[47] P. Kampanakis, D. Stebila, and T. Hansen, "PQ/T Hybrid Key Exchange in SSH", Internet Engineering Task Force, Internet-Draft draft-ietf-sshm-mlkem-hybrid-kex-02, Apr. 2025, Work in Progress, 14 pp.

[48] OpenSSH, *Openssh 9.9 release notes*, https://www.openssh.com/txt/release-9.9, retrieved: May, 2025.

[49] AWS Security Blog, *Post-quantum hybrid SFTP file transfers using AWS Transfer Family*, https://aws.amazon.com/de/blogs/security/post-quantum-hybrid-sftp-file-transfers-using-aws-transfer-family, retrieved: May, 2025.

[50] ETSI, *ETSI TR 103 617 V1.1.1 (2018-09) - Quantum-Safe Virtual Private Networks*, https://www.etsi.org/deliver/etsi_tr/103600_103699/103617/01.01.01_60/tr_103617v010101p.pdf.

[51] J. I. E. Pablos, M. E. Marriaga, and A. P. d. Pozo, "Design and Implementation of a Post-Quantum Group Authenticated Key Exchange Protocol With the LibOQS Library: A Comparative Performance Analysis From Classic McEliece, Kyber, NTRU, and Saber", *IEEE Access*, vol. 10, pp. 120 951–120 983, 2022. DOI: 10.1109/ACCESS.2022.3222389.

[52] J. Henrich, A. Heinemann, A. Wiesmaier, and N. Schmitt, "Performance Impact of PQC KEMs on TLS 1.3 Under Varying Network Characteristics", in *Information Security*, E. Athanasopoulos and B. Mennink, Eds., Cham: Springer Nature Switzerland, 2023, pp. 267–287, ISBN: 978-3-031-49187-0. DOI: 10.1007/978-3-031-49187-0_14.

[53] S. Paul, Y. Kuzovkova, N. Lahr, and R. Niederhagen, "Mixed Certificate Chains for the Transition to Post-Quantum Authentication in TLS 1.3", in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '22, Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 727–740, ISBN: 9781450391405. DOI: 10.1145/3488932.3497755.

[54] C. R. Garcia, A. C. Aguilera, J. J. V. Olmos, I. T. Monroy, and S. Rommel, "Quantum-Resistant TLS 1.3: A Hybrid Solution Combining Classical, Quantum and Post-Quantum Cryptography", in *2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2023, pp. 246–251. DOI: 10.1109/CAMAD59638.2023.10478407.

[55] M. Sosnowski *et al.*, "The Performance of Post-Quantum TLS 1.3", in *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT 2023, Paris, France: Association for Computing Machinery, 2023, pp. 19–27, ISBN: 9798400704079. DOI: 10.1145/3624354.3630585.

[56] R. Salz and N. Aviram, "TLS 1.2 is in Feature Freeze", Internet Engineering Task Force, Internet-Draft draft-ietf-tls-tls12-frozen-08, Apr. 2025, Work in Progress, 6 pp.

[57] D. O'Brien, *Protecting Chrome Traffic with Hybrid Kyber KEM*, https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html, retrieved: May, 2025.

[58] Cloudflare Radar, *Adoption & Usage - Post-Quantum Encryption Adoption*, https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption, retrieved: May, 2025.

[59] P. Kampanakis, D. Stebila, and T. Hansen, "PQ/T Hybrid Key Exchange in SSH", Internet Engineering Task Force, Internet-Draft draft-kampanakis-curdle-ssh-pq-ke-03, Work in Progress, 14 pp.

[60] M. Campagna, M. Goldsborough, and P. O'Donnell, *AWS post-quantum cryptography migration plan*, https://aws.amazon.com/de/blogs/security/aws-post-quantum-cryptography-migration-plan/, retrieved: May, 2025.

[61] R. Housley, *Use of the HSS/LMS Hash-Based Signature Algorithm with CBOR Object Signing and Encryption (COSE)*, RFC 8778, Apr. 2020. DOI: 10.17487/RFC8778.

[62] M. Prorock, O. Steele, R. Misoczki, M. Osborne, and C. Cloostermans, "ML-DSA for JOSE and COSE", Internet Engineering Task Force, Internet-Draft draft-ietf-cose-dilithium-06, Apr. 2025, Work in Progress, 19 pp.

[63] J. Schaad, B. C. Ramsdell, and S. Turner, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification*, RFC 8551, Apr. 2019. DOI: 10.17487/RFC8551.

[64] R. Housley and T. Hollebeek (WG Chairs), *Limited Additional Mechanisms for PKIX and SMIME (lamps)*, https://datatracker.ietf.org/wg/lamps/about/, retrieved: May, 2025.

[65] C. Döberl et al., "Quantum-resistant End-to-End Secure Messaging and Email Communication", in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23, Benevento, Italy: Association for Computing Machinery, 2023, pp. 1–8, ISBN: 9798400707728. DOI: 10.1145/3600160.3605049.

[66] MTG AG, *PQC Anwendungen jetzt testen!*, https://www.mtg.de/de/post-quantum-kryptografie/pqc-demo/#PQC_Testanwendungen, retrieved: May, 2025.

[67] A. Wussler, "Post-Quantum cryptography in OpenPGP", M.S. thesis, Wien, 2023. DOI: 10.34726/hss.2023.106226.

[68] S. Kousidis, J. Roth, F. Strenzke, and A. Wussler, "Post-Quantum Cryptography in OpenPGP", Internet Engineering Task Force, Internet-Draft draft-ietf-openpgp-pqc-07, Feb. 2025, Work in Progress, 107 pp.

[69] D. D. Tran, K. Ogata, and S. Escobar, "A formal analysis of OpenPGP's post-quantum public-key algorithm extension", in *Proceedings of the 2nd International Workshop on Formal Analysis and Verification of Post-Quantum Cryptographic Protocols (FAVPQC), 2023*, Brisbane, Australia: JAIST Press, 2023, pp. 22–35.

[70] S. Kölbl, R. Misoczki, and S. Schmieg, *Securing tomorrow today: Why Google now protects its internal communications from quantum threats*, https://cloud.google.com/blog/products/identity-security/why-google-now-uses-post-quantum-cryptography-for-internal-comms, retrieved: May, 2025.

[71] S. Lin *et al.*, *Post-quantum readiness for TLS at Meta*, https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/, retrieved: May, 2025.

[72] Microsoft, *Quantum-safe overview*, https://quantum.microsoft.com/en-us/vision/quantum-cryptography-overview, retrieved: May, 2025.

[73] Microsoft Research, *Post-quantum cryptography*, https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/, retrieved: May, 2025.

[74] A. Thipsay, *Microsoft's quantum-resistant cryptography is here*, https://techcommunity.microsoft.com/blog/microsoftsecurityandcompliance/microsofts-quantum-resistant-cryptography-is-here/4238780, retrieved: May, 2025.

[75] S. Turner, P. Kampanakis, J. Massimo, and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-kyber-certificates-10, Apr. 2025, Work in Progress, 71 pp.

[76] J. Massimo, P. Kampanakis, S. Turner, and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-dilithium-certificates-09, May 2025, Work in Progress, 89 pp.

[77] Telecommunication Statdardization Sector of ITU, *Directory Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*, Series X - Data networks and open system communication, Oct. 2019.

[78] A. Becker, R. Guthrie, and M. J. Jenkins, "Related Certificates for Use in Multiple Authentications within a Protocol", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-cert-binding-for-multi-auth-06, Dec. 2024, Work in Progress, 14 pp.

[79] C. Bonnell, J. Gray, D. Hook, T. Okubo, and M. Ounsworth, "A Mechanism for Encoding Differences in Paired Certificates", Internet Engineering Task Force, Internet-Draft draft-bonnell-lamps-chameleon-certs-06, Apr. 2025, Work in Progress, 52 pp.

[80] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, and S. Fluhrer, "Composite ML-DSA for use in X.509 Public Key Infrastructure and CMS", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-pq-composite-sigs-04, Mar. 2025, Work in Progress, 82 pp.

[81] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, and S. Fluhrer, "Composite ML-KEM for use in X.509 Public Key Infrastructure and CMS", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-pq-composite-kem-06, Mar. 2025, Work in Progress, 66 pp.

[82] R. Housley, J. Gray, and T. Okubo, *Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)*, RFC 9629, Aug. 2024. DOI: 10.17487/RFC9629.

[83] D. V. Geest, K. Bashiri, S. Fluhrer, S.-L. Gazdag, and S. Kousidis, "Use of the HSS and XMSS Hash-Based Signature Algorithms in Internet X.509 Public Key Infrastructure", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-x509-shbs-13, Dec. 2024, Work in Progress, 35 pp.

[84] A. Huelsing, D. Butin, S.-L. Gazdag, J. Rijneveld, and A. Mohaisen, *XMSS: eXtended Merkle Signature Scheme*, RFC 8391, May 2018. DOI: 10.17487/RFC8391.

[85] J. Fan et al., "Impact of post-quantum hybrid certificates on PKI, common libraries, and protocols", *International Journal of Security and Networks*, vol. 16, no. 3, pp. 200–211, 2021. DOI: 10.1504/IJSN.2021.117887.

[86] GlobalSign, *Post Quantum Computing - Future-proofing digital trust with safe certificates*, https://www.globalsign.com/en/post-quantum-computing, retrieved: May, 2025.

[87] Keyfactor, *Post-Quantum Cryptography Keys and Signatures*, https://doc.primekey.com/ejbca/ejbca-operations/ejbca-ca-concept-guide/certificate-authority-overview/post-quantum-cryptography-keys-and-signatures, retrieved: May, 2025.

[88] Entrust, *Post-Quantum Cryptography*, https://www.entrust.com/solutions/post-quantum-cryptography, retrieved: May, 2025.

[89] Open Quantum Safe, *Liboqs*, https://github.com/open-quantum-safe/liboqs, retrieved: May, 2025.

[90] PQCRYPTO, *Libpqcrypto*, https://libpqcrypto.org/, retrieved: May, 2025.

[91] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, "Improving software quality in cryptography standardization projects", in *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010.

[92] *pqc.js*, https://github.com/Dashlane/pqc.js/, retrieved: May, 2025.

[93] A. Faz-Hernandez and K. Kwiatkowski, *Introducing CIRCL: An Advanced Cryptographic Library*, Available at https://github.com/cloudflare/circl. v1.6.1 Accessed May, 2025, Cloudflare, Jun. 2019.

[94] *Bouncy Castle*, https://www.bouncycastle.org/, retrieved: May, 2025.

[95] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, "Improving software quality in cryptography standardization projects", in *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010.

[96] P. Ravi, A. Chattopadhyay, J. P. D'Anvers, and A. Baksi, "Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results", *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 1–54, Mar. 2024, ISSN: 1539-9087. DOI: 10.1145/3603170.

[97] C. Mujdei et al., "Side-channel Analysis of Lattice-based Post-quantum Cryptography: Exploiting Polynomial Multiplication", *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 1–23, Mar. 2024, ISSN: 1539-9087. DOI: 10.1145/3569420.

[98] A. T. Hoang et al., "Deep Learning Enhanced Side Channel Analysis on CRYSTALS-Kyber", in *2024 25th International Symposium on Quality Electronic Design (ISQED)*, 2024, pp. 1–8. DOI: 10.1109/ISQED60706.2024.10528674.

[99] Telegram, *Mtproto mobile protocol*, https://core.telegram.org/mtproto, retrieved: May, 2025.

[100] Threema, *Threema cryptography whitepaper*, https://digi77.com/software/public/threema_cryptography_whitepaper.pdf, retrieved: May, 2025.

[101] Signal, *Technical information*, https://signal.org/docs/, retrieved: May, 2025.

[102] M. Marlinspike, *Whatsapp's signal protocol integration is now complete*, https://signal.org/blog/whatsapp-complete/, retrieved: May, 2025.

[103] E. Kret and R. Schmidt, *The pqxdh key agreement protocol*, https://signal.org/docs/specifications/pqxdh/, retrieved: May, 2025.

[104] Apple Security Engineering and Architecture (SEAR), *iMessage with PQ3: the new state of the art in quantum-secure messaging at scale*, https://security.apple.com/blog/imessage-pq3/, retrieved: May, 2025.

[105] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, 1st. O'Reilly Media, Inc., 2014, ISBN: 1449374042.

[106] novo nordisk, *Blockchain x clinical trials*, https://techlife.novonordisk.com/cases/epid, retrieved: May, 2025.

[107] DHL Trend Research, *Blockchain in logistics*, https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf, retrieved: May, 2025.

[108] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks", *IEEE Access*, vol. 8, pp. 21 091–21 116, 2020. DOI: 10.1109/ACCESS.2020.2968985.

[109] N. K. Parida, C. Jatoth, V. D. Reddy, M. M. Hussain, and J. Faizi, "Post-quantum distributed ledger technology: A systematic survey", *Scientific Reports*, 2023. DOI: 10.1038/s41598-023-47331-1.

[110] D. A. Bard, J. J. Kearney, and C. A. Perez-Delgado, "Quantum advantage on proof of work", *Array*, vol. 15, p. 100 225, 2022, ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2022.100225.

[111] H. Gharavi, J. Granjal, and E. Monteiro, "Post-quantum blockchain security for the internet of things: Survey and research directions", *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1748–1774, 2024. DOI: 10.1109/COMST.2024.3355222.

[112] Algorand, *Leading on post-quantum technology*, https://algorand.co/technology/post-quantum, retrieved: May, 2025.

[113] M. Misiaszek-Schreyner, Ł. Kujawski, M. Kosik, P. Kulicki, and M. Sopek, *The QSB, quantum secured blockchain, a whitepaper*, https://www.quantumblockchains.io/wp-content/uploads/2023/06/QBCK_WhitePaper.pdf, retrieved: May, 2025.

[114] Ethereum, *Future-proofing Ethereum*, https://ethereum.org/en/roadmap/future-proofing/, retrieved: May, 2025.

[115] N. Anhao, *Bitcoin Post-Quantum*, https://bitcoinpq.org/download/bitcoinpq-whitepaper-english.pdf, retrieved: May, 2025.

[116] Fraunhofer HHI, *Quantum Key Distribution System for Future-Proof Security*, https://www.hhi.fraunhofer.de/en/departments/pn/research-groups/free-space-optical-systems/quantum-key-distribution-1.html, retrieved: May, 2025.

[117] ID Quantique, *Providing the ultimate, long-term data protection in a post-quantum world*, https://www.idquantique.com/quantum-safe-security/products/#quantum_key_distribution, retrieved: May, 2025.

[118] French Cybersecurity Agency (ANSSI), Federal Office for Information Security (BSI), Netherlands National Communications Security Agency (NLNCSA), Swedish National Communications Security Authority, Swedish Armed Forces, *Position Paper on Quantum Key Distribution*, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Quantum_Positionspapier.pdf, retrieved: May, 2025.

[119] National Security Agency/Central Security Service, *Quantum Key Distribution (QKD) and Quantum Cryptography (QC)*, https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/, retrieved: May, 2025.

Invisible Identifiers - How Browser Fingerprinting Challenges Internet Privacy and User Anonymity

Alexander Lawall
IU International University of Applied Science
Erfurt, Thüringen, Germany
alexander.lawall@iu.org

***Abstract*—Browser fingerprinting has emerged as a sophisticated and increasingly prevalent technique for identifying and tracking users online without relying on traditional methods like cookies. This paper provides a comprehensive overview of browser fingerprinting techniques, ranging from passive and active methods like Hypertext Transfer Protocol (HTTP) header analysis to advanced machine learning-assisted side-channel attacks. By evaluating the uniqueness, stability, and entropy of different methods, the study highlights how the synergistic combination of multiple techniques enhances the accuracy and persistence of user identification. The analysis demonstrates that browser fingerprinting poses a significant challenge to digital privacy by operating invisibly, often without user knowledge or consent. Despite regulatory frameworks such as the General Data Protection Regulation, the widespread use of fingerprinting scripts remains largely unchecked, exploiting legal loopholes and technological asymmetries. The paper also explores the potential of privacy-preserving fingerprinting systems for secure user authentication while emphasizing the urgent need for adaptive countermeasures, regulatory reforms, and increased user awareness to protect individual privacy in the evolving digital landscape.**

Keywords-browser fingerprinting; device fingerprinting; tracking; privacy; active fingerprinting; passive fingerprinting.

I. Introduction

This work is an extended version of *Fingerprinting and Tracing Shadows: The Development and Impact of Browser Fingerprinting on Digital Privacy*, published at SECURWARE 2024 [1], [2]. In the increasingly digitized world, the issues of online privacy and data security are becoming more complex. Particularly in tracking — monitoring users and their devices across different web servers — browser fingerprinting has emerged as an effective technique for creating detailed user profiles. Unlike the storage of information via cookies, which requires explicit user consent as mandated by the European General Data Protection Regulations (GDPR) guidelines, fingerprinting does not require such consent. A browser fingerprint can be generated in the background without any obvious signs to the end user, leaving them unaware of whether and to what extent they are being tracked.

It is possible to manipulate a device locally to alter its fingerprint. This is often not feasible for all users, unlike deleting cookies. This invisible threat is not apparent to the general public and raises significant privacy concerns, as individuals can be tracked unnoticed. These profiles can contain private information, depending on the server operators, including age group, ethnic origin, social circles, and interests of the affected person.

Browser fingerprinting poses a threat to the privacy of the general public. Contrary to being a threat, it is an opportunity to provide valuable information to enhance the authentication mechanisms. Both perspectives are explored throughout this paper. The focus will be on the various techniques of fingerprinting to understand how accurate and detailed user profiles can be created. The main research questions that this paper seeks to answer are:

RQ1 *"What methods are used in browser fingerprinting and what user data are collected in the process?"*

RQ2 *"How has the development of browser fingerprinting as a user identification method influenced user privacy and data protection in the digital space?"*

The paper is structured as follows: Section I introduces browser fingerprinting and its privacy implications. In Section II, the theoretical background explains how fingerprinting works and its legal challenges. Section III outlines techniques like HTTP Headers, Canvas, and WebGL Fingerprinting. Section IV examines the impact of fingerprinting on privacy and the regulatory landscape. Section V concludes with a summary of the findings, emphasizing the need for stronger privacy measures and further research on countermeasures.

II. Theoretical Background

This section lays the conceptual foundation for understanding browser fingerprinting, detailing its underlying mechanisms, legal ambiguities, and role in modern tracking practices. It introduces both passive and active techniques used to collect identifying data from users' browsers without explicit consent, highlighting the technical simplicity yet high effectiveness of these methods. Furthermore, it explores the growing tension between evolving tracking technologies and regulatory protections like the GDPR, illustrating how fingerprinting often operates in legal gray zones that undermine user privacy and control.

A. Fingerprinting

Browser fingerprinting refers to collecting characteristic information that the browser directly or indirectly reveals about itself. Often used to track users, this technology has also found applications in IT security, such as fraud detection. Unlike tracking methods like cookies, browser fingerprinting does not require storing data on the user's computer, allowing the process to occur secretly and without consent [3, p. 1]. Consequently, creating a new identity, similar to deleting cookies, is not easily achievable, and GDPR privacy laws often provide little protection. Unlike cookie tracking, browser fingerprinting is not explicitly mentioned in the GDPR. It should fall under the collection of identifiable information but website operators frequently claim “legitimate interest”, enabling such data collection without the user’s consent [4].

Active transmission of data is not required for browser fingerprinting, as loading a webpage can transmit various pieces of information, such as the user’s preferred language, within the HTTP headers. This passive data collection provides only a limited amount of information, so it is often supplemented with active data collection methods. An active approach typically employs JavaScript to interface with the browser and gather information, such as screen resolution, installed add-ons, and graphics card data, merging them into a unique fingerprint [5, pp. 1, 3].

Similar to human fingerprints, browser fingerprinting relies on the uniqueness of browser characteristics, which typically do not change significantly with regular use. This allows for accurate user identification over extended periods [5, p. 2]. However, not all collected data points are equally unique or stable, necessitating careful selection of information to achieve accurate results. The fingerprinting algorithm combines both passively and actively collected data into a unique string. Depending on the operator’s goals, adjustments can be made; for instance, using cookies, the fingerprint might be less stable but more unique, while tracking users without cookies requires high stability [6, pp. 1-5]. Eckersley’s study showed that participant browsers already had high entropy, indicating many unique characteristics sufficient for accurate fingerprinting, though not stable enough for long-term accuracy. In recent years, potential entropy has increased with new techniques like HTML Canvas, WebGL-based hardware fingerprints, audio API fingerprints, plug-in-based fingerprints, and methods utilizing mouse movements or differences in HTML parsing between browsers, making cross-browser user identification possible [5, pp. 4-5].

B. Concerns for Digital Privacy

Historically, the greatest threat to online tracking was posed by cookies, along with other technologies like Flash cookies, which have lost significance in recent years. Changes by browser manufacturers, such as Mozilla, which rendered many exploited technologies, so-called “super-cookies”, ineffective [7], and additional browsers planning to block or eliminate third-party cookies in the coming years [8], have shifted the landscape. Following the GDPR, the use of non-essential cookies has been further restricted and standardized for the first time, defining how users share their data through cookies [9]. In contrast, browser fingerprinting occurs in the background and leaves no stored information on the user’s computer. Thus, the use of fingerprints not only circumvents previous issues related to local storage, such as privacy laws and technical limitations but also persists even when local data is deleted or when incognito mode is used.

A 2021 study of the Alexa Top 100,000 websites found that nearly 10% of the sites used scripts to generate fingerprints [10, pp. 11-12]. Comparing this to a similar 2014 study, which recorded 5.5% of the top 100,000 sites using canvas fingerprinting scripts, reveals an almost doubling of usage over seven years [11]. This suggests a shift towards online tracking using this technology, which is much harder to detect and prevent compared to cookies. The creation of a fingerprint is imperceptible to the user, with no simple way to effectively change or delete their fingerprint. Cookie banners give a false sense of security while tracking continues in the background without consent.

Thus, browser fingerprinting poses an active threat to privacy, as users often have no control over the collection and use of their data. This stands in opposition to many current data protection principles, such as the GDPR.

III. Methods of Browser Fingerprinting

In the context of browser fingerprinting techniques, the methods of data collection are varied and comprehensive. Therefore, specific properties and criteria are used to select techniques. The following sections will encompass the explanation of the techniques in terms of their functionality and their applications will be discussed to provide a detailed understanding of their use. An evaluation based on the advantages and disadvantages of each technique is also included to weigh their effectiveness and potential risks. Given the ever-increasing number of techniques, only the most commonly used, established, or novel methods will be presented here.

A. HTTP Header Attributes

1) Definition and Basics: The HTTP request header is a part of every HTTP request exchanged between a client (web browser) and a server, transmitting various functional and compatibility-related information [12]. While individual attributes are typically not unique, their combination can enhance the distinctiveness of a client within a larger population. This explanation is based on HTTP version 1.1, with HTTP/2 introducing fundamental structural changes. However, most attributes remain in use within the modified header frame [13].

2) Analysis: The attributes of HTTP request headers can vary depending on the browser and its version. For fingerprinting purposes, it is crucial to select fields that remain consistent over time and are not easily influenced by user behavior. For example, the Host header, which conveys the target server’s domain, should be avoided as it is directly dependent on the request destination. In contrast, the User-Agent field typically exhibits high stability and provides extensive information, making it particularly suitable for fingerprinting [14].

Studies by AmIUnique [15, p. 880] and PanOptiClick [6, p. 5] identify the User-Agent, Accept, Content-Encoding, and Content-Language fields as reliable attributes. These studies collected user fingerprints voluntarily and demonstrated their effectiveness in user identification. The User-Agent field, although not standardized, frequently contains information about browser compatibility, version, and operating system, often with varying levels of detail. Due to its lack of standardization and manufacturer-specific implementations, the User-Agent field exhibits high entropy, with modifications typically occurring only through browser updates [16].

The Accept, Content-Encoding, and Content-Language fields convey less information individually but can reveal insights into the operating system, browser type, and language preferences. Uncommon languages or specific language-region combinations may yield unique fingerprints [17]–[19]. Additional fields such as Referer, Connection, Content-Length, X-Forwarded-For, Cookie, and Cache-Control can complement fingerprinting but provide minimal uniqueness on their own. However, the presence of certain headers like X-Forwarded-For may indicate specific configurations or proxy usage [15, pp. 879-880].

The DoNotTrack (DNT) header, although originally intended to signal tracking preferences, has paradoxically become a fingerprinting target due to its voluntary nature and lack of enforcement [20, p. 313]. Furthermore, the sequence of header fields may serve as an additional fingerprinting feature, particularly when combined with manipulated User-Agent information. Cookies, while transmitted within HTTP headers, require client-side storage and are thus excluded from this discussion.

3) Advantages: The primary advantage of utilizing HTTP headers for fingerprinting is the entirely passive nature of information collection. As described in the analysis section, header transmission occurs automatically with each request and can be extracted by most web servers, such as Nginx, without significant overhead [21]. Since all processing takes place on the server side, this method remains invisible to the user and does not require client-side scripts, making the network traffic indistinguishable from regular requests. In summary, this method is efficient, unobtrusive, and compatible with most web servers, processing data on the server side without a noticeable impact on the client.

4) Disadvantages: Despite their utility, HTTP headers offer limited information due to the low entropy of most attributes. The User-Agent field, while informative, is widely recognized and can be manipulated using browser extensions like User-Agent Switchers (i.e., User-Agent Switcher for Chrome). Consequently, the reliability of this attribute alone should be critically assessed.

Additionally, the use of HTTP header-based fingerprinting without explicit user consent raises significant privacy concerns under the General Data Protection Regulation (GDPR). Therefore, any implementation should undergo legal review prior to deployment to ensure compliance with data protection regulations [22].

B. Enumeration of Browser Plugins

1) Definition and Basics: Browser plugins, whether preinstalled or user-added, have historically constituted one of the most significant methods for system recognition, alongside font detection. Most browser features are indirectly modified, with the exception of extensions, which maintain their popularity. The capability to obtain precise enumeration of these extensions remains highly sought after [15, pp. 878-880].

2) Analysis: Information-rich plugins, such as Flash, have gradually disappeared from the market. Since 2016, most browsers, including Firefox, no longer support the formerly widespread Netscape Plugin Application Programming Interface (NPAPI). This development has resulted in the detection of installed extensions via JavaScript and the navigator.plugins object in modern browsers primarily revealing only standard plugins like PDF viewers [23]. Although the removal of plugins represents significant progress for privacy protection, the limited capability to read certain plugins for compatibility purposes continues to provide opportunities to identify differences between systems and browsers, thereby enabling inferences about the system. Direct detection of user-installed add-ons is not possible, which restricts the significance of collectible data for fingerprinting [15, pp. 886-887].

Despite the impossibility of directly reading user-installed extensions, researchers have discovered novel methodologies for their enumeration. Chromium-based browsers possess the capability to access extension settings via a local URL. A project in GitHub exploits this vulnerability by requesting internal resources such as images for over 1,000 different extensions in the background. The status code can indicate whether the respective extensions are installed [24].

Ad blockers represent particularly popular add-ons, and their behavior in removing unwanted content from pages can also be detected. Ad blockers typically employ known lists of advertising companies and CSS elements for removal. A script can create such an element and verify whether it has been modified. With a sufficient dataset, the existence of deployed blocklists can be demonstrated [25].

Currently, it is also possible to read a portion of the programs installed on a device beyond extensions. A vulnerability in various browsers allows for reading the status of the handler protocol to determine whether the associated software is installed. Programs such as Skype and Zoom add these protocols within the system to enable launching the corresponding program with parameters via a link.

3) Advantages: Given that extensions are installed by users and considering the extensive market of available extensions, this method offers high uniqueness coupled with stability for fingerprinting purposes.

4) Disadvantages: This technique provides profound insights into the privacy of unsuspecting users. A study demonstrated that beyond less sensitive information like interests, extremely sensitive data can be inferred, including health conditions, practiced religion, and political views [26, pp. 11-12].

Since precise reading of extensions is not possible, this process relies on limited methods, making it error-prone. Therefore, continuous maintenance and updating are required to ensure its reliability.

C. Canvas Fingerprinting

1) Definition and Basics: Canvas fingerprinting represents a technique for generating a digital fingerprint through the utilization of the Canvas element introduced in HTML5. This methodology employs the Canvas API to render a 2D graphic imperceptibly in the background. The manner in which various browsers and devices process this image varies due to differences in hardware acceleration, installed fonts, and graphic libraries. The resultant fingerprint exhibits exceptional stability and uniqueness [3, pp. 1-3].

2) Analysis: A script embedded within a webpage incorporates an invisible Canvas element that renders a predetermined 2D graphic in the background. Utilizing the Canvas context, textual elements can also be rendered with diverse fonts and font sizes. WebFonts additionally facilitate the dynamic loading of fonts from the internet. These can be specifically selected according to purpose to evaluate systems for uniqueness in font rendering.

The resulting image data can be extracted via the functions *getImageData* and *toDataURL*, subsequently formatted into a fingerprint as desired, for instance, through the application of a simple hashing algorithm. The hash is then transmitted to a server via a web request for processing and storage.

Beyond storing the fingerprint for subsequent identification, an alternative application methodology involves comparing the fingerprint with an extensive database of known fingerprints and corresponding system configurations. With a substantial dataset, reliable predictions regarding the system's configuration can be established [3, pp. 2-4].

3) Advantages: The research findings of Mowery and Shacham demonstrated that the implementation of Canvas fingerprinting is exceptionally straightforward, requiring minimal lines of client-side code. It utilizes fundamental JavaScript functions and is deployable across all common web applications. The fingerprint generation process occurs inconspicuously for the user and presents significant challenges for blocking. This is attributable to the frequent deployment of Canvas operations on the web and the complex challenge of distinguishing normal applications from fingerprinting scripts.

The creation of the fingerprint, due to its simplicity, can be executed with high velocity and exhibits high stability in conjunction with high uniqueness and entropy. Consequently, its application is particularly valuable in real-time tracking applications [3, pp. 1-5].

4) Disadvantages: Alterations in browser environments, such as updates or graphic settings, may influence the stability of the fingerprint. Additionally, the variability of hardware and software configurations can lead to inconsistencies. As an active technique, the execution of code on the client side is unavoidable and entails the risk of detection and potential blockage by, for example, blocklists targeting known fingerprinting scripts [3, pp. 3-7].

Although the utilization is imperceptible to the user, the limited number of interfaces for retrieving generated Canvas data ensures that these can be monitored and manipulated by extensions [10, p. 4]. Add-ons such as CanvasBlocker exploit this to provide users with the option to prevent data extraction or manipulate the data in the Canvas, thereby generating a continuously new fingerprint and rendering identification impossible [27].

Finally, while the implementation of Canvas fingerprinting is relatively straightforward, the data analysis and interpretation are comparatively complex and may require a certain level of expertise in the field to be processed correctly [3, pp. 6-8].

D. WebGL Fingerprinting

1) Definition and Basics: WebGL fingerprinting is a technique utilizing the WebGL JavaScript API, based on OpenGL ES 2.0, allowing web applications to render both 2D and 3D graphics with high performance by directly accessing the GPU [28]. Unlike Canvas fingerprinting, which focuses on 2D graphics and identifies software differences mainly through fonts and graphic libraries, WebGL fingerprinting provides deeper and more precise detection capabilities. It captures unique hardware information, particularly details about the graphics processor, distinguishing it significantly from Canvas fingerprinting and broadening its application for tracking purposes [3, p. 4]. The inherent trade-offs between WebGL and Canvas fingerprinting ensure that neither method entirely supplants the other; their complementary nature makes them suitable for different scenarios.

2) Analysis: WebGL fingerprinting uses a Canvas element to access the API. Similar to Canvas fingerprinting, it creates an invisible element performing 3D operations in the background to collect data without user interaction. A straightforward application involves accessing specific variables, such as *UNMASKED_VENDOR_WEBGL* and *UNMASKED_RENDERER_WEBGL*, using the *getParameter* function in the WebGL context. These variables provide information about the graphics hardware manufacturer (Vendor) and model (Renderer). For example, a Vendor entry like "Intel" indicates an integrated graphics unit, while "Nvidia" combined with "GeForce GTX 970" as Renderer indicates a dedicated graphics card. These details can reveal insights into the system being used [29, p. 17]. Privacy concerns have led browsers like Apple's WebKit to provide generic information instead of specific data to protect user privacy. Since 2020, WebKit has masked Vendor and Renderer information, as well as shading language details [30]. Firefox similarly groups graphics processor models into categories instead of displaying specific models [31]. In practice, this means that an Nvidia card from the 900 series onward, for example, is reported as "GeForce GTX 980" [32]. In summary, research investigating hardware fingerprinting using HTML5 demonstrated the capability to identify devices based on GPU performance. It utilizes the graphics processor's clock frequency and clock skew to render complex 3D graphics, measuring GPU performance based on the number of frames rendered within a period, providing insights into the GPU's frequency and core count [33, pp. 3-4].

Furthermore, WebGL fingerprinting can render graphics, employing techniques like shadows, textures, lighting, anti-aliasing, and transparency, to generate system-specific unique outputs. However, the three-dimensional environment results in increased client-side resource utilization and more complex code compared to the simpler 2D Canvas [3, p. 4]. While Laperdrix et al. initially deemed WebGL unreliable for fingerprinting in 2016, subsequent research demonstrated otherwise. Cao et al. [34] refuted Laperdrix et al.'s findings, attributing the inconsistencies to non-standardized rendering tasks and uncontrolled variables such as canvas size and anti-aliasing settings. By implementing 20 consistently defined tasks rendered under carefully controlled parameters, Cao et al. achieved a 99.24% success rate, surpassing Laperdrix et al.'s 90.84%. Their work also demonstrated the ability to identify a system across different browsers with a 91.44% stability [34, p. 2].

To enhance fingerprint stability, the "DrawnApart" project focuses on subtle variations in GPU Execution Units (EUs) rather than relying on differences in graphic rendering. This method exploits the unique characteristics of a device's GPU stack to detect speed variations across different EUs, creating a robust and reliable GPU signature. Experiments involving over 2,500 devices showed a fingerprint stability increase of up to 67% compared to other current techniques [35, pp. 1, 6-12].

3) Advantages: As demonstrated by Cao et al., WebGL can offer high uniqueness and stability [34]. Its direct interface with the system ensures consistency across browsers, making it challenging for users to evade identification through simple browser changes or reinstalls. Despite changes to enhance WebGL's resistance to fingerprinting, it reliably identifies users. The successor to WebGL, WebGPU, is currently in development, promising even more privacy risks due to its closer hardware access, allowing for classifications with up to 98% accuracy in 150 milliseconds, a reduction from the 8 seconds WebGL took [36].

4) Disadvantages: The complexity of WebGL fingerprinting is significantly higher compared to previous techniques, necessitating careful consideration whether a simpler Canvas approach combined with other methods might be accurate enough for specific use cases. Intensive tasks in a 3D environment can also strain the target system, leading to longer fingerprint creation times [3, p. 4]. Implementing WebGL requires caution, as shown by the cases of Laperdrix et al. and Cao et al., and opting for a ready-made solution might be advisable. Moreover, WebGL shares Canvas's vulnerability to blocked or misread data if detection methods rely on differences in rendered graphics. Even novel methods like DrawnApart can be mitigated through countermeasures, such as limiting to a single EU [35, p. 12]. WebGL may also not be available or disabled on some devices, necessitating consideration of alternatives, such as using the 2D Canvas.

E. Audio Fingerprinting

1) Definition and Basics: The Web Audio API is a JavaScript interface for processing and synthesizing audio signals in web browsers, part of the HTML5 standard. It can identify systems through manufacturing differences in audio hardware. Methods analyze signal processing characteristics, hardware differences, and system responses to specific audio signals for fingerprinting [37, pp. 1107-1109]. The API's indirect access to audio hardware allows for system identification based on subtle variations introduced during manufacturing.

2) Analysis: Audio fingerprinting involves various acoustic measurements to create a unique device fingerprint. It requires an AudioContext linking an AudioBuffer, Oscillator, and Compressor. The AudioBuffer represents a small audio segment, while the Oscillator generates a waveform at a defined frequency using a mathematical function. The Compressor manipulates the audio signal. The unique waveform generated and manipulated reflects system characteristics, allowing a unique fingerprint to be created by applying a hash function to the final waveform [38], [39]. This method, known as "Dynamic Compressor (DC)", is highly stable, producing the same fingerprint for the user each time using a reliable hash function [37, pp. 1109-1111].

Another method is the "Fast Fourier Transform" (FFT), converting audio signals from the time domain to the frequency domain. It measures hardware implementation differences to identify characteristics. FFT is less stable than DC, often requiring multiple attempts for consistent results. DC and FFT are often used together for more reliable outcomes [37, pp. 1111-1114].

Researchers from New Orleans compared various techniques, including custom-designed ones, alongside DC and FFT. These included creating "Custom Signals", "Merged Signals", and analyzing generated AM and FM waves. All techniques showed good stability, averaging two to four attempts for fingerprint matching [40, pp. 3-5].

3) Advantages: The generated fingerprints are highly stable and can differentiate systems based on their properties. Queiroz and Feitosa showed that mobile devices using Firefox could be consistently recognized and grouped by their stable fingerprints [37, p. 1119]. Techniques like DC are simple to implement and offer high stability. Other promising techniques, especially when used together, could enhance potential but are more challenging to implement [40, pp. 1-3].

4) Disadvantages: While audio fingerprinting offers high stability, it lacks uniqueness and accuracy on its own and should be used with other fingerprinting techniques [37, p. 1119]. Additionally, the Web Audio API can be disabled on devices or manipulated by add-ons like "Canvas Blocker", which also blocks and manipulates Canvas and WebGL [27].

F. Font Fingerprinting

1) Definition and Basics: Font fingerprinting is a browser fingerprinting technique that identifies devices by recognizing installed fonts. This method operates on the premise that each device possesses a specific combination of fonts. This combination can be unique or, when combined with other fingerprinting techniques, contribute to a unique and relatively stable digital fingerprint. Installed fonts are among the more unique identifiers of a device, often providing the highest entropy, especially when considered alongside other data points such as installed plugins, information gleaned via the Canvas API, and the browser's User-Agent [20, p. 314]. These elements together enable the creation of a detailed and individualized device profile, which can be used for tracking and identification purposes.

2) Analysis: Until the end of 2020, Adobe Flash was frequently used to enumerate installed fonts. With the deprecation of Flash Player and its removal from common browsers, new methods had to be developed [34, p. 10]. Since pure JavaScript does not offer a direct function to detect installed fonts, a fallback mechanism is employed. This involves applying a specific font, and if it is unavailable, the system defaults to a standard fallback font. The technique leverages the different dimensions that fonts require to render the same text. A text string is rendered in a specific font, and the resulting dimensions are compared to expected values. This allows the determination of whether a specific font is available or a fallback was used [20, p. 311].

Using JavaScript, invisible *div* elements can be created, containing selected texts with specified fonts. The dimensions of the element are then compared with known target values, and a match is recognized as the font being installed. The list of all installed fonts can then be combined into a fingerprint via a hash algorithm [20, p. 311].

Another method is using the Canvas element. As described in Section III-C, the Canvas can render texts in requested fonts and use fallbacks if these are not available. Unlike direct text, the Canvas element has a fixed size, but the *measureText* function of the Canvas context allows reading the width of the drawn text, allowing further inferences about available fonts [41, p. 12].

It should be noted that JavaScript under Chrome and Edge currently allows reading local fonts, but the Local Font Access API used for this is experimental, only available in these two browsers, and requires user consent, making it unsuitable for fingerprinting purposes [42].

3) Advantages: Font recognition offers high entropy and stability since fonts are rarely changed. Fonts can be installed by the user or by software, with each operating system pre-installing different fonts. This allows the identification of the operating system and potentially its version, as manufacturers can make adjustments. It also allows the detection of installed software packages like Office or Photoshop, which installs fonts for use [5, p. 7].

4) Disadvantages: Without Flash, font recognition is done through "brute-force" methods, reducing accuracy if unknown fonts are installed. This requires selecting a list of fonts to test and measuring them against the values to be tested. If fonts are installed that are not within the list, they cannot be detected, reducing the accuracy of the result [34, p. 10]. Another problem is fonts that have too strong similarities in their dimensions to possible fallbacks. This can lead to false positives, so a forced fallback test should be performed for a text beforehand. Since the fallback font is unknown, a non-existent font is requested, and the resulting dimensions are used to recognize other non-existent fonts [20, p. 311].

Finally, it is still possible to manipulate the read fonts through extensions [43] or, as in the case of Apple's WebKit, to only deliver values pre-installed by the operating system, causing users to blend into the crowd [44].

G. Screen Fingerprinting

1) Definition and Basics: Screen fingerprinting identifies a device by analyzing various screen-related characteristics, including screen resolution, pixel depth, color depth, and browser window size. This method leverages the uniqueness of screen configurations and browser modifications, which can create rare resolution combinations [45, p. 20].

2) Analysis: JavaScript provides attributes for screen and browser window characteristics through the *window.screen object*, offering details like color depth (*colorDepth*), screen orientation (*screenOrientation*), and screen dimensions (*screenHeight*, *screenWidth*). Values, such as *window.innerWidth* and *window.innerHeight*, determine the browser window's inner area, which can be altered by toolbars or bookmark bars [34, p. 3].

3) Advantages: Screen and window resolution information typically have high entropy, making them useful for stabilizing fingerprints when combined with other techniques. This method is particularly effective for distinguishing between desktop, tablet, and mobile devices, as these have distinct resolutions and aspect ratios compared to standardized desktop screens [37, p. 277].

4) Disadvantages: Since values are derived from browser attributes rather than hardware tests, they can be limited or altered by extensions or privacy settings. Browsers like TOR set the window to a fixed size of 1000x1000 pixels, reducing uniqueness, and browsers like Firefox always report a color depth of 24. Additionally, users with multiple monitors or those using zoom functions can affect the accuracy of screen fingerprinting, as there is no reliable way to determine the zoom factor directly, which reduces entropy [34, p. 10].

H. WebRTC Fingerprinting

1) Definition and Basics: WebRTC is a standard and accessible JavaScript interface available in most browsers. It facilitates real-time communication over stateless HTTP by establishing direct connections between participants, allowing the extraction of local network adapter information. This can reveal private and public IP addresses, which can be used for fingerprinting or identifying users behind proxies or VPNs [41, p. 12]. It also provides information about connected devices, such as microphones, webcams, and speakers.

2) Analysis: Unlike other browser mechanisms like camera or microphone access, establishing a WebRTC connection requires no permissions or user notifications. After successfully connecting to the target computer via a Session Traversal Utilities for NAT (STUN) server, the individual addresses can be read from the RTCPeerConnection object in the form of iceCandidates [46, p. 667].

This data can be used for fingerprinting, but the data collection does not have to stop there. Since WebRTC always seeks the shortest path for a connection, it is possible to enumerate the local network through, for example, port scanners, creating a unique picture of the target's environment. Furthermore, it is possible to read all local addresses of the adapters, which, in addition to connections to VPNs, can also include set-up virtual adapters for Virtual Machines [46, pp. 667-668].

The DetectRTC project [47] demonstrates what functions are directly available through WebRTC. The most important are the microphones, webcams, and speakers. However, the exact device names are not possible without the necessary permissions. WebRTC does, however, allow reading the Media Device IDs of the respective devices, which, in connection with the respective active WebRTC functions, lead to unique fingerprints [48].

3) Advantages: Extracting private and public IPs provides deep insights, especially for identifying targets behind VPNs or proxies. No other technique can silently reveal addresses behind Network Address Translation (NAT) [49, p. 273]. The collected data is highly unique; a study with 80 devices found over 97% uniqueness using only WebRTC [46, p. 668].

4) Disadvantages: WebRTC might be disabled in the target browser, or extensions might block its usage without user consent. To read the Media IDs of the devices, a request for access rights for the respective devices is required. This can alert the user that a page may be performing dubious actions in the background. This is therefore not recommended for a secret operation.

Finally, WebRTC requires an infrastructure in the form of a STUN server, which must be set up independently or used by third parties. This makes it a technique that requires further dependencies and should therefore be considered depending on the intended use.

I. CSS Fingerprinting

1) Definition and Basics: Different to the active fingerprinting techniques using JavaScript, CSS fingerprinting is a passive method. CSS is a stylesheet language primarily used to enhance the presentation of HTML elements. Over time, the CSS specification has expanded to include selectors and filters, enabling limited dynamic selections, which this technique leverages [50, p. 10].

2) Analysis: Until 2010, the *:visited* selector could identify if a website had been visited by changing the link color, detectable via JavaScript. This was possible because browsers displayed already visited links in a different color, and this color difference was read out by JavaScript. After this was patched, researchers explored time-based methods to read user history, but these required JavaScript and were impractical [51, p. 4].

In 2015, Takei et al. introduced a JavaScript-free method using CSS properties and multiple *@media* queries to fetch URLs based on defined rules. By considering the requesting IP address and URL parameters, the server could then identify system properties like screen dimensions, resolution, touchscreen presence, installed fonts, browser, and OS [52, pp. 3-5]. A current GitHub project demonstrates this method's practical capabilities [53]. Individual CSS properties were used together with a variety of *@media* queries to call up URLs according to defined rules.

3) Advantages: CSS fingerprinting's independence from JavaScript allows it to identify even cautious users who block JavaScript or use extensions like NoScript. Software projects like TOR usually block JavaScript or use extensions like NoScript to give the user the possibility to execute selected scripts. This technique can even detect if JavaScript is disabled via noscript tags [52, p. 2]. Since this method is currently little used and rather unknown, further research has shown that no practical solution currently exists for users to effectively prevent it.

4) Disadvantages: Takei et al.'s method provides limited data, which, without JavaScript, can only be supplemented by techniques like header analysis (as presented in Section III-A). Oliver Brotchie notes in his project repository that the method is not currently scalable, as each request requires over 1MB of CSS files to be downloaded. However, he warns that upcoming CSS Values 4 implementation could reduce download sizes significantly, making the method more practical. Additionally, font recognition relies on brute-forcing, which, considering network traffic, can be noticeable [53]. The font recognition, as presented in Section III-F, is based on the principle of brute forcing, i.e., the massive trying out of fonts, which can be conspicuous when considering the network traffic.

J. Additional JavaScript Attributes

1) Definition and Basics: Most of the previously discussed techniques actively use JavaScript to extract information from various interfaces. Additional possibilities are briefly mentioned here to provide a more comprehensive picture. Since these techniques share many characteristics with other JavaScript-based methods, listing their pros and cons is omitted.

2) Analysis: The *navigator* object in browsers provides information, such as DoNotTrack status, user agent details, platform, languages, cookies usage, granted and available permissions, and time zone [20, p. 9]. JavaScript implementation varies between browsers and versions, and Mowery et al. demonstrated that these differences are measurable and can indicate the software and hardware used [3].

However, the implementation of JavaScript itself can also vary from browser to browser and version to version. Mowery et al. proved in 2011 that the different implementations of functions are measurable and can therefore provide a conclusion about the software and hardware used [3]. In addition to the differences in the execution itself, there are also differences in whether various functions are built into the browser and usable on the platform. This offers an alternative way of UserAgent detection, should this have been manipulated by extensions, for example [54].

Another technique caused uncertainty among Tor users in the past. Despite disabled Canvas, the *getClientRects* function could be used to obtain the exact data of DOM elements. Similar to the Canvas fingerprint, these factors could change greatly depending on implementation, font sizes, and screen resolutions, enabling identification in the otherwise anonymous browser [55]. The vulnerability has been fixed in Tor but remains exploitable in other browsers [56].

3) Advantages: JavaScript-based fingerprinting techniques are highly versatile and widely applicable since JavaScript is essential for web functionality. These methods can collect a broad range of information, such as user agent details, time zones, and system settings, often without requiring user consent or visibility. The stealthy nature of JavaScript fingerprinting allows it to operate in the background, making it difficult for users to detect. Moreover, JavaScript-based attributes work consistently across different browsers, enabling effective cross-browser tracking.

4) Disadvantages: However, JavaScript fingerprinting is limited by browser-specific implementations, which can result in inconsistent data collection. Privacy-focused browsers like Tor or extensions, such as NoScript, actively block or obscure JavaScript-based tracking, reducing its effectiveness. Additionally, users are becoming more aware of privacy risks and increasingly use tools to disable or modify JavaScript functions. Finally, updates to browsers may close vulnerabilities or alter features that JavaScript fingerprinting relies on, decreasing its long-term viability.

K. Advanced Techniques Using Machine Learning

1) Definition and Basics: Most active techniques discussed so far use JavaScript to gather hardware and software information. They rely on unique data combinations based on implementation quirks or directly available information. Newer methods often employ "side-channels", capturing additional data by observing behavioral differences during various operations within the execution environment. Methods like plugin enumeration (cf. Section III-B), font fingerprinting (cf. Section III-F), and CSS fingerprinting (cf. Section III-I) use this approach in simple forms by testing known combinations to gain indirect information. These side-channel methods can be implemented with minimal effort but can also be used in more sophisticated ways with machine learning to gather otherwise unobtainable information [57, p. 1].

2) Analysis: Wang et al. explored using techniques such as cache usage, memory consumption, and CPU activity to identify visited websites. In earlier methods, CSS selectors were leveraged to glean browsing history, revealing significant privacy risks and prompting swift remedial actions. Side-channel techniques utilize an array of strategies to yield more accurate analyses of system behavior. These methods involve complex calculations that impose a load on the hardware, with machine learning models categorizing the results against expected values from known sites. Their tests demonstrated an accuracy rate of 80-90% in identifying websites [57, pp. 3-5]. While Wang et al. addressed multiple attack vectors, including compromised machines with direct operating system access, the feasibility of executing such attacks solely through JavaScript measurements remains uncertain. Further research is needed, but implementations using WebAssembly [58] and the Performance API [59] are conceivable.

3) Advantages: This method is invisible to the user and provides insightful information not available through conventional means. Currently, there are no effective methods to protect users from such techniques [57, pp. 1-3].

4) Disadvantages: While previous techniques aimed to identify a user over time, this method has the potential to offer dangerous insights into the individual's behavior behind the screen. However, the technique is still in its initial stage and remains a theoretical approach not yet tested in in real-world scenarios. It is unlikely to be reliably utilized by malicious actors in the near future [57, p. 6].

IV. Discussion

Browser fingerprinting can be used positively for security, as shown by technologies like BrFast and private, passive user recognition methods. Such technologies offer promising alternatives for user authentication by leveraging device-specific attributes without the need for intrusive cookies or explicit user interaction. They provide a non-invasive method to identify users, particularly for fraud detection and bot prevention. However, there's a significant risk of misuse, especially in the field of advertising and mass surveillance. The advertising industry, driven by creating accurate user profiles, heavily invests in digital advertising, with data-driven ads accounting for 60-70% of digital ad revenue in Germany. Personalized ads significantly impact Generation Z, who discover products primarily through social media and whose purchasing decisions are increasingly influenced by algorithmic recommendations.

Traditionally, data collection relied on cookies, but users developed ways to avoid tracking, such as deleting cookies or using incognito mode. However, unlike cookies, browser fingerprints are collected in the background, making them invisible and far more persistent. Fingerprints are difficult to alter, and their cross-browser and cross-device capabilities exacerbate the problem by enabling long-term tracking across multiple platforms [34]. GDPR regulations mandate user consent for data collection, but enforcement is inconsistent, and compliance with fingerprinting guidelines remains unclear, even with new laws like Germany's Telecommunications Telemedia Data Protection Act (TTDSG) [60].

A. Affected Demographics

Online tracking is ubiquitous, affecting nearly all user groups. A 2016 study of the top 1 million websites revealed extensive tracking, with services like Google and Facebook present on over 10% of sites [41]. Following the GDPR, fingerprinting scripts increased significantly, with 68.8% of the top 10,000 websites employing such methods by 2020 [10]. This shift illustrates how fingerprinting has replaced traditional cookie-based tracking in response to regulatory pressure.

TABLE I
OVERVIEW OF FINGERPRINTING METHODS

Fingerprinting Method	Uniqueness	Stability	Entropy	Impact on User Privacy	Defense Techniques
HTTP Header Attributes	Low	Moderate	Low	Moderate impact: limited detail but useful when combined with other methods.	Altering or masking headers (e.g., randomizing User-Agent).
Enumeration of Browser Plugins	Moderate	High	High	High impact: reveals sensitive data, such as installed plugins.	Disabling plugin enumeration, avoiding unnecessary add-ons.
Canvas Fingerprinting	High	Moderate	High	High impact: generates unique fingerprints based on rendering.	CanvasBlocker extension to block or manipulate rendering.
WebGL Fingerprinting	High	High	High	High impact: collects detailed hardware data for tracking.	Block or manipulate WebGL outputs.
Audio Fingerprinting	Moderate	High	Moderate	High impact: captures unique audio processing details.	Disable Web Audio API, use privacy extensions.
Font Fingerprinting	High	High	Moderate	High impact: identifies installed fonts, making it persistent.	Limit font access with privacy-focused browsers (e.g., Tor).
Screen Fingerprinting	Moderate	High	Low	Moderate impact: uses screen resolution and window size but less effective on mobile devices.	Fix window size or limit resolution reporting with privacy browsers.
WebRTC Fingerprinting	Very High	High	Very High	Very high impact: exposes real IP addresses, even behind VPNs.	Disable WebRTC, use extensions that block data collection.
CSS Fingerprinting	Low	Moderate	Low	Low impact: provides limited system and style information.	Limit or disable CSS fingerprinting through extensions or scripts.
JavaScript Attributes	Moderate	High	Moderate	Moderate impact: uses various browser features for tracking.	Disable unnecessary JavaScript functions or use privacy extensions.
Advanced Machine Learning Fingerprinting	Very High	Very High	Very High	Very high impact: uses side-channel data (e.g., CPU/cache) for tracking.	Limit access to Performance API and WebAssembly, emerging defenses needed.

However, fingerprinting does not affect all users equally. A study with 234 participants found that demographics like age, gender, education, IT background, and privacy awareness influence trackability. Men and those with higher education were found to be less trackable, while users with lower privacy knowledge or older devices were more easily identified [61]. Despite this, many participants believed they could protect themselves from fingerprinting, underestimating its stealth and technical complexity.

Additionally, fingerprinting poses a disproportionate risk to marginalized communities. Research by Queiroz and Feitosa shows that low-income users and those in the Global South — who are more likely to use older mobile devices — are significantly more identifiable through audio fingerprinting [37]. This privacy divide creates a vulnerability gap, where the users least capable of protecting themselves are the most exposed.

B. Convergence of Fingerprinting Techniques

Browser fingerprinting, as explored through various methods in this paper (cf. Table I), represents a comprehensive and evolving threat to digital privacy. Each fingerprinting technique, from HTTP Header Attributes to sophisticated methods like Canvas, WebGL, and Audio Fingerprinting, offers unique data points, but their power lies in their combinatorial use. This synergistic exploitation of passive and active methods creates a multi-dimensional profiling system capable of identifying users with extraordinary precision and stability.

The cross-browser stability of WebGL and machine learning-based techniques enables tracking across different devices and sessions, while WebRTC Fingerprinting reveals network-level information like private IP addresses. These methods complement traditional fingerprinting approaches by exposing additional system and network data layers, making countermeasures significantly more difficult.

Furthermore, machine learning-based fingerprinting represents the next evolutionary step in this domain. Research by Wang et al. demonstrated that side-channel attacks exploiting CPU cache timing and memory consumption can identify users with up to 90% accuracy without relying on any standard browser attributes [57]. This convergence of fingerprinting techniques into multi-layered profiling systems renders current countermeasures increasingly ineffective.

C. Ethical and Legal Implications

The stealthy nature of browser fingerprinting raises significant ethical concerns regarding user autonomy and consent. Although the GDPR explicitly defines personal data as any information that can identify an individual, browser fingerprinting often circumvents this regulation under the guise of legitimate interest [62].

However, recent court rulings suggest a tightening regulatory landscape. In 2023, the French data protection authority CNIL fined Criteo for failing to obtain consent for fingerprinting-based tracking, marking one of the first legal cases explicitly addressing browser fingerprinting under GDPR.

Nonetheless, global regulatory frameworks remain fragmented, and the majority of fingerprinting scripts operate without user knowledge or legal verification. This regulatory vacuum risks turning browser fingerprinting into a normalized surveillance practice embedded within the digital economy.

D. Towards Privacy-Respecting Fingerprinting

While fingerprinting is primarily associated with surveillance, several emerging technologies seek to repurpose it for privacy-enhancing applications. Projects like BrFast [15] and Apple's Private Access Tokens leverage ephemeral, cryptographically unlinkable fingerprints to authenticate users without persistent tracking.

However, the implementation of privacy-respecting fingerprinting requires transparent system design and regulatory oversight. Without proper safeguards, even privacy-preserving systems risk reinforcing the same surveillance mechanisms they aim to replace.

E. Future Outlook

The future of browser fingerprinting lies in the convergence of machine learning, side-channel attacks, and cross-device tracking. This hybrid approach creates persistent, adaptive tracking systems capable of circumventing existing countermeasures.

Future research should prioritize:

- Developing adaptive defenses against machine learning-assisted fingerprinting.
- Investigating cross-device tracking prevention methods.
- Designing transparent fingerprinting APIs that separate security-related use cases from surveillance.
- Studying the privacy divide and ethical implications of fingerprinting on vulnerable populations.

F. Consequences

Browser fingerprinting represents one of the most pervasive and least transparent forms of online tracking. Its rapid evolution from basic HTTP headers to machine learning-assisted side-channel attacks highlights the growing asymmetry between users and data collectors. The convergence of passive and active methods creates a multi-dimensional profiling system that is increasingly resistant to countermeasures, challenging both privacy frameworks and user efforts to remain anonymous online.

Despite its invasive applications, fingerprinting could also be repurposed for privacy-enhancing authentication systems — provided that transparent design principles and strict regulatory safeguards are enforced. Bridging the gap between security and privacy will be one of the defining challenges of digital privacy in the coming decade.

V. Conclusion

In this final section, the paper synthesizes its findings to assess the broader impact of browser fingerprinting on digital privacy. It reflects on the dual-use nature of fingerprinting—both as a security tool and as a surveillance threat—and reaffirms the urgent need for stronger countermeasures, privacy-oriented browser practices, and regulatory interventions. The conclusion also identifies key areas for further research and policy action, emphasizing that safeguarding user anonymity in the digital space requires a coordinated effort between technologists, regulators, and informed users.

A. Summary of the Research Outcome

This contribution has examined browser fingerprinting, a growing technique in online tracking. It has demonstrated that browser fingerprinting is a sophisticated method for identifying and tracking users online without traditional methods like cookies.

The analysis highlighted that browser fingerprinting poses a complex challenge from both technical and privacy perspectives. While it provides companies and advertisers with detailed insights into user behavior for targeted advertising, it raises significant privacy concerns as users are often tracked without their knowledge or consent. Despite stricter privacy laws like the GDPR in the EU, browser fingerprinting remains a grey area. Anti-fingerprinting techniques are limited and continually evolving to keep up with new tracking methods.

In conclusion, browser fingerprinting plays and will continue to play a significant role in the digital landscape. Both users and regulatory bodies must increase awareness of browser fingerprinting practices and their implications.

B. Implications for Practice

Consent and Cookies: Always accept only the necessary cookies in cookie banners and regularly delete cookies to hinder tracking and fingerprinting. This is particularly important for news sites, which often misuse collected data without user consent.

Blending in with the Masses: Reducing APIs and data sources for fingerprinting can ironically make users more identifiable [63]. Thus, widely adopted browsers and protection mechanisms should be used to stay less conspicuous.

Browser Choice: Choose browsers with robust privacy protections. On iOS, Safari is recommended due to its advanced tracking protection and large user base [64]. For Android, the Mull browser is highly rated for fingerprinting protection, while Brave is a good, widely-used alternative. On desktops, Brave, Librewolf, and Mullvad browsers are recommended for their privacy features and user bases [65].

Browser Extensions: Limit the use of browser extensions, as they can become sources of unique information. While some extensions block known trackers or modify API outputs, these protections are often already built into recommended browsers like Brave and Librewolf [26] [63].

C. Future Research

Future research in browser fingerprinting should focus on several key areas. First, countermeasures and defense mechanisms need to be explored further, especially in mitigating the newer techniques that leverage machine learning and side-channel attacks. These advanced methods can bypass traditional privacy safeguards, such as disabling JavaScript or using incognito modes, making the development of more robust anti-fingerprinting technologies imperative. Additionally, research should explore the ethics and regulatory frameworks surrounding fingerprinting, examining how existing privacy and data protection laws like GDPR can be adapted to better address fingerprinting practices. Another promising direction is improving cross-device tracking prevention by understanding how fingerprinting works across different platforms and hardware. Lastly, investigating user awareness and educational tools on fingerprint privacy risks will help empower the general public to protect their digital identities more effectively. Thus, future research should focus on developing more effective privacy techniques to balance commercial interests and user privacy rights.

REFERENCES

[1] A. Lawall, "Fingerprinting and Tracing Shadows: The Development and Impact of Browser Fingerprinting on Digital Privacy," in *Proceedings of the IARIA SECURWARE 2024 Conference*. IARIA, November 2024, pp. 132–140.

[2] ——, "Fingerprinting and Tracing Shadows: The Development and Impact of Browser Fingerprinting on Digital Privacy," *arXiv preprint arXiv:2411.12045*, 2024.

[3] K. Mowery and H. Shacham, "Pixel perfect: Fingerprinting canvas in HTML5," *Proceedings of W2SP*, vol. 2012, 2012.

[4] K. Szymielewicz and B. Budington. (2018) The GDPR and Browser Fingerprinting: How It Changes the Game for the Sneakiest Web Trackers. Accessed: 2024-09-27. [Online]. Available: https://www.eff.org/de/deeplinks/2018/06/gdpr-and-browser-fingerprinting-how-it-changes-game-sneakiest-web-trackers

[5] D. Zhang, J. Zhang, Y. Bu, B. Chen, C. Sun, and T. Wang, "A Survey of Browser Fingerprint Research and Application," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 3363335, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/3363335

[6] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies*, M. J. Atallah and N. J. Hopper, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–18.

[7] S. Englehardt and A. Edelstein. (2021) Firefox 85 Cracks Down on Supercookies. Accessed: 2024-09-27. [Online]. Available: https://blog.mozilla.org/security/2021/01/26/supercookie-protections/

[8] E. Woollacott. (2021) Browser fingerprinting more prevalent on the web now than ever before. Accessed: 2024-09-27. [Online]. Available: https://portswigger.net/daily-swig/browser-fingerprinting-more-prevalent-on-the-web-now-than-ever-before-research

[9] R. Koch. (2019) Cookies, the GDPR, and the ePrivacy Directive. Accessed: 2024-09-27. [Online]. Available: https://gdpr.eu/cookies/

[10] U. Iqbal, S. Englehardt, and Z. Shafiq, "Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 05 2021, pp. 1143–1161.

[11] G. Acar. (2014) Browser Fingerprinting and the Online-Tracking Arms Race. Accessed: 2024-09-27. [Online]. Available: https://www.esat.kuleuven.be/cosic/news/the-web-never-forgets-persistent-tracking-mechanisms-in-the-wild/

[12] "Request header," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/Request_header

[13] "HTTP/2 fingerprinting: A relatively-unknown method for web fingerprinting," accessed: 2024-09-27. [Online]. Available: https://lwthiker.com/networks/2022/06/17/http2-fingerprinting.html

[14] "HTTP Headers - MDN Web Docs," accessed: 2024-07-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

[15] P. Laperdrix, W. Rudametkin, and B. Baudry, "Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 878–894.

[16] "User-Agent," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent

[17] "Accept - HTTP," accessed: 2024-02-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept

[18] "Content-Encoding - HTTP," accessed: 2024-02-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Encoding

[19] "Content-Language - HTTP," accessed: 2024-02-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Language

[20] A. Gómez-Boix, P. Laperdrix, and B. Baudry, "Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 309–318. [Online]. Available: https://doi.org/10.1145/3178876.3186097

[21] NGINX, "Managing request headers," accessed: 2024-04-22. [Online]. Available: https://nginx.org/en/docs/http/ngx_http_headers_module.html

[22] B. Wolford. (2024) What are the GDPR consent requirements? Accessed: 2024-09-27. [Online]. Available: https://gdpr.eu/gdpr-consent-requirements/

[23] "Navigator: plugins property," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Navigator/plugins

[24] "Extension Detector," accessed: 2024-09-27. [Online]. Available: https://github.com/z0ccc/extension-detector

[25] "How ad blockers can be used for browser fingerprinting," accessed: 2024-09-27. [Online]. Available: https://fingerprint.com/blog/ad-blocker-fingerprinting/

[26] S. Karami, P. Ilia, K. Solomos, and J. Polakis, "Carnus: Exploring the Privacy Threats of Browser Extension Fingerprinting," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.

[27] "CanvasBlocker," accessed: 2024-09-27. [Online]. Available: https://github.com/kkapsner/CanvasBlocker

[28] "WebGL: 2D and 3D graphics for the web," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

[29] T. Stephenson, "A Comparative Study on Analyses of Browser Fingerprinting," Ph.D. dissertation, Wesleyan University, 2023.

[30] "WebKit," accessed: 2024-09-27. [Online]. Available: https://github.com/WebKit/WebKit/commit/ae710d34c23858295b385e3f95ad7f6edd29f9d7

[31] S. Lee, Y. Kim, J. Kim, and J. Kim, "Stealing webpages rendered on your browser by exploiting gpu vulnerabilities," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 19–33.

[32] S. J. Vaughn-Nichols, "Vendors draw up a new graphics-hardware approach," *Computer*, vol. 42, no. 05, pp. 11–13, 2009.

[33] G. Nakibly, G. Shelef, and S. Yudilevich, "Hardware Fingerprinting Using HTML5," *arXiv preprint arXiv:1503.01408*, 03 2015.

[34] Y. Cao, S. Li, and E. Wijmans, "(Cross-)Browser Fingerprinting via OS and Hardware Level Features," in *Network and Distributed System Security Symposium*, 2017.

[35] Laor et al., "DRAWNAPART: A Device Identification Technique based on Remote GPU Fingerprinting," *ArXiv*, vol. abs/2201.09956, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246276013

[36] M. Mantel. (2022) Browser-Fingerprinting: PCs, Smartphones & Co. lassen sich über die GPU tracken. Accessed: 2024-09-27. [Online]. Available: https://www.heise.de/news/Browser-Fingerprinting-PCs-Smartphones-Co-lassen-sich-ueber-die-GPU-tracken-6345233.html

[37] J. S. Queiroz and E. L. Feitosa, "A Web Browser Fingerprinting Method Based on the Web Audio API," *Comput. J.*, vol. 62, pp. 1106–1120, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:86644316

[38] T. S. Brandes, S. Kuzdeba, J. McClelland, N. Bomberger, and A. Radlbeck, "Rf waveform synthesis guided by deep reinforcement learning," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.

[39] E. Cheek, D. Khuttan, R. Changalvala, and H. Malik, "Physical fingerprinting of ultrasonic sensors and applications to sensor security," in *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*. IEEE, 2020, pp. 65–72.

[40] S. Chalise and P. Vadrevu, "A Study of Feasibility and Diversity of Web Audio Fingerprints," *arXiv preprint arXiv:2107.14201*, 2021.

[41] S. Englehardt and A. Narayanan, "Online Tracking: A 1-million-site Measurement and Analysis," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1388–1401. [Online]. Available: https://doi.org/10.1145/2976749.2978313

[42] "Local Font Access API," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Local_Font_Access_API
[43] "Font Fingerprint Defender," accessed: 2024-01-20. [Online]. Available: https://mybrowseraddon.com/font-defender.html
[44] "Tracking Prevention in WebKit," accessed: 2024-01-20. [Online]. Available: https://webkit.org/tracking-prevention/
[45] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser Fingerprinting: A Survey," *ACM Trans. Web*, vol. 14, no. 2, apr 2020. [Online]. Available: https://doi.org/10.1145/3386040
[46] A. Reiter and A. Marsalek, "WebRTC: your privacy is at risk," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 664–669. [Online]. Available: https://doi.org/10.1145/3019612.3019844
[47] "DetectRTC," accessed: 2024-09-27. [Online]. Available: https://github.com/muaz-khan/DetectRTC
[48] "Fingerprinting WebRTC," accessed: 2024-01-20. [Online]. Available: https://privacycheck.sec.lrz.de/active/fp_wrtc/fp_webrtc.html
[49] V. Bernardo and D. Domingos, "Web-based Fingerprinting Techniques," in *Proceedings of the 13th International Joint Conference on E-Business and Telecommunications*, ser. ICETE 2016. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2016, p. 271–282. [Online]. Available: https://doi.org/10.5220/0005965602710282
[50] T. Bujlow, V. Carela-Español, J. Solé-Pareta, and P. Barlet-Ros, "A Survey on Web Tracking: Mechanisms, Implications, and Defenses," *Proceedings of the IEEE*, vol. 105, no. 8, pp. 1476–1510, 2017.
[51] L. Olejnik, C. Castelluccia, and A. Janc, "Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns," *12th Privacy Enhancing Technologies Symposium (PETS 2012)*, 07 2012. [Online]. Available: https://petsymposium.org/2012/papers/hotpets12-4-johnny.pdf
[52] N. Takei, T. Saito, K. Takasu, and T. Yamada, "Web Browser Fingerprinting Using Only Cascading Style Sheets," in *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, 2015, pp. 57–63.
[53] "CSS-Fingerprint," accessed: 2024-09-27. [Online]. Available: https://github.com/OliverBrotchie/CSS-Fingerprint
[54] "Feature Detection," accessed: 2024-01-23. [Online]. Available: https://privacycheck.sec.lrz.de/active/fp_fd/fp_feature_detection.html
[55] "Advanced Tor Browser Fingerprinting," accessed: 2024-09-27. [Online]. Available: http://jcarlosnorte.com/security/2016/03/06/advanced-tor-browser-fingerprinting.html
[56] "Investigate impact of fingerprinting via getClientRects()," accessed: 2024-09-27. [Online]. Available: https://gitlab.torproject.org/tpo/applications/tor-browser/-/issues/18500
[57] H. Wang, H. Sayadi, A. Sasan, P. D. Sai Manoj, S. Rafatirad, and H. Homayoun, "Machine Learning-Assisted Website Fingerprinting Attacks with Side-Channel Information: A Comprehensive Analysis and Characterization," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*, 2021, pp. 79–84.
[58] "WebAssembly," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/WebAssembly
[59] "High precision timing," accessed: 2024-09-27. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Performance_API/High_precision_timing
[60] "Browser Fingerprinting und das TDDDG: Erlaubt oder nicht? [Browser Fingerprinting and the TDDDG: Allowed or not?]," accessed: 2024-09-27. [Online]. Available: https://dr-dsgvo.de/browser-fingerprinting-und-das-ttdsg/
[61] G. Pugliese, C. Riess, F. Gassmann, and Z. Benenson, "Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, pp. 558–577, 05 2020.
[62] K. Szymielewicz and B. Budington. (2018) The gdpr and browser fingerprinting: How it changes the game for the sneakiest web trackers. Electronic Frontier Foundation (EFF). Accessed: 2024-09-27. [Online]. Available: https://www.eff.org/de/deeplinks/2018/06/gdpr-and-browser-fingerprinting-how-it-changes-game-sneakiest-web-trackers
[63] N. Al-Fannah and C. Mitchell, "Too little too late: can we control browser fingerprinting?" *Journal of Intellectual Capital*, vol. ahead-of-print, 01 2020.
[64] K. Kollnig, A. Shuba, M. Van Kleek, R. Binns, and N. Shadbolt, "Goodbye Tracking? Impact of iOS App Tracking Transparency and Privacy Labels," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 508–520. [Online]. Available: https://doi.org/10.1145/3531146.3533116
[65] X. Lin, F. Araujo, T. Taylor, J. Jang, and J. Polakis, "Fashion Faux Pas: Implicit Stylistic Fingerprints for Bypassing Browsers' Anti-Fingerprinting Defenses," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 987–1004.

Towards a Trust Management Approach Encompassing Stakeholders for the Automotive Ecosystem

Marco Michl, Hans-Joachim Hof
Technische Hochschule Ingolstadt
Carissma Institute of Electric, Connected and Secure Mobility
Ingolstadt, Germany
email: marco.michl@carissma.eu,
hof@thi.de

***Abstract*—The rise of connected services in modern vehicles, combined with the target of software-defined vehicles, makes new approaches to securing the automotive ecosystem necessary. One of these approaches is implementing computational trust models within vehicles to secure interactions in a way inspired by the intuitive concept of trust. Involved stakeholders and their relations are essential to creating a system representing trust. We identified relevant stakeholder groups involved in the communication of modern cars. We characterized them based on their lifecycle phase, the user agents and devices used to communicate, and their relations and roles. Furthermore, we describe the necessity for trust in the automotive ecosystem, the connection between trust and authorization, and the trust relations between the stakeholders. A formalization approach for the gathered knowledge about stakeholders and their characteristics is presented, utilizing a set-theory-based framework to review the definition of trust relations between stakeholders compared to proposed trust management systems. This approach shows that stakeholders in the automotive domain mainly gain their trust through their roles rather than their behavior. The difference between stakeholders and other entities is shown using the introduced framework. The provided stakeholder analysis, their roles in the automotive environment, and the formalization approach to linking stakeholders to trustworthy decisions are thus a basis for designing general trust management systems for the automotive ecosystem that cover multiple entity types.**

***Keywords*-automotive; ecosystem; trust; authorization; stakeholder; formalization.**

I. INTRODUCTION

Modern vehicles offer various services to their passengers and the surrounding area. The interaction with devices and infrastructure outside of the vehicle is essential for these connected services that use different technologies like Vehicular Ad-Hoc Networks (VANETs) or mobile networks. With the integration of these technologies, the vehicle is no longer an isolated device. It becomes part of the Internet of Vehicles (IoV), a term inspired by the Internet of Things (IoT) to describe the ecosystem built by interconnected vehicles that makes use of an IoT-like architecture [2][3][4]. The functions aim to provide traffic functions or increase traffic safety by contributing to driver assistance or autonomous driving functions.

Different stakeholders interact with the ecosystem in this network to use functions or fulfill services. In this context, a stakeholder is defined as a person or organization that is in some way affected by decisions or actions, influences them, or even considers itself to be affected [5][6]. As multiple stakeholders are involved in the automotive ecosystem, it is a multi-stakeholder system.

In this multi-stakeholder system, trust is a relevant concept that is necessary for cooperation. Although trust originally is more a sociological and psychological concept that eases or enables decision-making between persons, it can be stretched to interactions with non-natural entities [7][8]. It describes the relation between two entities: a truster that places trust in services, data, or the general behavior of a trustee. Therefore, the stakeholders and their relations must be known in order to evaluate and define trust in a system. This also involves relations in automotive use cases, where misplaced trust can have severe consequences due to safety implications.

In computer science, computational trust is closely related to authorization systems. This is reasonable, as trust is a concept to decide about cooperation, and authorization is similar to such a decision. Especially use cases where a truster has to determine whether or not to use data provided by a trustee are comparable to a trustful decision process [9]. Use cases similar to this model are becoming more common with the rise of IoV.

For this purpose, this work aims to identify relevant stakeholders in the automotive ecosystem, assign appropriate characteristics, and describe their trust relationships. This builds a basis for trust models in automotive systems that secure communication between stakeholders and automotive systems. Therefore, the focus is on stakeholders that use electronic communication, excluding, e.g., contractual relations between stakeholders. Furthermore, only standard series vehicles are in scope, and no special vehicles, like emergency, driving school, or shared vehicles with specific adaptations, are included. A further restriction concerns the focus on vehicles in the scope of UN Regulation No. 155 that introduces mandatory measures to handle cyber security in the automotive domain [10]. This restriction is applied as we use the lifecycle introduced by this regulation. However, the results are not significantly affected by this limitation.

Based on this stakeholder analysis, we propose a formal framework describing the involved parties, their relations, and their trust in this context. This framework is built on set theory and function descriptions to allow more specific statements. Utilizing the proposed framework, we show how trust management systems work and why they only focus on interactions between artificial agents rather than including stakeholders. We propose a simple yet effective method to integrate personal or organizational entities into trust management systems by applying binary trust values according to the stakeholders' roles.

In summary, this paper contains the following contributions:

- Provide an analysis of stakeholders in the automotive ecosystem, their user agents, and the lifecycle phases they are active in
- Show how trust amongst the stakeholders is established
- Discuss the applicability of trust management systems on relations that include stakeholders
- Introduce a formal framework to describe trust-based decisions that contains stakeholders, and show how stakeholder relations can be included in trust management systems

This work is a follow-up to our paper [1]. We extended it with the formal framework and the approach to integrating stakeholders into trust management systems.

The rest of this paper is organized as follows. Related work is presented in Section II. This review shows that no comparable analysis exists. The necessary characteristics to describe the collected stakeholders are developed in the third section. Based on these parameters, the stakeholders are presented in Section IV. The results of the trust relation analysis are followed in the next section before these findings are further analyzed in the formal framework in Section VI. Finally, our findings are evaluated based on an exemplary case study. The last section summarizes the content of this work and gives an overview of its further use and limitations.

II. Related Work

Originating from project management, a stakeholder describes a person or organization that can affect or is affected by a decision or an activity [5][6]. This involves all entities that interact with the system in any way. Following Kosch [11], automotive stakeholders are connected to this specific environment in different steps, like the development, production, or usage phase. Furthermore, stakeholders can be categorized into different groups. Marner et al. [12] conducted a stakeholder analysis that mainly involves different stakeholders within an Original Equipment Manufacturer (OEM).

A comparable analysis was performed by Gomez et al. [13] with a focus on automotive digital forensics. The involved entities are necessary in this domain as their requirements are fundamental to answering forensic questions. This study presents two general stakeholder survey approaches: the brainstorming method based on Bryson [14] and snowball sampling as introduced by King et al. [15]. Only the first seems applicable, as the stakeholders in automotive digital forensics involve criminals, making a snowballing method including all stakeholders impossible. Using various brainstorming sessions with experts, a list of relevant stakeholders and a Venn diagram describing their main interests was created.

Mansor collected stakeholders regarding security in the automotive ecosystem [16]. This work also proposes a trust model for the automotive ecosystem, incorporating the three stakeholders OEM, service or application provider, and vehicle driver or owner. The trust relations between these entities are described. This model does not focus on trust relations on a technical level but instead on an interpersonal level.

Knauss et al. [17] collected a list of stakeholders and their relations in the automotive ecosystem. They gathered their information in interviews at an OEM and mainly focused on the interactions during vehicle development. As such, they did not focus on the electronic communication between stakeholders in the automotive ecosystem.

In various articles, trust relations are described using formal methods. Douceur [18] utilized a set-theory-based method to describe an attack threatening decentralized systems, including trust management systems. A formalized description of the trust concept by Marsh [19] has led to much attention and research in this field. This work uses set theory amongst function signatures with specific value ranges and the combination of the introduced variables using mathematical functions. Another approach using set theory by Habib [20] introduces propositional logic elements to specify the connection between elements and set memberships. Despite the various existing methods, the framework we introduce in this work focuses on a particular part of trust management systems in the automotive domain and provides new insights.

To our knowledge, a collection of stakeholders in the automotive domain and their trust relations and communication interactions does not yet exist. This gap also means that there is no formalized description. The present work will close this gap.

III. Automotive Stakeholder Characteristics

Appropriate characteristics are necessary to describe and characterize the collected stakeholders. For this work, three factors are considered necessary to describe stakeholders in the automotive domain. These consist of the lifecycle phase of vehicles the stakeholder is involved in, the user agents or devices used for communication, and the stakeholders' rights and responsibilities.

A. Automotive Lifecycle

Vehicle and vehicle projects are divided into several lifecycle phases. These phases are suitable to describe stakeholders, as several only appear in specific phases, and because they also take on different roles in different phases [11]. In this work, we combine two different methods to structure the automotive lifecycle. The first describes the *vehicle lifecycle*, whereas the latter focuses on the *vehicle project lifecycle*.

Hawkins et al. conducted a lifecycle analysis of battery-electric vehicles and used the three lifecycle phases *production, use,* and *end of life* [21]. Their approach is aimed at individual vehicles that are produced, used, and ultimately reused or disposed of, describing the *vehicle lifecycle*.

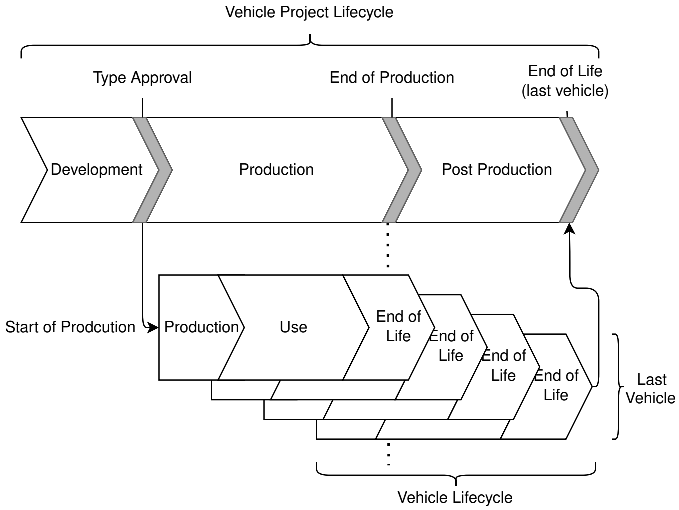


Figure 1. Vehicle Project and Vehicle Lifecycle in comparison.

The second approach targets vehicle projects, as the UN Regulation 155 does. In this regulation, the three phases *development, production,* and *post-production* are distinguished [10]. The phases seem similar to Hawkins' approach. Still, they cut the lifecycle of vehicle projects that are differentiated by the date of the type approval (between *development* and *production phase*) and the end of production date (between *production* and *post-production*). Individual end-user vehicles are only produced in the *production phase*. The last individual vehicle entering its *end of life phase* according to the *vehicle lifecycle* defines the end of the R155 *post-production phase*.

For this work, we assume stakeholders in both the vehicle individual and vehicle project-related lifecycle phases are relevant. Therefore, the generic lifecycle phases *development, production, use,* and *end-of-life* are utilized. We note that during the *development phase*, no publicly visible and customer-used vehicles are available. The *post-production phase* used in UN R155 is a phase to structure activities regarding the cyber security of cars after the *end-of-production* while vehicles are still in use. We argue that no additional stakeholders are involved in this phase compared to the *production phase*. Therefore, that phase is not considered explicitly in this work. Figure 1 overviews the lifecycle phases used.

B. User Agents used by Automotive Stakeholders

This work focuses on the security of the automotive ecosystem. As such, the electronic communication between the stakeholders and the communication within the automotive ecosystem is of central interest. As the presented stakeholders are natural, organizational, or legal entities, they use devices or interfaces for their electronic communication. As proposed by Kuschel in [22], we expand the vehicle to an interconnected automotive ecosystem that is used by various stakeholders to fulfill their workflows. This ecosystem consists of connected and communicating devices, which the stakeholders can use to interact with the ecosystem and other stakeholders. The ecosystem does not consist only of devices. Therefore, we use the term user agents for the relevant components of the automotive ecosystem, as stakeholders can utilize them for their communication.

TABLE I. USER AGENTS USED FOR COMMUNICATION IN THE AUTOMOTIVE ECOSYSTEM

User Agent	Description
Vehicle	Systems and ECUs contained inside the vehicle.
Backend	Applications on servers accessed online, often operated by the OEM or service providers. This user agent is distinct from frontends in the way that, in this case, the specific operator of the backend service accesses the service.
Diagnostic Devices	Devices used to interact with the vehicle's diagnostic system. Operations going beyond the legally prescribed actions like OBD [23] often require vehicle-specific information, which the OEM must also provide to non-affiliated workshops [24].
Frontends	Frontends for services accessed through the internet, including mobile apps.
RSUs	Devices located near street infrastructure that directly communicate with vehicles using VANETs.
Charging Station	Infrastructure to charge electric or hybrid vehicles.

These agents are listed in Table I and form a part of the automotive ecosystem. The list was created based on the stakeholder analysis and the evaluation of exemplary use cases originating in different lifecycle phases, like vehicle usage by end-users, online and workshop updates, the setup of new vehicles by customers, etc.

User agents must enable stakeholders to take on different roles based on their respective rights, which depend on the lifecycle phase.

C. Responsibilities and Rights in the Automotive Ecosystem

Interactions in the automotive ecosystem should only be possible if the acting stakeholder is allowed to make them. This authorization depends on the stakeholder, action, and context. One part of the context is the lifecycle phase the vehicle (project) is in. As such, the responsibilities and rights of automotive stakeholders are relevant characteristics and are, therefore, added to the stakeholders' description.

A simple but frequently discussed example of authorization is the application of software updates. While only the OEM can release and publish software for a vehicle, it is up to the owners of the cars to have it installed, as it entails a permanent change to the vehicle's condition. However, this division of tasks is only relevant in the use phase, as during development, the OEM itself has all rights to the pre-series vehicles and can, therefore, decide on changes to the condition itself. In the use phase, the authorizations to release and install software are divided among stakeholders, where the OEM maintains its products, but the owner decides on their property.

The vehicle ecosystem has to handle the relevant roles and responsibilities and consider changes within them if the lifecycle phase or, e.g., the ownership of the vehicle changes.

Otherwise, the ecosystem might not be able to correctly reflect contractual or business relations, leading to possible vulnerabilities. As this work provides an overview, such specific vulnerabilities are not in scope.

IV. Automotive Environment Stakeholders

The set of stakeholders, their relations, and interactions presented here was created using a comparable method as Gomez et al. [13] based on Bryson [14], as multiple brainstorming and discussion sessions, including various participants, were conducted. The stakeholders involved in the different lifecycle phases were collected within these sessions, and their roles were discussed. The participants included several employees of an automotive supplier, two employees of a start-up in the domain of decentralized identities with connections to OEMs and various suppliers, members of an automotive security research group partially with a background at different OEMs as well as a Professor researching in the automotive security domain.

Table II provides an overview of the stakeholders in the automotive ecosystem, the lifecycle phase they are active in, and the user agents they are using. The following section discusses the rights and responsibilities of each stakeholder.

a) OEM: During the development phase, the OEM is the driving force behind the development project, is responsible for its overall success, and bears the risk. This responsibility also means that the OEM has all the rights regarding communication and authorization in the ecosystem. These rights change when the vehicle is handed over to the customer. After that, the OEM no longer has direct physical access to the vehicle and can only communicate with connected vehicles via its backend. Indirect access is possible using the workshops, which receive instructions and tools for maintenance and repair from the OEM. The authorization to release changes to the vehicle, for example, through updates or modifications, can only lie with the OEM, as it must ensure compliance with regulations. The OEM remains involved after the utilization phase, as the reuse of components must be planned, for example, for second-life applications of batteries [25] or the use of spare parts from old vehicles, which may have to be approved for reuse in other vehicles [26].

For development, the OEM uses all clients that will be used in the later usage phase, even if only for testing purposes, as with RSUs. In later phases, direct communication between the OEM and the vehicle is only possible via the manufacturer-specific backend.

b) Supplier: OEMs develop new cars with the help of multiple suppliers. As supply chains get more complex, a distinction between different suppliers (Tier 1-3) is commonly used [17][27]. Suppliers get the task of developing, integrating, and supplying certain vehicle parts according to the requirements of the OEM. Their deliverable includes hardware (e.g., mechanical parts, ECUs) or software. With the shift from hard- to software-defined functions in vehicles [28] and the target of software-defined vehicles, together with the shift to more centralized E/E architectures [29], different suppliers need to work closely together to develop their functions.

How suppliers interact with the automotive ecosystem depends on the function they provide. There is no communication between the supplier and the ecosystem for mechanical parts, and there is no further interaction after the part's delivery during the production phase. For software functions, there are often additional activities for updates provided by the supplier or even direct interactions with the ecosystem in case of connected functions, such as if the supplier operates backend services or cooperates with service and content providers. The final diagnostic devices are utilized while developing the development interfaces of ECUs, especially in later development steps. This interface is provided by the OEM to enable suppliers to fulfill their tasks.

The limited communication between suppliers and the ecosystem reflects the supplier's rights in the use phase. As the vehicles' later users mainly interact with the OEM, and the OEM covers its suppliers, they do not have explicit, own rights or responsibilities in the ecosystem.

c) Development Service Provider: For certain activities during development, OEMs commission Development Service Providers to execute tasks, e.g., to test functions or devices regularly. For their activities during the development, the OEM grants them access to necessary parts of the ecosystem that can include all the systems an OEM also uses. They do not have explicit rights or responsibilities, especially not in later lifecycle phases.

d) Service and Content Provider or Operator: Modern, connected vehicles consume information from outside the vehicle and deliver their data to external services, forming the automotive ecosystem. To do so, data is provided by service providers, and infrastructure, such as mobile networks, RSUs or charging stations, are utilized that are operated by their operators. For the development of the connected services and the integration into vehicles, these stakeholders are involved in the development and production phase. During the use phase, they provide services, communicate with the vehicles, and are part of the vehicle ecosystem. Services are then mostly offered to the vehicle user, including specific rights and responsibilities according to their services.

e) Owner: Owners of vehicles are a heterogeneous group of stakeholders. Vehicles are owned either privately or for business. Business owners may again use cars for their business or provide them to others, e.g., car rental or sharing companies. Owners are distinct from the driver or user of the vehicle. Therefore, only fleet owners are considered in this study, as they can use special fleet services to manage their vehicles, although they do not directly use them. In this case, access to the vehicle ecosystem is possible through the frontends of fleet services. Furthermore, in the context of this work, the owner is regarded as the primary holder of the rights to his vehicle during the use phase, so the owner must authorize any changes. This assumption is subject to a restriction if the owner is the lessor of the vehicle and transfers it to the lessee

TABLE II. Stakeholders involved in the automotive ecosystem. An "X" marks the lifecycle phases this stakeholder is involved in as well as the user agents that are utilized.

Stakeholder	Phases				User Agents						Description
	Development	Production	Use	End of Life	Vehicle	Backend	Diagnostic Device	Frontends	RSUs	Charging Stations	
OEM	X	X	X	X	X	X	X	X		X	Develops, produces and sells the vehicle and is furthermore responsible for providing updates, service instructions, and service access
Supplier	X	X			X		X			X	Develops, manufactures, and delivers hard- or software for the product according to the OEM's requirements
Development Service Provider	X				X	X	X	X		X	Supports the OEM during the development by taking on specific tasks, especially testing
Service and Content Providers or Operators	X	X	X			X			X	X	Offer, adapt or develop services, that are integrated into the later product
Owner			X					X			Legally owner of the vehicle
Driver			X		X			X		X	Entity using the vehicle to drive
Workshops			X	X	X		X	X			Authorized and free workshops offering maintenance and repairs for vehicles
Authorized Test Organizations			X				X				Organizations authorized to verify the conformity of vehicles, e.g., in the PTI
Recycler				X	X		X				Manages recycling and disposing process

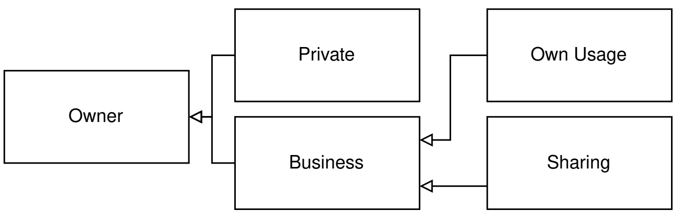


Figure 2. Different types of vehicle owners are divided into private and business owners. Business owners can use the vehicle for their own mobility or provide it as a rental or sharing company.

in its entirety. An overview of vehicle owner types is given in Figure 2.

f) Driver: Drivers are the actual users of the vehicle. They directly interact with the vehicle, its interfaces, and the frontends intended for end-users. Due to the distinction with owners, drivers have permission to use and drive the vehicle as intended, but they are, e.g., not allowed to manipulate or change the vehicle permanently.

g) Workshop: During the use phase, vehicles require workshops for maintenance and repairs. Electronic communication between the workshop and the vehicle becomes vital with more software functions. OEMs provide special equipment to access the necessary diagnostic interfaces. Due to legal reasons, access to these tools has to be given to independent workshops and must not be restricted to OEM partner workshops [24]. The owner authorizes the workshops to conduct repairs and maintenance, although this authorization is not currently represented in electronic communication.

h) Authorized Test Organization: To ensure the safety of vehicles on public roads, in various countries PTIs are legally required. Authorized Test Organizations carry these out. Communication with the vehicle is necessary during the test procedures, e.g., to access emission-related data via OBD [30].

i) Recycler: At the end of a vehicle's life, recyclers take care of its disposal and reuse. This also requires communication with the vehicle, for example, to trigger the end-of-life function of airbags, which releases the pyrotechnic elements and thus renders them harmless. This is done either via the vehicle's diagnostic system or by direct communication with the airbag control unit [31].

V. Trust Relations in the Automotive Ecosystem

Trust is a characteristic of the relationship between two entities. In the computational trust domain, these entities are not restricted to be humans or organizations, they can also be devices equiped with algorithms that enable them to make decisions based on algorithms mimicking trust. In the automotive domain, three types of trust relations exist: trust between two stakeholders, natural or organizational entities as described in the introduction's description, one stakeholder and a device within the automotive ecosystem, and two devices of the automotive ecosystem.

The target of trust is to make decisions for or against cooperation, although one's own welfare depends on the decision and the behavior of another entity that can neither be controlled nor whose behavior can be predicted with certainty [7]. As such, it is closely related to authorization.

In the automotive ecosystem, such a mechanism can be embedded in an ECU that checks, e.g., the signature of a firmware update before installing it. In this case, the policy tests whether the firmware was signed with a specific key. For example, the OEM controls the necessary private key. This is reasonable, as the OEM is responsible for providing updates and keeping a vehicle safe and secure. The vehicle, therefore, trusts the OEM to provide firmware updates. In this simple use case with only one stakeholder, the OEM is also responsible for specifying and implementing the trust relation. The OEM must also include other relations, providing a particular gatekeeper position.

Trust always has to be considered in a specific context. As the vehicle trusts the OEM in the example above to provide valid software updates, the OEM is not authorized to open the vehicle in the use phase. The vehicle should not trust or follow a request by the OEM to open the car unless it was authorized to do so by the owner or driver of the vehicle. Such a use case becomes relevant if vehicles include functions to unlock them remotely.

Both examples describe an authorization scenario in which the vehicle, as part of the vehicle ecosystem, trusts a stakeholder in different contexts. The stakeholders' responsibilities and roles clearly define the trust relation.

For the sake of completeness, two examples of relations between stakeholders and between devices are given. The function "plug and charge" is considered for the first-mentioned. This function allows payment to be processed without the user's additional authentication. The user stores their data in the vehicle, which authorizes the charging station operator to process the payment. For the second category, direct communication between vehicles in VANETs can be considered, in which vehicles exchange information. No stakeholder is directly involved, and a trusting relationship arises between the two vehicles.

The following gives trust relations between the relevant stakeholders for each lifecycle phase.

a) Development Phase: The various stakeholders in the development phase are all authorized by the OEM responsible for the development process. Therefore, the OEM alone has the right to allow other stakeholders to communicate with the automotive ecosystem. The connections within the automotive ecosystem are also governed by the OEM that has complete control over the ecosystem in this phase. Trust relations between stakeholders and the ecosystem devices of all categories are managed by the OEM.

b) Production Phase: The structure of responsibilities in the production phase is similar to the development phase. The OEM is responsible for orchestrating the cooperation of involved suppliers, service, and content providers that might have to cooperate during production. For example, a Mobile Network Operator (MNO) might have to prepare the cellular network module during production. Again, the relations and the access are managed by the OEM.

c) Use Phase: When the vehicle is handed to the owner, there is a shift in the responsibilities and role structure. The OEM no longer has control over the entire ecosystem. Instead, the owner has extensive rights over its property and can, therefore, also determine which other stakeholders should interact with it. Beyond the scope of this work, it is necessary to discuss the extent to which vehicle ownership and physical control also justify exclusive rights concerning electronic interactions and to what extent a manufacturer may legitimately restrict these rights through End-User Licence Agreements (EULAs), particularly for services offered. Relations in the other direction are also possible, as service providers can authorize drivers to consume their services based on subscriptions.

More complex relations are possible as well. If we consider an OEM that releases maintenance instructions that have to be performed, the workshop usually receives them within their diagnostic systems. The owner can then authorize the workshop to execute these tasks.

As the rights in this phase are more distributed between stakeholders, this can lead to conflicts. An example of such a conflict led to the right-to-repair movement, where OEMs were forced to provide repair instructions and tools to free workshops alongside their partner workshops [24]. The regulation stated that the owner can decide which workshop should perform maintenance and repair tasks. In contrast, some OEMs wanted to restrict them to authorized workshops by withholding necessary tools. The access to the automotive ecosystem for third parties, as, for example, test organizations are, is often only possible by regulations that force OEMs to provide interfaces. As these interfaces are provided by regulation, there is no real trust or authorization connection between different stakeholders. From the automotive ecosystem perspective, all interactions compliant with the regulations are authorized.

d) End of Life: During the end-of-life phase, the disposal and reuse of the vehicle are the focus. OEMs have to enable the reuse of electronic vehicle parts that workshops can reinstall. Recyclers are responsible for safely disposing of parts that are not directly reusable and, therefore, need to communicate with the vehicle to disengage the airbags. The necessary interface for this interaction is based on regulation and, thus, does not have to be authorized by the OEM, and there is no real trust relation.

VI. Formal Framework

We create a formalized description of our findings to enable a more precise description of the entities and trust relations in the automotive domain. For this purpose, we utilize set theory to describe the different stakeholder groups, the user agents they use, and the lifecycle phases. These sets are then combined to explain the connection between these elements. A general approach to defining trust relations in the automotive domain is presented to show the integration of stakeholders in such a system. The difference between stakeholders and artificial agents implementing trust management algorithms is discussed, and a method to integrate both into a common system is presented.

A. Definition of Basic Sets

The formalization approach starts with the definition of various sets that describe the findings of the stakeholder analysis. The first set describes the stakeholders in the automotive domain. We define a set S that includes all stakeholders in the automotive domain. S has various subsets, one for each identified stakeholder group. Each of these groups is defined as a proper subset of S, as multiple of them exist, while most of the subsets (except for S_{Owner} and S_{Driver} that may include the same element) are disjoint.

- $S_{OEM} \subset S$ for OEMs
- $S_{Supplier} \subset S$ for the suppliers
- $S_{DSP} \subset S$ for Development Service Providers
- $S_{SCP} \subset S$ for Service and Content Providers or Operators
- $S_{Owner} \subset S$ for Owners, possible subsets for Private or Business Owners
- $S_{Driver} \subset S$ for Drivers
- $S_{Workshop} \subset S$ for Workshops, with possible subsections for free and authorized workshops
- $S_{ATO} \subset S$ for Authorized Test Organizations,
- $S_{Recycler} \subset S$ for Recyclers

As discussed in the stakeholders' description, the set S_{Owner} of owners can be divided into various subsets describing different, more specific vehicle owners. Following the structure in Figure 2, the set S_{Owner} includes the following subsets:

- $S_{PrivateOwner} \subset S_{Owner}$ for private vehicle owners
- $S_{BusinessOwner} \subset S_{Owner}$for business vehicle owners
  - $S_{DirectBusinessOwner} \subset S_{BusinessOwner}$ for business vehicle owners that directly use vehicles on their own
  - $S_{SharingBusinessOwner} \subset S_{BusinessOwner}$ for business vehicle owners that share their vehicles, e.g., to make money with them

Using this distinction, a more precise and detailed analysis of trust relations is possible.

Next, the utilized user agents are defined. For this, the subsets of A are defined as follows, representing a subset per identified user agent category:

- $A_{Vehicle} \subset A$ for vehicle client
- $A_{Backend} \subset A$ for backend systems
- $A_{DiagnosticDevice} \subset A$ for diagnostic devices
- $A_{Frontend} \subset A$ for frontend systems
- $A_{RSU} \subset A$ for RSUs
- $A_{ChargingStation} \subset A$ for charging stations

An equal definition is made for the lifecycle phases within the set $L = \{L_{Dev}, L_{Prod}, L_{Use}, L_{EOL}\}$:

- $L_{Dev} \in L$ for the development phase
- $L_{Prod} \in L$ for the production phase
- $L_{Use} \in L$ for the use phase
- $L_{Use} \in L$ for the end-of-life phase

In defining the basic sets, we distinguish between the stakeholder groups and the user agents as subsets and the lifecycle phases as members of their respective sets. For S and A, the defined groups or categories are subsets containing the specific elements, like a particular vehicle owner in S_{Owner} or a physical car in $A_{Vehicle}$. The set of lifecycle phases is a closed set with the four defined elements described in Section III-A.

B. Basic Mappings

The introduced set definitions allow us to formalize the basic connection between S, A, and L that have already been discussed.

Coming from the basic definitions in the previous section, we note that single elements out of sets or subsets are often used. The elements can be part of any subsets defined for the used sets if not specified further. For example, $s \in S$ means that s is any element in S, so it may be an element in S_{OEM}, $S_{Supplier}$, or any other subset.

One result of the stakeholder analysis is defined in Table II. It includes which stakeholder is active in which lifecycle state and which user agents are utilized by this stakeholder. A mapping describes the first outcome is

$$f_1 : S \to \mathcal{P}(L), \quad s \mapsto f_1(s) \subseteq L \tag{1}$$

where a stakeholder $s \in S$ is mapped on the power set of L, describing the lifecycle phases in which this specific stakeholder is active.

Next, the mapping

$$f_2 : S \to \mathcal{P}(A), \quad s \mapsto f_2(s) \subseteq A \tag{2}$$

describes which user agents a stakeholder utilizes.

These two mappings are also contained in Table II. New statements can be created if two parameters are combined.

$$\begin{aligned} g_1 : S \times A \to \mathcal{P}(L), \\ (s,a) \mapsto g_1(s,a) \subseteq L \quad \forall (s,a) \in S \times A \end{aligned} \tag{3}$$

$$\begin{aligned} g_2 : S \times L \to \mathcal{P}(A), \\ (s,l) \mapsto g_2(s,l) \subseteq A \quad \forall (s,l) \in S \times L \end{aligned} \tag{4}$$

The mapping 3 describes more precisely the lifecycle a stakeholder utilizes a specific user agent, 4 the user agents a client uses in a particular lifecycle phase. These mappings have been described in textual, the provided mappings can help to make more precise statements. An example is given for the stakeholder S_{OEM} that is active in the following phases and utilizes the following user agents:

$$1 : f_1(s) = \{Dev, Prod, Use, EOL\}, \quad s \in S_{OEM} \tag{5}$$

$$2 : f_2(s) = A \setminus A_{RSU}, \quad s \in S_{OEM} \tag{6}$$

A more specific statement can be created following the mapping 4.

$$\begin{aligned} 4 : f_2(s,l) = \{A_{Backend}, A_{Frontend}\}, \\ s \in S_{OEM}, l = \{L_{Use}\} \end{aligned} \tag{7}$$

The statement in (7) shows that an OEM has only a minimal possibility to access the vehicle ecosystem in L_{Use}.

C. Trust Relationships

This section discusses trust between entities in the automotive domain, especially the difference between the integration of stakeholders and other entities in these mappings.

Trust, as a characteristic of the relationship between two entities, is represented by the following term:

$$T_{x,y,z}, \quad x, y \in E, \quad z \in Z \tag{8}$$

Trust (T) is described as characteristic of the relation between a truster (x) and a trustee (y) out of a set of entities E in a situation or context (z), where $Z = \{z_1, z_2, ..., z_n\}$ describes the set of all possible contexts. As trust is a directed, not necessarily mutual relation between truster and trustee, $T_{x,y,z}$ is not necessarily $T_{y,x,z}$ [32]. If so, this results from the used model rather than being implied by the definition of trust.

An action can define a context, like opening a vehicle, retrieving information about a car, or changing its configuration through maintenance actions. Various other attributes can be considered in a context, like the time, location, or objects relevant to the action. As an example, a workshop (truster) might be authorized by the owner (trustee) to perform a specific software update (action) at its workshop (location) on the owner's car (object). For the simplicity of this work, we do not further distinguish the elements defining a context. As we focus on the role of stakeholders and their used agents, the three parameters of trustee, truster, and context are used here. However, other authors distinguish the various elements of the context and also see trustee and truster as part of it [33].

As stated earlier, truster and trustee do not necessarily have to be a natural or organizational entity. The stakeholder analysis was based on these two types of entities. Therefore, x, y are not necessarily contained in the set S, but in a more extensive set, described as the set of entities E, where $S \subseteq E$. $x, y \in S$ is valid for trust relations between two natural or organizational entities, as is the case for an owner authorizing a workshop to maintain a vehicle.

Electronic communication is especially interesting from a security point of view. In such situations, the communication between several entities has to be secured according to the trust relations of all involved entities. Next to the stakeholders S, other entities can be software functions in the user agents that provide functions to the stakeholders or for different user agents. In this case, the user agents do not act as simple clients to enable interaction with the automotive ecosystem for the stakeholders; they act as entities or agents themselves in a specific context.

The calculation of trust values in such systems can use various input parameters. Systems utilizing direct trust based on direct interactions between these entities can learn from the trustee's behavior and adapt the trust value according to past interactions, as the natural concept of trust does. Zhang et al. based their trust management system only on direct trust, as this does not require the definition of additional communication that might be hard to establish in the automotive domain [34]. Indirect measures can be used if other nodes share their experiences with an entity with others. The systems proposed in [35][36][37][38][39] are examples of the combination of direct and indirect trust paths. Although beneficial, especially in the VANET domain with highly dynamic network topology and high mobility [40][41][42][43], additional vulnerabilities can be introduced with wrong recommendations. Such attacks are often defined as good or bad-mouthing [44]. Subjective opinions are a core concept of trust, describing that every truster can trust a different trustee differently. Some proposed systems should be characterized more correctly as reputation systems, as reputation is a term for the public, cumulative knowledge about an entity's trustworthiness [45][46][47]. The possibility of forming a common opinion through consensus protocols without a central authority has led to a multitude of reputation systems in which the reputation values are formed using Distributed Ledger Technology (DLT) [48][49][50][51][52][53][54]. Apart from subjective, local values, the decisions in such systems are at least partially based on a globally synchronized value.

Apart from the behavior of nodes, other characteristics like their capabilities or competence can be integrated [46].

D. Trustful Decisions

Trust is a relationship characteristic used to decide if two entities cooperate. Restricted to electronic communication, binary decisions must usually be made. A trustful decision mechanism's binary output O is defined as follows.

$$O = \{allow, deny\} \tag{9}$$

We use the terms $allow$ and $deny$ here on purpose to illustrate the similarity to authorization mechanisms that use multiple input parameters to decide whether a service or resource (object) can be accessed by the subject in the requested way. Regarding trust management, the subject trying to access a service or resource is the trustee, the governing authority of the resource is the truster.

Following Jøsang in [9], the truster does not have to provide a service to the trustee. In some applications, like in VANET, the nodes share information with all their neighbors. Based on the application, the information source, and other attributes, each node decides whether the information is used or if the message is ignored. In this case, the node sharing the information is the trustee, and the consumer is the truster. The difference between traditional authorization systems and trust management and decision systems is that trust usually involves a subjective opinion of trustworthiness that might differ from entity to entity, and the truster's welfare somehow depends on the trustee. These characteristics are generally not given in authorization systems. These systems usually define a global policy utilizing objective attributes, as if someone holds an authorization token issued by a specific entity or provides valid credentials for which the necessary access rights are defined. In these systems, authorized entities can still be untrustworthy, for example, if inside attackers are considered [55].

An approach to defining a trust-based decision is given in the following function definition:

$$T_{x,y,z} \times \mathcal{P}(M) \rightarrow O, \quad (10)$$

In a trust-based decision process, the trust value between two specific entities x and y for this defined context z is used together with zero or multiple more attributes, where M establishes the set of all attributes.

The truster considers attributes when transferring a trust value into a trusting decision. Values considered here are usually not included in the trust value calculation. For example, Marsh [19] proposes quantifying competence, risk, and importance to decide how high the trust value has to be to enter cooperation. Mayer [56] regards risk as a central component in decision-making: a trust value reflects the risk that the trustee is willing to take, whereas the decision means that the trustee is taking the risk. This complies with the reliability and decision trust terms introduced by Jøsang [45].

One or more thresholds are a fundamental decision method based on trust value. As in many systems, a range is used as a metric for the trust, like $T_{x,y,z} = [0, 1]$ in [48][57][49][58][50], a threshold in this range can decide whether cooperation is entered or not. In some systems, multiple thresholds reflect decisions based on one trust value [38][53][52]. This can be reasonable if multiple alternative cooperation types or attributes are possible with more or less risk for the truster or if specific trust values trigger additional activities, like exclusion from the network [53]. Defining thresholds can be a complex task, so advanced methods like machine learning [59] or fuzzy set theory are applied by some authors [60]. The latter reflects uncertainty in categorizing trustees according to their trust values.

E. Binary Trust Values to Integrate Stakeholders

As defined above, a trust value describes a subjective opinion on the trustee's trustworthiness to behave as the truster expects. Various input parameters can be considered to determine these trust values.

This definition does not entirely comply with integrating stakeholders into trust management systems. In this case, the main reason for trust relations is the membership of a stakeholder group. For example, a workshop is authorized by the OEM to conduct maintenance measures according to the OEM's instructions because a workshop is defined as authorized to do so. A vehicle's owner can access the vehicle's data on the frontend because of its role. The trust relations involving a stakeholder are binary, as an entity either is in the necessary role for an action or is not. The trust value can only have two possible states in such a relation.

$$T_{x,y,z} = \begin{cases} 0, & \text{if } y \in S_a, S_a \text{ is authorized for } z \\ 1, & \text{if } y \notin S_a, S_a \text{ is authorized for } z \end{cases} \quad (11)$$

A ruleset based on contractual and legal requirements defines many trust relations in the automotive domain that include stakeholders. These requirements do not integrate a measure of the stakeholder's behavior or gain benefit from analyzing other attributes.

The integration of stakeholders into trust management systems can be achieved by applying the trust values given in Equation (11) to the stakeholders in specific contexts. These trust values, describing some blind trust and therefore not following the definition of trust relations [19][7] as the truster does not have a real choice, can reflect the permissions based on the stakeholder's role. As blind trust is used for a trust relations the truster is not questioning, in this case the term *given trust* suits better, as the trust relation is defined outside the context of truster and trustee by external, often contractual or legal conditions. The thresholds or other methods to make decisions on trust values must use this unconditional trust and always allow cooperation in necessary contexts, or deny it if the trustee does not have the required role. In these contexts, the system is comparable to a Public Key Infrastructure, which can be used to reflect unconditional attributes bound to an entity.

The proposed integration benefits from combining all types of trust relations in a common trust management system for the automotive domain. The trust value is used to decide whether or not to enter into cooperation. This trust value is calculated beforehand based on various input parameters, which may include the trustee's role. If such a role affiliation is relevant for a context, the binary determination of a trust value ensures the decision is made under this affiliation.

As part of the evaluation, we provide an example of how to implement the integration of a stakeholder into a trust management system.

VII. Evaluation and Exemplary Case Study

The results from this work are evaluated in various ways. First, stakeholders were discussed in different groups consisting of people working in the automotive domain and researchers in the automotive security domain. Secondly, exemplary scenarios were considered, and the stakeholders involved and their interactions were compared with the previous results. An excerpt of these scenarios is briefly presented below. The scenarios were selected to represent various trust relations, including different types of entities, as described in Section V.

The proposed formalization and integration of stakeholders into trust management systems in the automotive domain is described for each scenario.

a) Online Software Updates: In an online software update, the OEM provides new software for vehicle components that is usually downloaded over a backend connection and is installed without additional diagnostic equipment at the customer's location. In this case, the OEM is responsible for the overall process and approves the software before it is made available. Software may be supplied by suppliers but is tested and released by the OEM. Infrastructure operators are also included in the scenario to provide necessary services. Either the vehicle's owner or an authorized user usually approves the installation. Finally, workshops are involved in case the installation fails. Additionally, inspired by the terms of dis- and untrust introduced by Marsh et al. [61], a trust relation between the owner and the OEM might not even be necessary, as the owner may not have a choice other than installing mandatory updates, otherwise risking the shut down of the vehicle.

In a formal way, an entity $o \in S_{OEM}$ has the role of approving and releasing software installed on vehicles. Regardless of the developer of an update, the OEM is legally responsible for ensuring the safety of the software. The owner of a vehicle $e \in S_{Owner}$ has the right to decide what modifications are applied to his property. In a policy defining the decision, if an update is applied, the update client in the vehicle may refuse to install an update that is not signed by an entity $o \in S_{OEM}$ that has a specific role, e.g., *release-sw-update* or if the update is not wanted by the owner $einS_{Owner}$ of that precise vehicle, that has a role like *approve-sw-installation*.

b) Plug and Charge: The plug-and-charge scenario has already been briefly discussed in the trust section. In this case, the OEM has to provide necessary functions in the vehicle and the connected services (back- and front-end) to store the required information of a financial service provider that handled the payment. The driver then authorizes a charging station provider to request charging fees from the financial service provider.

To allow this use case, the vehicle owner or user has to enter into a contractual relationship with a financial service provider. The service provider then allows the owner's vehicle to charge while handling billing. In a document describing this relation, in a technical implementation, the financial service provider issues some kind of certificate, which the charging station then accepts to start charging. These relations are based on roles and contractual relations that can be implemented in a trust management system with the described approach.

c) VANETs: VANETs are a special network in which vehicles, RSUs, and other devices like mobile devices owned by Vulnerable Road Users (VRUs) communicate directly to exchange information about the current environment to enable cooperative driving functions or to increase road safety. In this scenario, devices within the automotive environment may communicate without the participation of a stakeholder. Involvement of service and infrastructure providers, operators, and drivers is possible, as advertised services are contained in the standardization of VANETs. Trust relations are interesting in this scenario, as no clear and pre-defined interactions exist in this ad-hoc network. Because of this, many automotive trust management systems concentrate on VANET applications [62].

Applying the proposed integration of stakeholders is unnecessary here, as no stakeholders are directly involved in the communication. This is one reason why trust in VANETs is so extensively analyzed, as there are no binary, pre-defined relations.

VIII. Conclusion and Future Work

Trust is an essential concept necessary for decision-making between people. The stakeholders involved and their relations must be known in order to evaluate trust and develop trust management systems in the automotive domain. As a comparable analysis did not yet exist, the relevant stakeholders have been collected in multiple sessions with different people working or researching in the automotive and automotive security domain. The interactions and trust relations between the collected stakeholders were determined by analyzing relevant use cases. To characterize the stakeholders, the lifecycle phase of vehicles in which they are involved, the user agents or devices they utilize to communicate in the automotive ecosystem, and their roles and responsibilities were used. The gained insights are used in a formalized framework to represent the findings more specifically. Based on the formal framework, the difference between trust relations between artificial agents, as studied in many trust management systems, and trust relations, including stakeholders, is discussed. An approach to integrate both types in a common system is presented by applying trust values to stakeholders at either end of the value range.

The stakeholders and their descriptions are general to provide an overview of the automotive domain. Although this was necessary for this work, it is a limitation, as in some scenarios, the same stakeholder groups are involved multiple times. A more in-depth analysis is required for specific scenarios. This also applies to the description of the automotive ecosystem, which can be considered in much more detail. Furthermore, the evaluation of the proposed stakeholder set can be extended to close possible gaps and ease the model's application in other studies. Moreover, the decision-making and enforcement of trust-based decisions, including the proposed approach to integrate stakeholders, will be discussed in future work. Despite the limitations, the insights gained can be used to define requirements for a trust management system that can map different use cases in the automotive ecosystem.

Acknowledgment

Parts of this work were created in the research project TRADE, funded by the German Federal Ministry of Research and Education under grant 16KIS1409.

References

[1] M. Michl and H.-J. Hof, "Towards a Stakeholder-Centric Trust Management Approach for the Automotive Ecosystem," in *SECURWARE 2024*, vol. 18, Nice, France, Nov. 2024, ISBN: 978-1-68558-206-7.

[2] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of Vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, Oct. 2014, ISSN: 1673-5447. DOI: 10.1109/CC.2014.6969789.

[3] A. Rehman *et al.*, "CTMF: Context-Aware Trust Management Framework for Internet of Vehicles," *IEEE Access*, vol. 10, pp. 73 685–73 701, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3189349.

[4] Y. Kuang, H. Xu, R. Jiang, and Z. Liu, "GTMS: A Gated Linear Unit Based Trust Management System for Internet of Vehicles Using Blockchain Technology," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Wuhan, China: IEEE, Dec. 2022, pp. 28–35, ISBN: 978-1-6654-9425-0. DOI: 10.1109/TrustCom56396.2022.00015.

[5] *ISO 9000: Quality management systems – Fundamentals and vocabulary*, Geneva, Switzerland, Dec. 2005.

[6] *DIN-69901: Project management Project management systems Part 5: Concepts*, Jan. 2009.

[7] N. Luhmann, M. King, and C. Morgner, *Trust and Power*. Malden, MA: Polity, 2017, ISBN: 978-1-5095-1945-3.

[8] E. Pöll, "Engineering the trust machine. Aligning the concept of trust in the context of blockchain applications," *Ethics and Information Technology*, vol. 26, no. 2, p. 37, Jun. 2024, ISSN: 1388-1957, 1572-8439. DOI: 10.1007/s10676-024-09774-6.

[9] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, Emerging Issues in Collaborative Commerce, vol. 43, no. 2, pp. 618–644, Mar. 2007, ISSN: 0167-9236. DOI: 10.1016/j.dss.2005.05.019.

[10] *UN Regulation No. 155 - Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system*, UN Regulation, Mar. 2021.

[11] T. Kosch, Ed., *Automotive Internetworking* (Intelligent Transportation Systems). Hoboken, N.J: Wiley, 2012, ISBN: 978-0-470-74979-1.

[12] K. Marner, S. Wagner, and G. Ruhe, "Stakeholder identification for a structured release planning approach in the automotive domain," *Requirements Engineering*, vol. 27, no. 2, pp. 211–230, Jun. 2022, ISSN: 0947-3602, 1432-010X. DOI: 10.1007/s00766-021-00369-x.

[13] K. Gomez Buquerin and H.-J. Hof, "Identification of Automotive Digital Forensics Stakeholders," SECUREWARE 2021, p. 7, 2021.

[14] J. M. Bryson, "What to do when Stakeholders matter: Stakeholder Identification and Analysis Techniques," *Public Management Review*, vol. 6, no. 1, pp. 21–53, Mar. 2004, ISSN: 1471-9037, 1471-9045. DOI: 10.1080/14719030410001675722.

[15] C. S. King, K. M. Feltey, and B. O. Susel, "The Question of Participation: Toward Authentic Public Participation in Public Administration," *Public Administration Review*, Public Administration Review, vol. 58, no. 4, pp. 317–326, Jun. 1998.

[16] H. Mansor, "Security and Privacy Aspects of Automotive Systems," Ph.D. dissertation, Royal Holloway, University of London, London, Jul. 2017.

[17] E. Knauss and D. Damian, "Towards Enabling Cross-Organizational Modeling in Automotive Ecosystems," in *MD^2P^2 2014 – Model-Driven Development Processes and Practices*, Valencia, Spain, 2014-09-28/2014-10-03.

[18] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01, Berlin, Heidelberg: Springer-Verlag, 2002, pp. 251–260, ISBN: 3-540-44179-4.

[19] S. P. Marsh, "Formalising Trust as a Computational Concept," Ph.D. dissertation, University of Stirling, Stirling, Scotland, UK, Apr. 1994.

[20] S. M. Habib, "Trust establishment mechanisms for distributed service environments," Ph.D. dissertation, Technische Universität, Darmstadt, 2014.

[21] T. R. Hawkins, B. Singh, G. Majeau-Bettez, and A. H. Strømman, "Comparative Environmental Life Cycle Assessment of Conventional and Electric Vehicles," *Journal of Industrial Ecology*, vol. 17, no. 1, pp. 53–64, Feb. 2013, ISSN: 1088-1980, 1530-9290. DOI: 10.1111/j.1530-9290.2012.00532.x.

[22] J. Kuschel, "The Vehicle Ecosystem," in *Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion*, G. León, A. M. Bernardos, J. R. Casar, K. Kautz, and J. I. De Gross, Eds., vol. 287, Boston, MA: Springer US, 2008, pp. 309–322, ISBN: 978-0-387-87502-6 978-0-387-87503-3. DOI: 10.1007/978-0-387-87503-3_18.

[23] ISO, *ISO 15031-3: Road vehicles - Communication between vehicle and external equipment for emission-related diagnostics - Part 3: Diagnostic connector and related electrical circuits: Specification and use*, Geneva, Switzerland, Feb. 2023.

[24] H.R.1449 — 112th Congress (2011-2012), *Motor Vehicle Owners Right to Repair Act of 2011*, https://www.congress.gov/bill/112th-congress/house-bill/1449, Legislation, Apr. 2011.

[25] J. Blümke, K. Mayer, and H.-J. Hof, "An Analysis of Security Concerns in Transitioning Battery Management Systems from First to Second Life," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, Vienna Austria: ACM, Jul. 2024, pp. 1–11, ISBN: 979-8-4007-1718-5. DOI: 10.1145/3664476.3671010.

[26] Volkswagen AG, *Notes about theft protection and application of a FAZIT/ GeKo authorization < Volkswagen AG erWin Online*, https://erwin.volkswagen.de/erwin/showOnlineServices.do.

[27] A. Bucaioni and P. Pelliccione, "Technical Architectures for Automotive Systems," in *2020 IEEE International Conference on Software Architecture (ICSA)*, Salvador, Brazil: IEEE, Mar. 2020, pp. 46–57, ISBN: 978-1-7281-4659-1. DOI: 10.1109/ICSA47634.2020.00013.

[28] G. Gut, C. Allmann, M. Schurius, and K. Schmidt, "Reduction of Electronic Control Units in Electric Vehicles Using Multicore Technology," in *Multicore Software Engineering, Performance, and Tools*, D. Hutchison *et al.*, Eds., vol. 7303, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 90–93, ISBN: 978-3-642-31201-4 978-3-642-31202-1. DOI: 10.1007/978-3-642-31202-1_11.

[29] J. Dobaj, G. Macher, D. Ekert, A. Riel, and R. Messnarz, "Towards a security-driven automotive development lifecycle," *Journal of Software: Evolution and Process*, Nov. 2021, ISSN: 2047-7473, 2047-7481. DOI: 10.1002/smr.2407.

[30] *ISO 27145: Road vehicles - Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements - Part 1: General information and use case definition*, Standard, Geneva, Switzerland, Aug. 2012.

[31] *ISO 26021-1:2022: Road vehicles – End-of-life activation of in-vehicle pyrotechnic devices – Part 1: Application and communication interface*, Standard, Geneva, Switzerland, Mar. 2022.

[32] M. S. Ferdous, G. Norman, A. Jøsang, and R. Poet, "Mathematical Modelling of Trust Issues in Federated Identity Management," in *Trust Management IX*, C. Damsgaard Jensen, S. Marsh, T. Dimitrakos, and Y. Murayama, Eds., vol. 454, Cham: Springer International Publishing, 2015, pp. 13–29, ISBN: 978-3-319-18490-6 978-3-319-18491-3. DOI: 10.1007/978-3-319-18491-3_2.

[33] J. Urbano, A. P. Rocha, and E. Oliveira, "The Impact of Benevolence in Computational Trust," in *Agreement Technologies*, D. Hutchison *et al.*, Eds., vol. 8068, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 210–224, ISBN: 978-3-642-39859-9 978-3-642-39860-5. DOI: 10.1007/978-3-642-39860-5_16.

[34] D. Zhang, F. R. Yu, Z. Wei, and A. Boukerche, "Software-defined Vehicular Ad Hoc Networks with Trust Management," in *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, ser. DIVANet '16, New York, NY, USA: Association

for Computing Machinery, Nov. 2016, pp. 41–49, ISBN: 978-1-4503-4506-4. DOI: 10.1145/2989275.2989285.

[35] J. Zhao, F. Huang, L. Liao, and Q. Zhang, "Blockchain-Based Trust Management Model for Vehicular Ad Hoc Networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023, ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2023.3318597.

[36] W. Yong-hao, "A Trust Management Model for Internet of Vehicles," in *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, ser. ICCSP 2020, New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 136–140, ISBN: 978-1-4503-7744-7. DOI: 10.1145/3377644.3377664.

[37] D. Zhang, F. R. Yu, and R. Yang, "A Machine Learning Approach for Software-Defined Vehicular Ad Hoc Networks with Trust Management," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6, ISBN: 978-1-5386-4727-1. DOI: 10.1109/GLOCOM.2018.8647426.

[38] D. Zhang, F. R. Yu, R. Yang, and L. Zhu, "Software-Defined Vehicular Networks With Trust Management: A Deep Reinforcement Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1400–1414, Feb. 2022, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.3025684.

[39] D. Wang, Y. Yi, S. Yan, N. Wan, and J. Zhao, "A node trust evaluation method of vehicle-road-cloud collaborative system based on federated learning," *Ad Hoc Networks*, vol. 138, p. 103 013, Jan. 2023, ISSN: 15708705. DOI: 10.1016/j.adhoc.2022.103013.

[40] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, "A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network," *IEEE Access*, vol. 9, pp. 31 309–31 321, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3060046.

[41] W. Li and H. Song, "ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960–969, Apr. 2016, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2015.2494017.

[42] Q. Alriyami, A. Adnane, and A. K. Smith, "Evaluation criterias for trust management in vehicular ad-hoc networks (VANETs)," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, Vienna, Austria: IEEE, Nov. 2014, pp. 118–123, ISBN: 978-1-4799-6729-2. DOI: 10.1109/ICCVE.2014.7297525.

[43] S. Tangade and S. S. Manvi, "CBTM: Cryptography Based Trust Management Scheme for Secure Vehicular Communications," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore: IEEE, Nov. 2018, pp. 325–330, ISBN: 978-1-5386-9582-1. DOI: 10.1109/ICARCV.2018.8581173.

[44] M. Michl, H.-J. Hof, and S. Katzenbeisser, "Classification, Impact, and Mitigation Strategies of Attacks in Automotive Trust Management Systems," in *Proceedings of the 2024 on Cyber Security in CarS Workshop*, ser. CSCS '24, Salt Lake City, UT, USA: ACM, Nov. 2024, pp. 61–75, ISBN: 979-8-4007-1232-6. DOI: 10.1145/3689936.3694691.

[45] A. Jøsang, *Subjective Logic* (Artificial Intelligence: Foundations, Theory, and Algorithms). Cham: Springer International Publishing, 2016, ISBN: 978-3-319-42335-7 978-3-319-42337-1. DOI: 10.1007/978-3-319-42337-1.

[46] L. Viljanen, "Towards an Ontology of Trust," in *Trust, Privacy, and Security in Digital Business*, D. Hutchison *et al.*, Eds., vol. 3592, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 175–184, ISBN: 978-3-540-28224-2 978-3-540-31796-8. DOI: 10.1007/11537878_18.

[47] I.-L. Yen, A. Tiwari, and F. Bastani, "Access Control in Dynamic IoT Scenarios," in *2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS)*, Mexico City, Mexico: IEEE, Mar. 2023, pp. 1–8, ISBN: 978-1-6654-6451-2. DOI: 10.1109/ISADS56919.2023.10092159.

[48] P. Bhattacharya, A. Shukla, S. Tanwar, N. Kumar, and R. Sharma, "6Blocks: 6G-enabled trust management scheme for decentralized autonomous vehicles," *Computer Communications*, vol. 191, pp. 53–68, Jul. 2022, ISSN: 01403664. DOI: 10.1016/j.comcom.2022.04.024.

[49] F. Kandah, B. Huber, A. Skjellum, and A. Altarawneh, "A Blockchain-based Trust Management Approach for Connected Autonomous Vehicles in Smart Cities," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA: IEEE, Jan. 2019, pp. 0544–0549, ISBN: 978-1-7281-0554-3. DOI: 10.1109/CCWC.2019.8666505.

[50] H. Zhang, J. Liu, H. Zhao, P. Wang, and N. Kato, "Blockchain-Based Trust Management for Internet of Vehicles," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1397–1409, Jul. 2021, ISSN: 2168-6750, 2376-4562. DOI: 10.1109/TETC.2020.3033532.

[51] Z. Yang, R. Wang, D. Wu, B. Yang, and P. Zhang, "Blockchain-Enabled Trust Management Model for the Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 044–12 054, Jul. 2023, ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2021.3124073.

[52] X. Wang *et al.*, "Blockchain-enhanced trust management for mobile edge computing-enabled intelligent vehicular collaboration in the 6G era," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 7, e4791, 2023, ISSN: 2161-3915. DOI: 10.1002/ett.4791.

[53] D. Wang, L. Zhang, C. Huang, and X. Shen, "A Privacy-Preserving Trust Management System based on Blockchain for Vehicular Networks," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China: IEEE, Mar. 2021, pp. 1–6, ISBN: 978-1-7281-9505-6. DOI: 10.1109/WCNC49053.2021.9417492.

[54] P. K. Singh *et al.*, "Blockchain-Based Adaptive Trust Management in Internet of Vehicles Using Smart Contract," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3616–3630, Jun. 2021, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.3004041.

[55] M. Ebrahimi, M. S. Haghighi, A. Jolfaei, N. Shamaeian, and M. H. Tadayon, "A Secure and Decentralized Trust Management Scheme for Smart Health Systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1961–1968, May 2022, ISSN: 2168-2208. DOI: 10.1109/JBHI.2021.3107339.

[56] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An Integrative Model of Organizational Trust," *The Academy of Management Review*, vol. 20, no. 3, p. 709, Jul. 1995, ISSN: 03637425. DOI: 10.2307/258792. JSTOR: 258792.

[57] Y. Yao, W. Chen, X. Chen, J. Ding, and S. Pan, "A Blockchain-based Privacy Preserving Scheme for Vehicular Trust Management Systems," in *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*, Zhenjiang, China: IEEE, Nov. 2020, pp. 1–5, ISBN: 978-1-7281-9301-4. DOI: 10.1109/ITIA50152.2020.9312254.

[58] V. Venkatraman, S. Pal, Z. Jadidi, and A. Jolfaei, "A Conceptual Trust Management Framework under Uncertainty for Smart Vehicular Networks," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, New York, NY, USA: IEEE, May 2022, pp. 1–7, ISBN: 978-1-6654-0926-1. DOI: 10.1109/INFOCOMWKSHPS54753.2022.9797996.

[59] S. A. Siddiqui, A. Mahmood, W. E. Zhang, and Q. Z. Sheng, "Poster: A Machine Learning based Hybrid Trust Management Heuristic for Vehicular Ad hoc Networks," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19, New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 1–3, ISBN: 978-1-4503-6169-9. DOI: 10.1145/3300061.3343404.

[60] S. Abbasi, N. Khaledian, and A. M. Rahmani, "Trust management in the internet of vehicles: A systematic literature review of blockchain integration," *International Journal of Information Security*, Jul. 2024, ISSN: 1615-5262, 1615-5270. DOI: 10.1007/s10207-024-00878-0.

[61] S. Marsh and M. R. Dibben, "Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side," in *Trust Management*, D. Hutchison *et al.*, Eds., vol. 3477, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 17–33, ISBN: 978-3-540-26042-4 978-3-540-32040-1. DOI: 10.1007/11429760_2.

[62] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A Survey of Current Solutions and Future Research Opportunities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2553–2571, May 2021, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.2973715.

Understanding Human Aspects in Phishing Detection: The Role of Demographics, Eye Movements and User Experience in Security Software

Fabian Engl, Meret Kristen, Jürgen Mottok
Software Engineering Laboratory for Safe and Secure Systems
OTH Regensburg
Regensburg, Germany
email: {fabian.engl | meret.kristen | juergen.mottok}@oth-regensburg.de

Abstract—**This paper builds upon a previous study that analyzed phishing detection using eye-tracking data from 103 participants tasked with classifying 18 emails. Additionally, a phishing awareness system (PAS) was introduced, highlighting relevant information for half of the participants. While the original analysis found no significant improvements in detection effectiveness, the eye-tracking data did reveal that participants using the supportive software spent less time examining key phishing indicators. Expanding on these findings, this work incorporates further questionnaire data and a more advanced Area of Interest (AoI) analysis to provide deeper insights. The results indicate that demographic factors such as age, gender, and education have no significant impact on phishing detection. However, industry sectors and weekly screen time did influence performance, particularly in terms of the time required for classification. A qualitative eye movement analysis further revealed distinct AoI hit patterns between participants who correctly classified all emails and those who misclassified more. Additionally, gaze behavior varied based on participants' usability and user experience ratings of the supportive software, highlighting a potential impact for specific user groups, when it comes to phishing detection efficiency.**

Keywords-Phishing; Security Awareness; Eye-Tracking; IT-Security; Usability and UX.

I. Introduction

This paper builds on previous research that investigated phishing detection using eye-tracking analysis [1]. Despite widespread awareness of phishing and its associated risks, these attacks remain a persistent daily threat. The German Federal Office for Information Security (BSI) highlighted in its 2024 IT-Security report that many individuals continue to underestimate the severity of phishing, often realizing the long-term consequences only when it is too late [2]. Phishing attacks typically disguise themselves as legitimate emails or messages to deceive individuals into revealing sensitive information, such as login credentials, financial details, or confidential data. As the volume of emails continues to rise and phishing tactics grow more sophisticated, individuals are becoming increasingly vulnerable. Historically, phishing emails primarily impersonated financial institutions, requesting monetary transfers; however, in recent years, they have shifted towards everyday communications, making these attacks both more pervasive and harder to detect [2].

Given the increasing prevalence and complexity of phishing attacks, equipping individuals with the skills to recognize these threats is more crucial than ever, both in personal and corporate settings. Traditional in-company security awareness trainings - often based on theoretical knowledge - have sparked a debate regarding their effectiveness in preventing such attacks. Some argue that humans are the weakest link in cybersecurity [3] [4] and that dedicated training can significantly improve individuals' ability to recognize phishing threats [5]. However, studies such as the one conducted by Lain et al. suggest that such training has no significant impact on a person's ability to detect phishing emails [6].

Phishing research is typically conducted using questionnaire-based studies [7] [8] [9]. However, questionnaires may not fully capture an individual's decision-making process, often providing an incomplete or distorted picture of the cognitive mechanisms involved in phishing detection. Eye-tracking technology, on the other hand, offers a more precise representation of decision-making processes [10]. This journal paper first conducts a statistical analysis to determine whether and which demographic factors influence phishing email detection and then explores how eye-tracking data can provide deeper insights into decision-making patterns that remain hidden in traditional phishing studies.

Section II reviews recent literature published since the initial study. Sections III to V define and address eight research questions, beginning with statistical tests on questionnaire data and ending with a qualitative analysis of eye movement patterns. These sections also provide a detailed overview of the participants, the technical setup, and the study design. Section VI further investigates the usability and user experience of a software add-on designed to highlight phishing-relevant information. Finally, Sections VII and VIII discuss the study's limitations, summarize key findings on the effectiveness of phishing training, and outline directions for future research.

II. Recent Developments in Literature

The literature review in [1] revealed that previous eye-tracking research studying phishing either relies on relatively small sample sizes or focuses on adaptive mechanisms designed to enhance users' ability to recognize phishing attempts. However, there remained a significant gap in understanding how users engage with available tools and warnings, as well as which phishing indicators they tend to overlook when falling victim to such attacks. To address these gaps, the study in [1] was developed. Since the literature review for that article had to be carried out before the start of the study and the implementation of the study itself took several months, the literature review did not include articles published after February 2024. In the last 12 months, several new papers have been published studying phishing emails with eye-tracking technology. For this reason, a discussion of these new papers and how their results compare to those found in [1] is included here.

In [11], published in November 2024, the authors examined how individuals visually inspect phishing and legitimate emails. A key hypothesis was that participants would pay more attention to the sender's identification in phishing emails compared to legitimate ones, which was confirmed by the results. However, contrary to expectations, participants were not more likely to misidentify phishing emails; instead, they tended to misclassify legitimate emails more frequently.

The study involved 68 participants, predominantly women (77.9 %), with an average age of 23.91 years (ranging from 18 to 48). More than half of the participants (51.5%) had completed their 12th grade education.

Data was collected using Tobii Pro Fusion Eye-Trackers. A total set of 28 emails was examined, consisting of 13 phishing emails, 13 legitimate emails, and two control emails, each with predefined areas of interest (AOI), including the header of the email, the subject line, the sender's information, the body of the email, the salutation, the links, misspellings, financial indicators, threats, and urgency signals. Each participant was shown 15 randomly selected emails from two groups.

The study analyzed the total number of fixations and the fixation duration (in milliseconds) within each AOI. Statistical comparisons were conducted using Mann-Whitney U-tests. The general setup of the study is very similar to that presented in [1] and the results show that participants spent more time looking at the sender information in phishing emails. Since [1] only tested whether AOI hits on the sender information differ between the group with PAS and the group without, this result will be tested against the original data set from [1] in research question 6, to see whether the collected data is consistent.

Furthermore, the authors of [11] suggest that future research should differentiate between specific phishing characteristics, such as financial content, threats, spelling errors, and urgency cues. This was already addressed in [1]. Another suggestion was to examine the visual inspection patterns of phishing experts and previous victims, which is addressed in research question 7 below.

In [12], a literature review with the search string [phishing AND EEG], [phishing AND "eye-tracking" OR eye-tracking], [phishing AND BCI] in Elsevier ScienceDirect, IEEE Xplore, Research-Gate, Springer, and the ACM Digital Library is presented. Similarly to the literature review in [1], the found papers are compared with regards to participants, types of investigated phishing attacks and results. The examined literature suggests that user personality traits, such as attention control, may have a direct impact on their susceptibility to phishing. The paper describes the controversy surrounding the impact of demographic factors on phishing susceptibility and the limited scope of current studies. It suggests further research to explore other phishing types, assess resilience to multiple attacks, and incorporate advanced AI methods and real-world conditions.

[13] presents an eye-tracking study with $n = 40$ participants and 18 emails. This study explores the effects of visual risk indicators on phishing detection behavior using an eye-tracking experiment, and provides implications for how organizations can effectively integrate and calibrate such indicators to mitigate phishing attempts. It studied how displaying a phishing risk indicator affects visual attention, trust, and time taken to come to a decision. It was discovered that the visual risk indicator has a significant impact on trust, which subsequently influences the behavior of the participants' email responses.

[14] investigates how workload influences an individual's likelihood of falling for phishing attacks, utilizing eye-tracking technology to track how participants read and engage with personalized phishing emails. By combining both quantitative and qualitative approaches, it analyses participants' focus on two key phishing cues: the sender's email address and hyperlink URLs. Results reveal that paying attention to the email sender helps reduce phishing vulnerability, but no link between noticing the actual URL and improved phishing detection was found. In contrast, focusing on the text hiding the links tends to increase phishing risk. These suggestions are addressed in Research Questions 6 and 7.

Lastly, [15] presents an eye-tracking study with 42 participants that focuses on spear phishing. The results show that the participants have shorter total fixation durations on spear phishing emails than on legitimate emails. Phishing training was not shown to have a main effect on eye movement behaviors. Participants tended to focus their attention on the email body, followed by the subject line and sender information, but neglected the sent time.

III. Research Objectives

Several new questions arise from the review of literature published in the last year. Together with further analysis of the data set presented in [1], this gives rise to the following set of research questions:

RQ1 How do demographic differences such as age, gender, and education affect phishing recognition?

RQ2 Are there differences between employees of different industries in regards to effectiveness and efficiency of phishing recognition?

RQ3 Does the ability to recognize phishing emails differ among employees based on their weekly screen time?

RQ4 Does knowing the sender company affect the recognition of phishing emails?

RQ5 Are IT security experts better at detecting phishing emails than laypersons?

RQ6 Do users focus more on the sender when examining phishing emails compared to legitimate emails?

RQ7 How do gaze patterns differ between individuals who correctly identify a high versus a low number of phishing emails?

RQ8 How do gaze patterns differ between individuals who rate the usability and user experience (UX) of the PAS as low compared to those who rate it as high?

Based on these research questions, the following hypothesis were developed:

H1 Age, gender and education level have little to no effect on phishing recognition rates.

H2 Employees of different industries express different levels of phishing recognition efficiency and effectiveness, proportional to their use of email in daily life.

H3 Participants with increased weekly screen-based work hours show higher rates of phishing recognition.

H4 Knowing the sender company will effect phishing recognition rates.

H5 IT security experts are expected to perform better at the phishing recognition task than laypersons.

H6 In line with the results found in [11] it is expected that users focus more on the sender when examining phishing emails.

H7 Individuals who correctly identified more phishing emails used the PAS more compared to the individuals that misclassified phishing emails.

H8 Individuals who rated the usability and especially the UX as high spend more time interacting with the PAS compared to those who rated both low.

IV. Study design

As described in [1], this eye-tracking study was conducted at the University of Applied Sciences in Regensburg (OTH Regensburg) and as part of a service offered by the European Digital Innovation Hub "Digital Innovation Ostbayern" (DInO). DInO offers free consulting services to small and medium-sized enterprises (SMEs) and the public sector (PSEs), especially in Eastern Bavaria. Since IT security training is mandatory for many German companies, this study was designed as an interactive extension to traditional theoretical training.

Beyond corporate use, the study also aimed to help individuals develop a better awareness of phishing emails and improve their ability to detect them. To ensure relevance and familiarity, the phishing emails used in the study were sourced primarily from real interactions. All were genuine phishing attempts, collected from colleagues and relatives. In some cases, minor modifications—such as translations or company name changes—were made to prevent reputational harm to smaller businesses.

Notably, while all participants were exposed to the same phishing emails, half of the group had access to an additional tool called the "Phishing Awareness System" (PAS), which highlighted specific information. This system will be introduced in Section IV-D.

A. Participants

A total of 120 participants took part in the study. However, since the study was also offered as a complementary phishing training, eleven participants opted to participate only in the training without being included in the study. Their recordings were deleted immediately after the session and were not included in the final dataset. Additionally, six participants had to be excluded due to severe visual impairments, as they failed to meet the calibration threshold of $0.75°$, primarily due to extreme diopter levels or incompatible glasses and contact lenses. Before beginning, all participants filled out a consent and demographic form.

In the final dataset, 103 participants remained, of whom 36.89% were female ($n = 38$) and 63.11% were male ($n = 65$), with an average age of 32.81 years. Among them, 52 had access to the Phishing Awareness System (PAS), while 51 relied solely on the email content for their decisions. 91.26% ($n = 94$) reported knowing what phishing emails look like, and 60.19% ($n = 62$) had attended at least one phishing training session in the past. Additionally, 38.83% ($n = 40$) received phishing emails daily, 28.85% ($n = 30$) weekly, 8.65% ($n = 9$) monthly, and 23.30% ($n = 24$) rarely or never.

A closer look at participants' educational backgrounds revealed an atypical distribution. Based on the German education system, four educational attainment groups were identified:

- 57 participants had a general or subject-specific university entrance qualification (German: Abitur/Allgemeine oder fachgebundene Hochschulreife).
- 10 participants had a technical college entrance qualification (German: Fachhochschulreife).
- 23 participants had a general secondary education diploma (German: Realschulabschluss/Mittlere Reife).
- 11 participants completed basic secondary schooling (German: Hauptschulabschluss).
- 2 participants reported other forms of schooling.

Participants were also asked about their professional qualifications and degrees. Since this was a multiple-choice question, the number of responses exceeds the total number of participants:

- 52 participants had completed an apprenticeship or professional training (German: Berufsausbildung)
- 31 participants had a bachelor's degree
- 32 participants had a master's degree
- 2 participants had a PhD

This distribution is particularly noteworthy since eye-tracking studies are often academically biased, predominantly consisting of students and university employees as well as teachers [16] [17] [18]. The fact that over half of the participants had completed an apprenticeship or professional training highlights not only the scale but also the diversity of this study. A further demographic analysis showed that 85.44% ($n = 88$) of participants were employed, while 14.56% ($n = 15$) were self-employed. Among all, 66.99% ($n = 69$) worked full-time, 18.45% ($n = 19$) worked part-time, and the remaining participants reported other forms of employment, including apprenticeships or mini-jobs. The average weekly working hours were 35.90 ($min = 8$, $max = 55$, $std = 9.05$), with participants spending an average of 27.16 hours in front of a computer screen ($min = 0$, $max = 55$, $std = 12.21$). The average work experience was 14.27 years ($min = 0$, $max = 45$, $std = 12.99$).

B. Technical Setup

Up to nine Tobii Pro Fusion eye-trackers were used to record the data, with a recording frequency of $250Hz$. Participants were positioned approximately 65 cm from a 21-inch monitor set to a resolution of 1920×1080 pixels, running at $60Hz$. These specifications align with the quality analysis and recommendations in [19]. Following these guidelines, participants were instructed to remain still during the recording and avoid head movements.

The study was conducted using Tobii Pro Lab software (Version 1.232.52758) and employed the Tobii I-VT fixation filter. The Tobii Pro Fusion devices operated on firmware version 1.19.22.

C. Stimuli

To enhance the study design, emails were categorized into three groups, each representing a common type of phishing attack. A total of 18 emails were included in the study, evenly distributed as follows:

- **Control emails**
  Legitimate, harmless emails, such as notifications from energy providers or PayPal.
- **"Badly made" phishing emails**
  Contained multiple red flags, such as cryptic sender addresses or severe misspellings, making them easier to identify.
- **"Well-crafted" phishing emails**
  More sophisticated attempts with only minor misspellings, subtle anomalies, or unusual attachments, for example Word documents containing macros.

These distinguishing features, which allow for the classification of phishing emails, will be referred to as phishing markers throughout the study. Each email category was further divided into three common phishing attack techniques:

- Two emails with attachments containing relevant documents, primarily invoices or monthly billing statements.
- Two emails urging the recipient to click a link to complete an action, such as reactivating an account.
- Two emails requesting money, either through a direct demand or an implicit threat of financial consequences.

To create a realistic testing environment, emails were displayed within a typical Outlook email interface. Outlook was chosen because it is among the most widely used email clients [20] and often used in corporate settings.

D. Phishing Awareness System

As mentioned earlier, this eye-tracking study followed an in-between-subject design, with one group having access to a prototype of the Phishing Awareness System (PAS). This

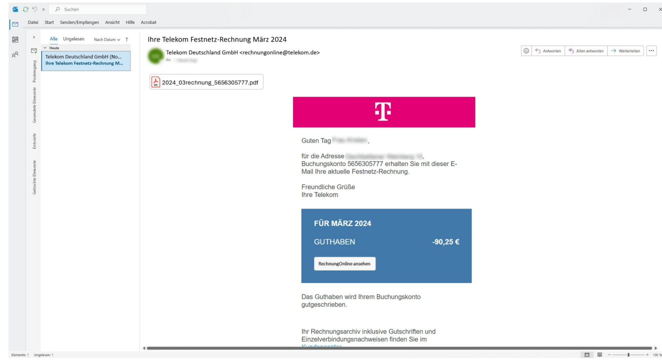


Figure 1. Email from the control group containing an attachment without the PAS.

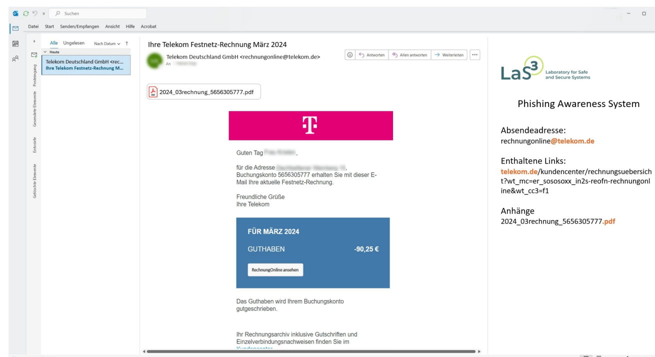


Figure 2. Email from the control group containing an attachment with the PAS at the right side of the screen.

system aggregated and displayed key information to assist in identifying phishing attempts. It highlighted critical elements such as the sender domain, URLs within links, and attachment types, helping to expose spelling errors and other suspicious indicators. An illustration comparing the same email with and without the PAS system is shown in Figures 1 and 2.

Participants in the PAS group were introduced to the tool and its functionality during the study briefing. However, in order to prevent potential bias, they were not required to use it.

A secondary objective of this study was to evaluate whether the PAS improved participants' accuracy and efficiency in detecting phishing emails, as well as assessing their perception of its usefulness. Section VI will provide a detailed analysis of usability and user experience related to the PAS.

E. Areas of Interest

To analyse participant gaze patterns more effectively, Areas of Interest (AoIs) were predefined. These AoIs represent specific screen regions crucial for determining whether an email is phishing or legitimate. They were drawn over key phishing markers in each email, allowing for the aggregation of eye movements within these targeted areas [21]. By using predefined AoIs, the study systematically examined where

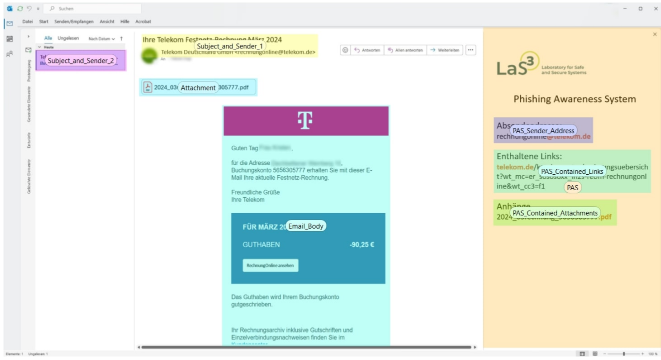


Figure 3. Highlighted AoIs for the email from the control group containing an attachment with the PAS at the right side of the screen.

participants focused their attention and how gaze behavior differed between groups.

Figure 3 provides an overview of these AoIs. In this study, four distinct types of AoIs were defined:

- **Sender Address and Email Subject**
  This information appears twice within the Outlook environment — once at the top of the email and again in the preview pane on the left-hand side. It includes the senders email address, its domain, and the email subject.
- **Email Body**
  This AoI covers the main content of the email, including all text, embedded links, and any other relevant details.
- **Attachment**
  Attachments are typically displayed between the sender information and the email body. This AoI captures the attachment name, file extension, and its icon, all of which provide visual cues about the file type.
- **PAS**
  This AoI is exclusive to the PAS group and consists of: One large area covering the entire PAS interface and three smaller AoIs highlighting the sender domain, included URLs, and attachment details within it.

F. Study Environment and Methodological Challenges

Since this study was also integrated into existing IT security training programs for SMEs and PSEs, it required a different approach compared to traditional eye-tracking studies conducted in laboratory settings. The challenges between these two environments differ significantly, with mobile studies being inherently more complex, particularly when participants have no prior experience with eye-tracking technology [22].

One of the primary concerns in mobile eye-tracking studies is data quality, which is influenced by two key factors: External distractions where Participants may be interrupted by background noise, other participants, or changes in the study environment and technical as well as environmental factors such as poor lighting conditions, calibration problems, and recording errors [19], [22], [23].

To ensure reliable data collection, the eye-tracking system was calibrated to each participant before the study began. Due

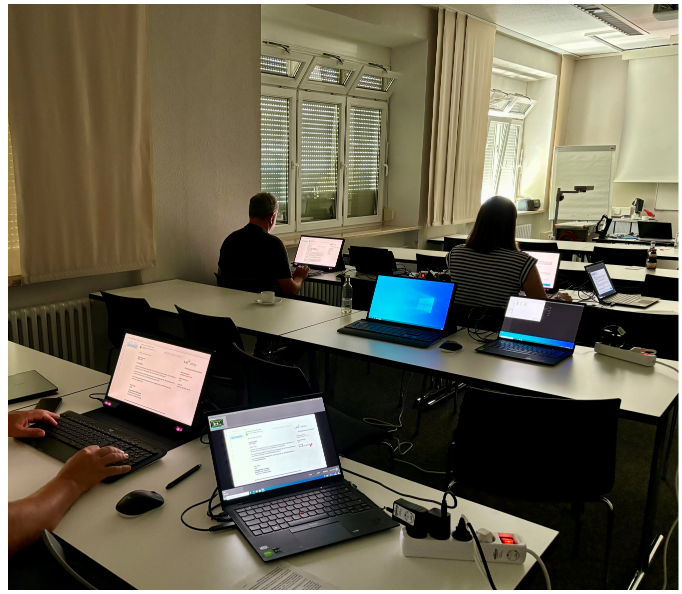


Figure 4. Exemplary study setup for conducting eye-tracking studies in a workshop format.

to the study's relatively short duration (average of $6:40$ minutes), re-calibrations were not performed between stimuli. However, a strict calibration and validation threshold of $0.75°$ was set, and any participant failing to meet this standard was excluded from the study.

To minimize distractions and external influences, several measures were implemented: Firstly, the laptop screen was turned away from participants to prevent distractions. Furthermore, participants were seated directly behind each other to obstruct the view of other screens. Secondly, direct and overhead lighting was turned off and blinds were closed whenever possible to reduce glare. Figure 4 illustrates the typical setup used during workshops.

Beyond technical and environmental factors, participant behavior also played a significant role in data quality. Despite clear instructions to ask questions only during the introduction, some participants raised concerns mid-study, often triggering a chain reaction where others looked away from their screens to listen. In rare instances, discussions emerged among participants, particularly when encountering unusual or suspicious emails. When this occurred, the conductors intervened as discreetly and quickly as possible to minimize disruptions.

For future studies, introducing dedicated breaks between stimuli for questions and short rest periods could be beneficial and combat such behavior. This would allow participants to clarify doubts without disrupting the study flow and help prevent eye strain—an issue raised by participants who needed more time to process all emails.

Despite these challenges, the study demonstrates that parallelism-by-design can enable efficient eye-tracking studies in workshop settings with multiple participants at a time. This was achieved by relying on questionnaires for triangulation, allowing study conductors to oversee multiple sessions si-

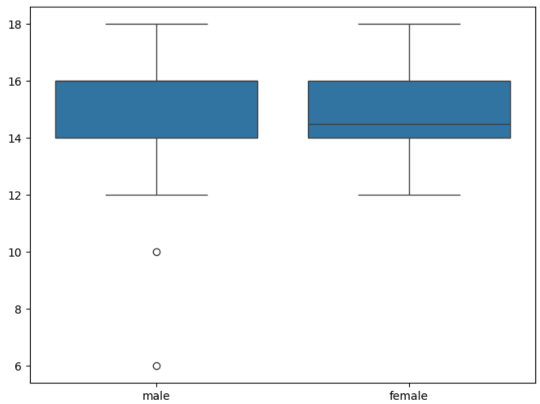


Figure 5. Total number of correctly identified emails for male and female participants.

multaneously. This would not be possible when using think-aloud-protocols or requiring input and validation through the researchers during the study. Self-paced digital instructions further helped the participants progress at their own speed and take out additional stress - which could potentially even introduce bias. All introductions were integrated into the Tobii Pro Lab project, ensuring that everybody received the same information. This had the additional benefit that gaze recordings could be reviewed post-study to verify whether participants actually read the provided instructions.

By implementing these strategies, the study balanced data collection challenges with the practical constraints of real-world IT security training environments, enabling researchers to monitor a higher number of participants while maintaining data integrity.

V. Results

Since no significant differences in phishing detection were found between the group with the Phishing Awareness System (PAS) and the group without it [1], this section further analyses possible correlations by testing demographic differences across the entire dataset, without differentiating between participants with or without PAS.

To test RQ1, a Shapiro-Wilk test [24] revealed that the dependent variable "correctly identified emails" was not normally distributed within the "male" and "female" groups. Therefore, a Mann-Whitney U-test was used, which found no significant differences in the number of correctly identified emails between the two groups at $\alpha = 0.05$ ($z = 1473.00$, $p = 0.099$, $r = 0.16$). Figure 5 displays the results for male and female participants, respectively. Furthermore, no significant group differences were found when comparing the total time taken to complete the task.

Secondly, the results were compared based on participants' highest level of general education. The assumption of normality was assessed using Shapiro-Wilk tests, which revealed non-normal distributions across all groups. Given the lack of normality, non-parametric statistical tests were used. The Kruskal-Wallis test [25] was applied to assess overall group differences, with Mann-Whitney U tests used for post-hoc comparisons. Due to tied ranks in the dataset, p-values were approximated, and continuity correction was applied. Additionally, a Bonferroni correction [26] was used to adjust p-values for multiple comparisons. The Kruskal-Wallis test showed no statistically significant differences between educational attainment groups with regard to Correctly Identified Emails Total ($\chi^2(3) = 3.72$, $p = 0.293$, $\eta^2 = 0.01$). This suggests that educational attainment had a negligible effect on email identification accuracy. Post-hoc Mann-Whitney U tests with Bonferroni correction confirmed the absence of significant differences between any pair of groups, as shown in Table I.

TABLE I. Pairwise Mann-Whitney U-test results for educational levels

	Abitur	**Fachhoch-schulreife**	**Realschul-abschluss**	**Hauptschul-abschluss**
Abitur		$p = 1.000$	$p = 1.000$	$p = .604$
Fachhoch-schulreife	$p = 1.000$		$p = 1.000$	$p = 1.000$
Realschul-abschluss	$p = 1.000$	$p = 1.000$		$p = 1.000$
Hauptschul-abschluss	$p = .604$	$p = 1.000$	$p = 1.000$	

However, statistically significant differences were observed when analysing the time participants needed to complete the study. The Kruskal-Wallis test revealed values of $\chi^2(3) = 15.10$, $p = 0.002$, $\eta^2 = 0.12$, indicating a significant difference at $\alpha = 0.05$ with a moderate effect size.

Pairwise comparisons using Mann-Whitney U tests showed significant differences at $\alpha = 0.05$ between the 'Abitur' and 'Hauptschulabschluss' groups ($p = 0.015$), as well as between the 'Fachhochschulreife' and 'Hauptschulabschluss' groups ($p = 0.026$). These comparisons were conducted with approximated p-values, continuity correction, and a Bonferroni correction applied by multiplying the p-value by the number of tests performed. The results are shown in Figure 6.

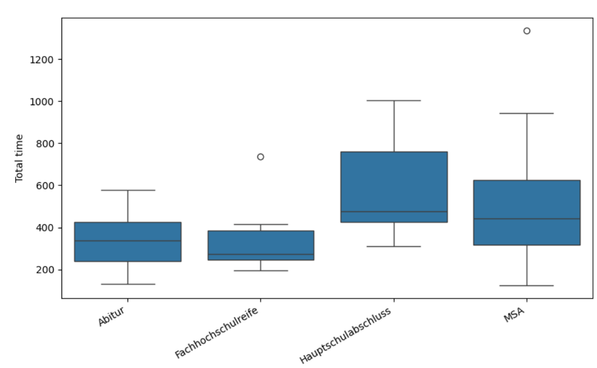


Figure 6. Total time needed to complete the study by highest general education degree

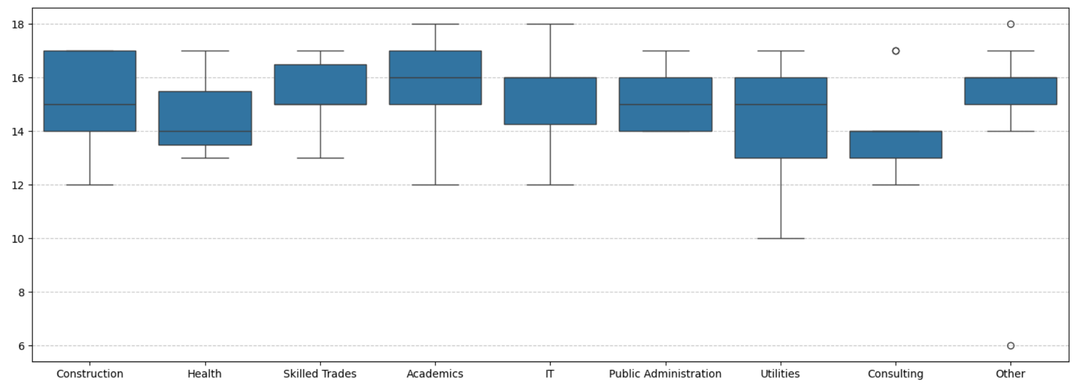


Figure 7. Total number of correctly identified emails by industry sector.

Lastly, the relationship between the number of correctly identified emails and age was examined. The data is described using the median (Mdn), interquartile range (IQR), and sample size (n): For the number of correctly identified emails, $Mdn = 15.00$, $IQR = 2.00$, and $n = 103$. For age, $Mdn = 32.00$, $IQR = 18.50$, and $n = 103$. Both variables are not normally distributed as accessed with Shapiro-Wilk tests: For the number of correctly identified emails the test yields $W = 0.90$, $p < .001$, which is significant at $\alpha = 0.05$, indicating that the sample is not normally distributed. For age, the test yields $W = 0.92$, $p < .001$, which is significant at $\alpha = 0.05$ and again indicates that the sample is not normally distributed. Due to the non-normal distribution of both variables, median (Mdn) and interquartile range (IQR) were used to describe the sample. As a result, a non-parametric test was conducted. Of the two popular non-parametric correlation analyses, Spearman's [27] and Kendall's [28], the latter is considered more conservative (i.e., more likely to not identify significance when it does not exist). Therefore, Kendall's correlation was used in this analysis. Given the presence of ties in the data (i.e., multiple measurements of one variable with the same value), the p-value was approximated. The Kendall's correlation test showed no significant correlation between age and the number of correctly identified emails at $\alpha = 0.05$, with $z = -0.14$, $p = 0.889$, and $r = 0.01$. These findings suggest that there are no statistically significant relationships between email identification accuracy and participants' age, gender, or highest level of general education. Despite some variation in median scores, effect sizes were negligible, and no pairwise comparisons reached statistical significance. The amount of time required to complete the study varied significantly based on participants' highest general education level, but age and gender did not have an effect. These results imply that education plays a significant role in phishing detection, whereas demographic factors such as gender and age do not.

For the second research question, participants were asked to state the industry sector they work in. Any sector that was listed less than three times is listed under "Other", to allow for more accuracy in the statistical tests. To test whether the groups differ in effectiveness and efficiency of phishing detection, it was tested whether the dependent variables "number of correctly identified emails" and "total time spent for the task" were normally distributed. Since this was only the case for 7 out of the 9 groups, non-parametric tests were employed. Here, Kruskal-Wallis test should be used to test for group differences, while Mann-Whitney-U tests with adequate Bonferroni correction may be used as post-hoc tests. For the latter, as there are sample sizes of each two groups are higher than 20, the p-value can be extracted very well from an approximation. Due to unequal sample sizes for both groups, continuity correction is applied. For post-hoc tests in general, the p-values must be adjusted since multiple tests are calculated on the same data. Here, Bonferroni correction is used, which means that p-values are multiplied by the number of pairwise comparisons tests. The Kruskal-Wallis test showed no significant difference in effectiveness at $\alpha = 0.05$ with merely no effect, as shown by $\chi^2(8) = 8.05$, $p = .428$, $\eta^2 = .00$. This can also be seen in Figure 7. However, for the efficiency, a significant difference between the groups was detected. Further pairwise testing with Mann-Whitney U-tests confirmed that the sectors "IT" and and "Construction" differed significantly, shown by a p-value of .015, as well as a significant difference between the groups 'IT' and 'Utilities', shown by a p-value of $p = .003$ after Bonferroni-Correction. None of the other groups showed significant differences in efficiency. The differences between the three relevant industry sectors are shown in Figure 8.

This shows that employees of companies in the IT sector need significantly less time to decide whether an email is legitimate or not than employees in the construction or utilities sector. This effect might be due to familiarity with emails and phishing attempts, advanced knowledge on how possible email scams can look like, and overall confidence in working with a computer. While the industries did not differ in effectiveness of phishing recognition, a difference in efficiency is a good starting point and it should be further analyzed if and how

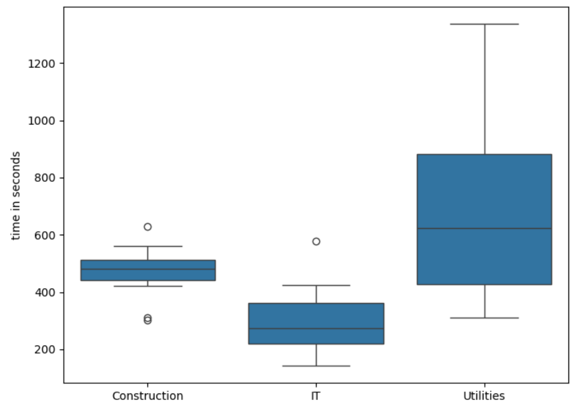


Figure 8. Total time spent on the email sorting task for industry sectors

employees of other industries could be enabled to catch up to the level of expertise shown by employees in the IT sector.

To further evaluate this difference, research question 3 tested whether the weekly screen-based work hours had an effect on the effectiveness and efficiency of phishing recognition. Since employees in the IT-sector naturally spent more of their weekly work hours in front of a computer screen than employees in the Utilities or Construction sector, this is to be expected. And, similarly to RQ2, it was found that while weekly screen-based work hours have only a negligible effect on the total number of correctly identified emails, it has a significant effect on the amount of time needed to complete the task. Since none of the variables are normally distributed, Kendall's correlation analysis was employed and showed that both the total weekly screen-based work hours and the relative weekly screen-based work hours (in relation to total weekly work hours) are significantly correlated to the total amount of time needed to complete the task, as shown by values of $z = -3.13$, $p = .002$, $r = .22$ and $z = -2.71$, $p = .007$, $r = .19$ respectively. This is shown in Figure 10. Furthermore, work experience measured in years had no effect on the efficiency and effectiveness of phishing recognition. To double-check, it was tested whether the number of correctly identified emails and the time spent on the task were correlated, but this was not the case.

Research question 4 answers whether previous knowledge of the sender affects the recognition rates of phishing emails. To test this, the question "From which of the following companies have you already received emails (newsletters, etc.)?" was implemented into the questionnaire for each company presented in the stimuli. Afterwards, the data "correctly classified or not" and "previously known sender or not" were compared for each participant and each email stimulus. A chi-square test of independence (also called a chi-square contingency test) [29] was used to check whether the two binary variables are statistically related. The test revealed a Chi-square statistic of 2.96 and a p-value of 0.085, thus no significant association could be found. Figure 9 shows that participants recognized phishing emails from known senders slightly better than those from unknown senders, but not enough to reach statistical significance.

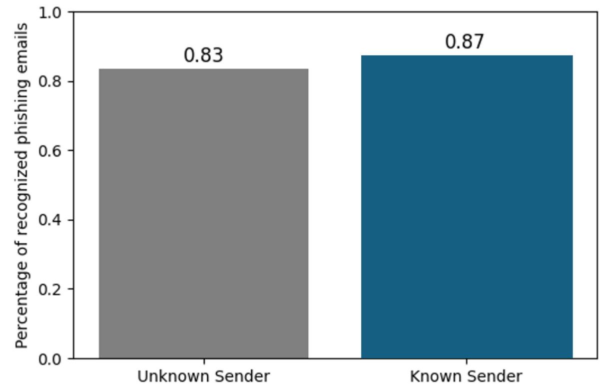


Figure 9. Proportion of correctly identified phishing emails depending on whether the sender was previously known to the participant

To answer research question 5, participants were asked to indicate their agreement or frequency of behavior based on the statements shown in Section II. Responses were recorded using binary values or a 5-point Likert scale whenever suitable, with the values Never, Rarely, Sometimes, Often, Always. The 5-point scale allows for a nuanced assessment of participant behavior rather than a binary yes/no response. Statements are formulated in the first-person to enhance self-reflection and reduce response bias.

TABLE II. QUESTIONNAIRE TO ASSESS PARTICIPANTS IT-SECURITY KNOWLEDGE LEVEL

Statement	Response
I am familiar with the appearance of phishing emails and can identify examples of suspicious characteristics.	yes/no
I use the same password for multiple accounts.	yes/no
I use multi-factor authentication whenever possible.	yes/no
When an update for software or operating systems is available, I install it immediately.	Likert scale (1–5)
I verify the sender's email address before clicking on a link in an email.	Likert scale (1–5)
I check the URL before clicking on a link in an email.	Likert scale (1–5)
I verify the format of attachments before opening them.	Likert scale (1–5)
I open attachments from senders I do not know.	Likert scale (1–5)

For evaluation, the answer "yes" was translated to the numerical value "1" and the answer "no" to "0", except for the question "I use the same password for multiple accounts.", where the value 1 was given to the answer "no" and the value 0 to the answer "yes". This way, a higher score represents a deeper understanding and internalization of IT-security awareness actions. Similarly, for the questions with a Likert scale response, the values were translated as $0 =$ Never, $0.25 =$ Rarely, $0.5 =$ Sometimes, $0.75 =$ Often, $1 =$ Always, except for the last question were the values are reversed in order for the higher score to represent a higher level

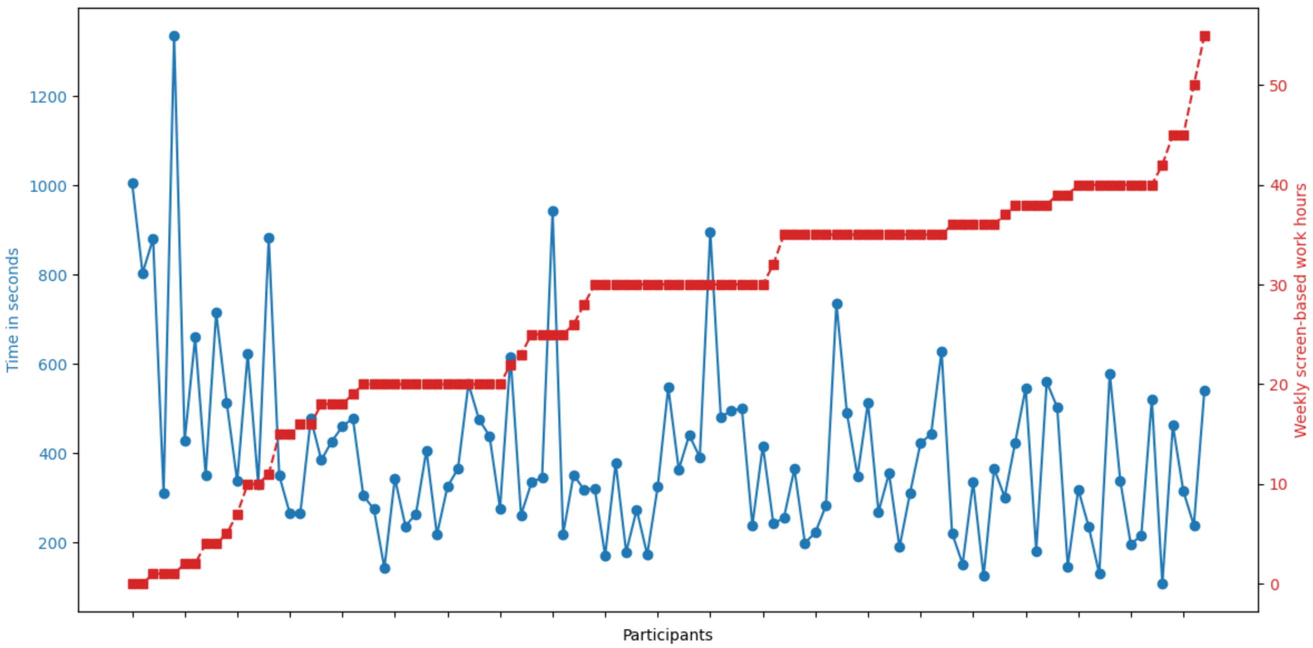


Figure 10. Total time spent on the email sorting task versus weekly screen-based work hours

of IT security awareness. Using these numerical values, the maximum attainable IT-security awareness score was 8, and the minimal score was 0. The mean score was 6.4 ($min = 2.5$, $max = 8$, $std = 1.26$), with a median score of 6.75. The IT-security knowledge level of participants is displayed in Figure 11. The scores varied slightly between industries sectors, with employees in the IT sector showing slightly higher scores than employees in the construction or utilities sector, but not enough to reach statistical significance. Similarly to before, Kendall's correlation test was not able to detect a correlation between the level of IT security knowledge and the number of correctly identified emails. Only a correlation between the IT security knowledge level and the time needed to complete the task was detected ($z = -2.94$, $p = .003$, $r = .20$). Grouping the participants into IT security experts (25th percentile) and novices (75th percentile) shows no differences in number of correctly identified emails (see Figure 12).

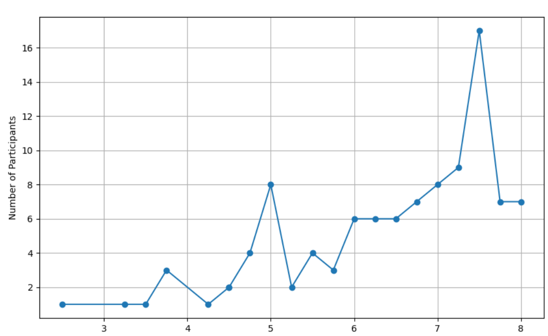


Figure 11. IT security awareness knowledge level of participants

To answer RQ6: it was found that the AOI hits on the Subject and Sender differ significantly between phishing emails and legitimate emails. However, the study was not able to replicate the results found in [11]. On the contrary, it was found that users focused more on the sender when examining legitimate emails than when examining phishing emails. The total AOI hits on all Subject and Sender AOIs were combined (including the PAS-Sender-Address, where the sender address was displayed in the PAS) and it was tested whether these AOI hits differ between phishing emails and legitimate emails. A significant difference was detected by a Mann-Whitney U-test with the values $z = 454328.00$, $p < .001$, $r = .15$, showcasing a significant difference at $\alpha = 0.05$ with small effect. The median for AOI hits on legitimate emails was $Mdn = 494.00$, as compared to a median of $Mdn = 314.00$ for the group of phishing emails. This is shown in Figure 13. This effect might

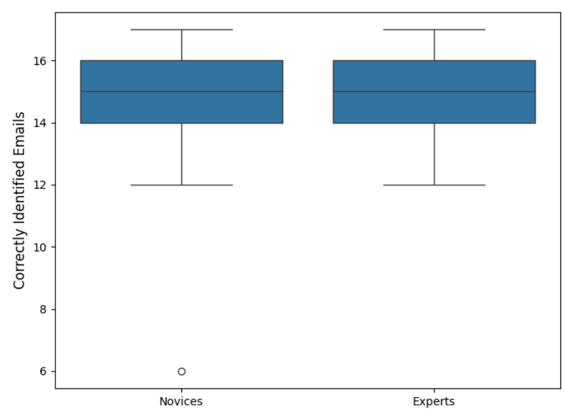


Figure 12. Correctly identified emails by IT security knowledge level

be explained by the difference in data sets between the two studies. It might have been the case that the phishing emails were easy to spot for the participants, whereas the legitimate ones proved to be more of a challenge. Participants expectancy to be "fooled" could have played a role in their skepticism towards legitimate emails. To test this, it was tested whether participants tended to misclassify legitimate emails more often than phishing emails. In [11], the authors found participants to be more likely to misclassify legitimate emails. The same is the case here, where a total of 85% of phishing emails were recognized correctly, in contrast to only 80% of legitimate emails being recognized as such.

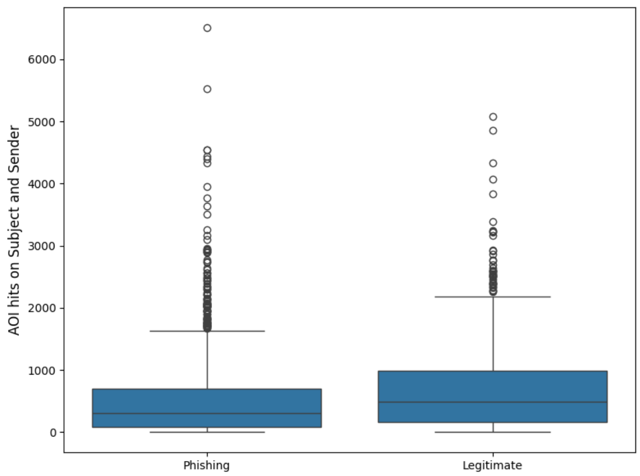


Figure 13. AOI hits on the Subject and Sender Area for phishing emails and legitimate emails

RQ7 builds upon research questions four and five from the original paper [1], offering a deeper analysis of eye-tracking-specific metrics with a focus on the presence of the PAS and its influence on the time spent examining phishing markers. The original paper's AoI analysis indicated that participants with access to the PAS could accurately identify phishing emails equally as efficient while spending less time examining the relevant areas compared to those without the add-on. However, this evaluation was conducted at the group level, without analysing individual participants or emails. Therefore, RQ7 seeks to explore how individual gaze patterns differ between participants who correctly identified most phishing emails and those who misclassified more.

To achieve this, a qualitative analysis is conducted using scarf plots. These visualizations - which are becoming increasingly popular in eye-tracking studies - allow for aggregating gaze movements over time, particularly between AoIs [30].

To compare data at the participant level, appropriate groups must first be defined. Since RQ7 focuses on extremes — participants who correctly identified all phishing emails and those who struggled the most — the groups are determined using quartiles. Examination of the 5th and 95th percentiles for correctly identified phishing emails shows $Q_{0.05} = 8$ $(n = 10)$ and $Q_{0.95} = 12$ $(n = 22)$.

Figure 14. Scarf Plot for the off-brand shoe store email: Visualizing AoI Transitions between phishing markers from participant NOT using the PAS. Participants 1 to 3 are within the $Q_{0.05}$ and participants 4 to 16 are within the $Q_{0.95}$ of correctly identified emails.

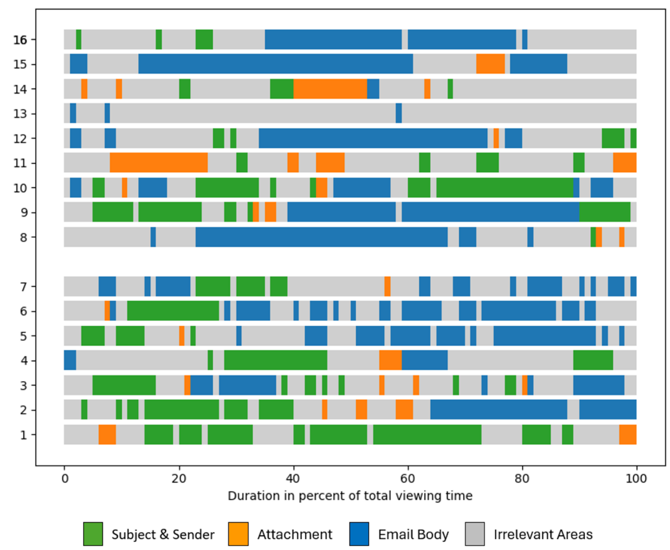


Figure 15. Scarf Plot for the off-brand shoe store email: Visualizing AoI Transitions between phishing markers from participant using the PAS. Participants 1 to 7 are within the $Q_{0.05}$ and participants 8 to 16 are within the $Q_{0.95}$ of correctly identified emails. AoI hits from the PAS and the email itself are combined.

In this case, the 95th percentile consists entirely of participants who correctly identified all 12 phishing emails. In contrast, the 5th percentile group misclassified at least one-third of the phishing emails. Table III provides an overview of participants within the $Q_{0.05}$ range and the phishing emails they misclassified. For clarity and readability, the original participant IDs have been omitted, and participants are renumbered sequentially starting from 1. For all following scarf plots the two groups - with and without the PAS - are separated by

TABLE III. Participants within the $Q_{0.05}$ of falsely identified emails. The phishing emails they fell for are marked as "X"

	Harmful Attachment				Harmful Link				Injunction to send money			
Study Design	Shoe store	Zalando	Mediamarkt	Pustet	Edeka	GMX	DB	iCloud	Schufa	Amazon	Spotify	DHL
Without PAS	X			X			X				X	
Without PAS	X		X	X			X					
Without PAS			X	X						X		X
With PAS	X				X		X				X	
With PAS			X	X			X	X				
With PAS	X						X	X		X		
With PAS		X			X		X	X	X	X	X	X
With PAS	X			X	X		X	X				X
With PAS	X	X	X				X					
With PAS	X		X		X			X				
Sum	**7**	2	5	5	4	0	**8**	5	1	3	3	3

a blank row, with the group that misclassified the most emails listed at the bottom.

All original data, including participants' responses, the stimuli used, and the raw eye-tracking data, can be found on Zenodo (see Section VIII).

Upon reviewing the table, two emails stand out: one from a no-name shoe store and another from the german railway operating company Deutsche Bahn (DB). These were mistakenly classified as legitimate by 7 and 8 out of the 10 participants, making them the focus of the qualitative analysis.

The first of the two phishing emails contained a .zip attachment and a misspelled email address, making it a "bad" phishing email according to the study design.

When analysing the group without the PAS, a clear trend emerges: participants who correctly identified all phishing emails spent more time examining the email body, often scanning this AoI for large sections at a time (see Figure 14). However, participants 5, 15 and 16 stand out in particular, as they spent considerable time looking at irrelevant areas of the screen, areas that could not have contributed to their decision-making. Among those who misclassified the email as legitimate participant 3 stands out. He spent most of his time focusing on the .zip attachment, suggesting that he recognized the potentially harmful file type but did not consider it sufficient enough evidence of a phishing attempt. Interestingly, Participants 1 and 2 ignored the attachment entirely, with Participant 2 not even looking at relevant areas at all.

However, it has to be noted that during the training, several employees emphasized that sending files as a .zip archive is still common practice in small and medium-sized enterprises. Many participants mentioned that in their daily work, they would have reached out or asked a colleague for clarification before making a judgment. Since this option was unavailable in the study, most leaned toward classifying the email as legitimate rather than fraudulent.

A different pattern emerged in the group with the PAS. Here, participants had access to both the email content and additional information from the PAS, highlighting phishing markers. For visualization reasons the scarf plots combine AoIs hits from both the email and the PAS, meaning that participants could examine attachment details within the email or through the PAS, with both being represented as one in the diagram. Participants who correctly identified all phishing emails spent longer periods examining AoIs, switching mainly between different types of information. In contrast, participants who misclassified more emails exhibited fragmented AoI patterns, with frequent short glances at phishing markers (see Figure 15). This suggests they may have mistrusted the PAS and cross-referenced the highlighted phishing markers with the original email content to verify the information manually.

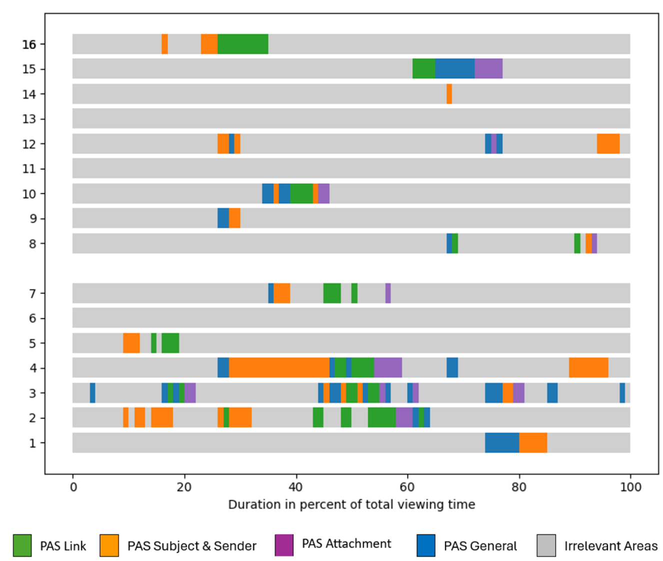


Figure 16. Scarf Plot for the off-brand shoe store email: Visualizing AoI Transitions within the PAS. Participants 1 to 7 are within the $Q_{0.05}$ and participants 8 to 16 are within the $Q_{0.95}$ of correctly identified emails.

However, this trend is not universal. Figure 16 visualizes AoI hits specifically within the PAS. Participants 2 to 4 engaged in the verification process by spending only short periods reviewing the phishing markers highlighted by the PAS, while others barely interacted with the PAS at all. This indicates that participants who misclassified the email either did not trust the PAS or preferred to verify the details manually, if they used the add-on at all. On the other hand, participants 8 to 16, despite rarely using the PAS, all except one looked at the attachment information at least once. Even this brief engagement with this information may have been enough to help them recognize the email as a phishing attempt.

The second phishing email, from Deutsche Bahn, claimed that the recipient's account would be deactivated unless they took action and clicked on a re-activation link. Unlike the previous email, this one included the company logo and had only a minor misspelling in the sender domain (missing the letter "e": support@**deutsch**-bahn.de). Due to its more convincing appearance, it was categorized as a "good" phishing email in the study design.

Figure 17. Scarf Plot for the Deutsche Bahn email: Visualizing AoI Transitions between phishing markers from participant NOT using the PAS. Participants 1 to 3 are within the $Q_{0.05}$ and participants 4 to 16 are within the $Q_{0.95}$ of correctly identified emails.

For participants without the PAS, Figure 17 shows no clear difference between those who fell for the email and those who correctly identified it as phishing. As with the previous email, Participant 2 barely looked at any relevant areas, which might suggest he did not take the study seriously or maybe was overwhelmed with the task. The same is true for participant 16 in this scarf plot.

However, when analysing the group with the PAS, a similar trend to the previous phishing email emerges. Participants who misclassified the email as legitimate exhibited more frequent, short, and abrupt switches between different AoIs (see Figure 18).

Interestingly, when focusing solely on PAS usage, participants who correctly identified the email as phishing showed significantly higher engagement with the PAS compared to the previous "bad" phishing email (see Figure 19). This suggests that the PAS is particularly helpful in more subtle cases where crucial phishing markers are easy to overlook. Additionally, the increase in PAS usage toward the end of the decision-making process indicates that participants trusted the information provided by the PAS, using it either as the basis for their decision or at least as a final verification.

Figure 18. Scarf Plot for the Deutsche Bahn email: Visualizing AoI Transitions between phishing markers from participant using the PAS. Participants 1 to 7 are within the $Q_{0.05}$ and participants 8 to 16 are within the $Q_{0.95}$ of correctly identified emails. AoI hits from the PAS and the email itself are combined.

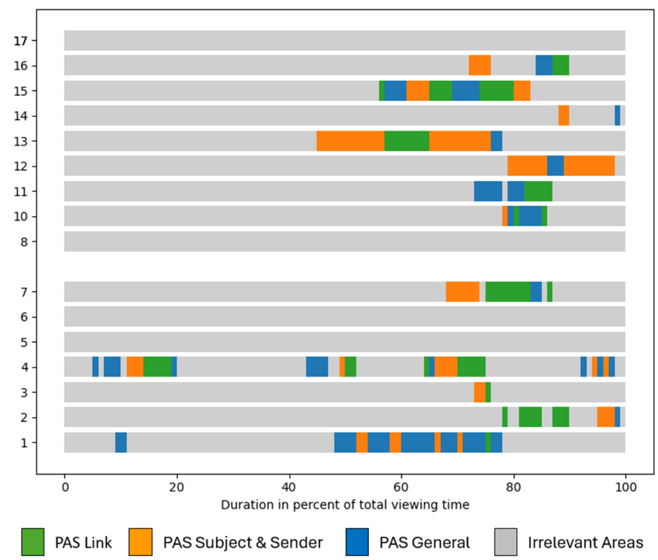


Figure 19. Scarf Plot for the Deutsche Bahn email: Visualizing AoI Transitions within the PAS. Participants 1 to 7 are within the $Q_{0.05}$ and participants 8 to 16 are within the $Q_{0.95}$ of correctly identified emails.

Concluding this, RQ7 can be answered: gaze patterns do differ between individuals who correctly identified all phishing emails and those who misclassified more to some extent. However, these differences are not uniform but manifest in multiple ways. First of all, participants who misclassified more emails tend to have shorter, more abrupt AoI viewing patterns, frequently switching between AoIs, particularly in the group with the PAS. Secondly, PAS usage varies based on phishing email complexity. When phishing markers were less obvious, participants who correctly identified all emails were more likely to use the PAS toward the end of their decision-making process. Adding to this, for easier-to-detect phishing emails, participants who sorted all emails correctly studied the email body more carefully, suggesting they were quick to identify obvious phishing markers and validated their findings by examining additional cues.

A. Summary of results

Demographic factors such as age, gender, and highest general education degree were found to have no significant impact on phishing recognition rates. However, working in different industry sectors and the number of weekly screen-based work hours had a notable effect on the time participants needed to recognize phishing attempts. Employees from the IT sector were able to recognize phishing emails much faster compared to those from the construction or utilities sectors. Interestingly, prior knowledge of the sender before the study did not influence participants' ability to identify phishing emails.

Grouping the participants into IT security experts and novices revealed that participants with higher IT security knowledge were significantly faster at recognizing phishing attempts. However, this did not translate to a higher accuracy in identifying phishing emails. This finding partially aligns with the results of Ribeiro et al. [11], where users were more likely to misclassify legitimate emails than phishing ones. However, unlike the results reported in [11], participants in this study spent significantly more time examining the sender area in legitimate emails, while phishing emails did not garner as much attention in this area.

Further analysis of Areas of Interest (AoI) hits revealed that participants who misclassified more phishing emails tended to glance at relevant phishing markers for shorter, more abrupt periods. In contrast, participants who correctly identified phishing emails appeared to engage with the AoIs more thoroughly, especially when the phishing email was well-made. Notably, AoI hits on the PAS indicated that participants who successfully identified phishing emails relied on the PAS primarily when the email was particularly convincing and typically just before making their final decision.

VI. Usability analysis

The previous paper already highlighted a positive correlation between high usability ratings of the PAS and participants' ability to correctly identify phishing emails [1]. Combining these findings with insights from RQ7, the question arises: Do usability ratings of the PAS also correlate with the users' gaze patterns, particularly when interacting with the PAS itself? Previous research in fields such as machine learning and human-computer interaction has shown that specific eye movement patterns can reflect the usability of a system [31] [16]. This observation leads to the introduction of the last research question 8: Does the usability (and possibly the user experience) of the PAS relate to participants' gaze patterns during their?

For RQ8, only participants who interacted with the PAS will be considered. Furthermore, since this analysis is not tied to specific email stimuli, gaze patterns from all emails in the study will be aggregated into a single timeline and analyzed as a whole.

1) Usability and UX Questionnaires: Two types of questionnaires were used to assess usability and user experience (UX): the System Usability Scale (SUS) and the short version of the User Experience Questionnaire (UEQ-S).

The SUS measures perceived system usability through a ten-item questionnaire. Developed by John Brooke in the late 1990s, the SUS was designed to align with the ISO 9241-110 standard, making it universally applicable across different systems and contexts [32]. Respondents answer on a five-point Likert scale, with half of the items formulated to elicit agreement and the other half to elicit disagreement. The final usability score is calculated by weighing the responses, yielding a score between 0 and 100, with higher scores indicating better usability. In some cases, this score is further categorized into grades from A to F, with a score around 50 or lower indicating poor usability [33].

In contrast, the UEQ assesses not only usability but also the overall user experience (UX). Developed by Laugwitz et al., the UEQ distinguishes between pragmatic quality (associated with usability) and hedonic quality (related to UX) [34]. It's important to note that these two dimensions represent one of many perspectives on usability and UX, with alternative definitions existing in the literature [35]. The UEQ originally consists of 26 items; however, since evaluating the PAS's usability and UX was not the primary focus of this study, only the short version - UEQ-S - was used. This version contains eight items, offering a concise but comprehensive assessment of both pragmatic and hedonic quality [36]. Like the SUS, it uses a Likert scale, but with seven points instead of five. Scores for both pragmatic and hedonic quality are calculated by averaging the responses to the relevant items, with scores below 3.2 indicating poor results and those above 4.8 indicating high results [37].

2) Usability and UX Results: Similar to RQ7, percentiles are employed to categorize individuals into groups representing opposite extremes on the usability scale. Yet, in this case, the $Q_{0.10}$ and $Q_{0.90}$ percentiles are used, as the usability and UX scores fluctuated more than the number of correctly identified phishing emails. This leads to the following percentiles:

- SUS $Q_{0.1} = 52.75$ ($n = 6$)
- SUS $Q_{0.9} = 92.5$ ($n = 7$)
- UEQ-S Pragmatic Quality $Q_{0.1} = 4$ ($n = 8$)
- UEQ-S Pragmatic Quality $Q_{0.9} = 7$ ($n = 10$)
- UEQ-S Hedonic Quality $Q_{0.1} = 3.25$ ($n = 7$)
- UEQ-S Hedonic Quality $Q_{0.9} = 6.25$ ($n = 7$)

Starting with usability, an analysis of PAS usage — measured by any Area of Interest (AoI) hits within the sidebar — reveals a significant difference in behavior between participants who rated the tool as less usable and those who rated it as highly usable. Participants who perceived the PAS as less usable (SUS Score ≤ 52.75) used the tool significantly

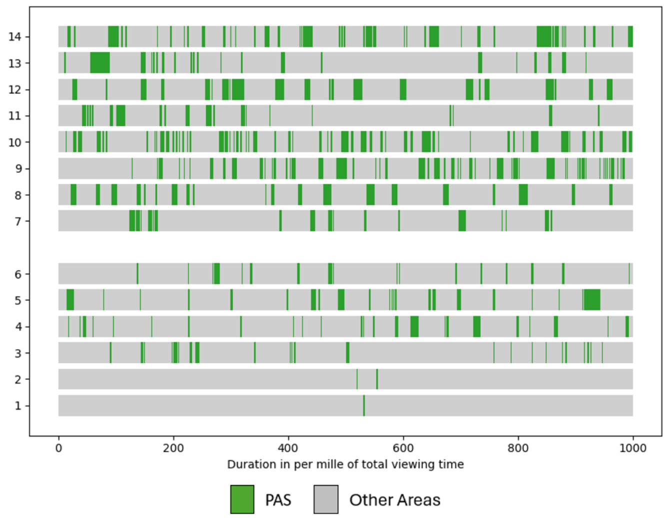


Figure 20. Scarf plot visualizing AoI hits within the PAS in relation to the SUS score. Participants 1 to 6 are within the $Q_{0.1}$ and participants 7 to 14 are within the $Q_{0.9}$ the SUS score.

less (see Figure 20). In the scarf plots, green areas indicate PAS usage, while grey areas represent time spent looking at the email itself or other PAS-unrelated screen areas. This viewing behavior was expected, as both low usability and user experience typically leads to reduced acceptance and adoption of software [38].

In contrast, participants who rated the PAS as highly usable (SUS Score ≥ 92.5) tended to use the tool more frequently and for longer periods. A similar pattern emerges when examining the pragmatic quality results of the UEQ-S (see Figure 21). Those who rated the PAS's pragmatic quality as low to neutral (UEQ-S Pragmatic Quality Score ≤ 4) also used the tool less. However, one outlier - participant 2 - used the tool just as much as participants who rated the pragmatic quality as high (UEQ-S Pragmatic Quality Score $= 7$).

Figure 21. Scarf plot visualizing AoI hits within the PAS in relation to the UEQ Pragmatic Quality score. Participants 1 to 6 are within the $Q_{0.1}$ and participants 7 to 14 are within the $Q_{0.9}$ the UEQ Pragmatic Quality score.

Interestingly, the results for hedonic quality show an unexpected trend. As seen in Figure 22, participants who rated the hedonic quality as low to neutral (UEQ-S Hedonic Quality Score ≤ 3.25) actually spent more time looking at the tool than those who rated it as high (UEQ-S Hedonic Quality Score ≥ 6.25). This could indicate confusion or a lack of trust in the PAS, leading to prolonged examination of the tool. Alternatively, participants proficient in detecting phishing emails may generally need less time overall, and their efficiency leads to them spending less time with the PAS. In a study setting where participants are primed and racing against the clock, this seems plausible — there is little time to appreciate the design, while visual irregularities may cause the participant to stop. However, without additional UX data, this cannot be explained definitively.

Figure 22. Scarf plot visualizing AoI hits within the PAS in relation to the UEQ Hedonic Quality score. Participants 1 to 6 are within the $Q_{0.1}$ and participants 7 to 14 are within the $Q_{0.9}$ the UEQ Hedonic Quality score.

Despite these contradicting findings, all three scarf plots clearly demonstrate that gaze patterns vary between individuals with differing usability and UX ratings. Nevertheless, hypothesis H8 must be partially rejected: While higher usability did indeed lead to increased PAS usage, participants who perceived UX as high actually spent less time looking at the tool.

VII. Limitations

It is a consistent pattern throughout the entire study that differences between groups were only observed in the amount of time needed to complete the email classification task, but never in the number of correctly identified emails. This could be attributed to the nature of statistical tests: many tied values within a recorded quantity reduce the statistical power to detect significant differences. As a result, while participants may vary in the speed at, which they complete the task, their accuracy appears to remain consistently high across all groups. This finding emphasizes the importance of measuring multiple dimensions of performance when evaluating differences between groups. Focusing solely on one aspect, such as accuracy, may overlook meaningful variations in other areas, such as task efficiency.

VIII. Conclusion and Future Work

The findings from this study indicate that prior training or specific knowledge, such as being an IT security expert, do not influence the number of correctly identified phishing emails. This suggests that errors in identifying phishing emails are more likely because of genuine user mistakes and oversights, rather than a lack of knowledge. Despite this, it was observed that certain factors, such as education level, industry sector, IT security knowledge, and weekly screen-based work hours, had a significant impact on the time required to recognize phishing attempts. Participants with higher education levels, more IT security knowledge, greater weekly screen exposure, or those working in the IT sector, performed the task more efficiently, needing less time to identify phishing emails. Interestingly, no significant differences were found between individuals with IT-related backgrounds and those without, indicating that phishing detection training may be beneficial for all participants, regardless of their profession or expertise.

The nature of statistical testing, particularly with discrete variables like the number of correctly identified emails, makes it difficult to detect significant differences between groups when there are many tied values. While it is challenging to present participants with large datasets due to time constrains, especially demographic correlation analysis would benefit from a bigger dataset. With more data it could even be possible to measure influences of demographic factors, which yielded no effect in this study. However, this data limitation lies in the nature of eye-tracking studies, which are not infinitely scalable due to the need for specific technical equipment and participant monitoring by the conductors.

Future research could explore ways to improve phishing detection across all user groups, including those with limited IT security knowledge. Further studies could investigate whether longer or more detailed training sessions can enhance detection accuracy and speed for participants with less prior knowledge. Additionally, expanding the study to include larger and more varied dataset, perhaps with more frequent exposure to phishing attempts or even a redesigned version of the PAS, would help address the limitations of the current approach and provide further insights into the role of experience and training in phishing recognition.

Acknowledgments

This study was conducted as part of the EU-funded EDIH *Digital Innovation Ostbayern (DInO)*. DInO is funded by the European Union (Project Reference 101083427) and the European Funds for Regional Development (EFRE) (Project Reference 20-3092.10-THD-105). The eye-tracking study was approved by the Joint Ethics Committee of the Bavarian Universities (GEHBa) with the reference number GEHBa-202312-V-155-R.

This article and the one presented in [1] are the result of a joint collaboration between all authors, with each contributing equal effort. Therefore, authorship is alternated in this paper, which does not reflect a change in contribution levels.

The individuals in Figure 4 - despite being only shown from behind - both agreed to be shown in this paper.

Data

The eye-tracking and questionnaire data collected and evaluated in this study is free to use and can be found on Zenodo under the following link doi.org/10.5281/zenodo.13171791.

References

[1] M. Kristen, F. Engl, and J. Mottok, “Enhancing phishing detection: An eye-tracking study on user interaction and oversights in phishing emails,” in *SECURWARE 2024, The Eighteenth International Conference on Emerging Security Information, Systems and Technologies*, 2024.

[2] Bundesamt für Sicherheit in der Informationstechnik (BSI), “Die Lage der IT-Sicherheit in Deutschland 2024,” de, 2022.

[3] M. Bada, A. M. Sasse, and J. R. Nurse, “Cyber security awareness campaigns: Why do they fail to change behaviour?” *arXiv preprint arXiv:1901.02672*, 2019.

[4] A. M. Sasse, S. Brostoff, and D. Weirich, “Transforming the ‘weakest link’ — a human/computer interaction approach to usable and effective security,” *BT Technology Journal*, vol. 19, no. 3, pp. 122–131, Jul. 1, 2001. DOI: 10.1023/A:1011902718709.

[5] A. Heinemann and G. Schembre, “Zur Wirksamkeit von Security Awareness Maßnahmen,” ger, in *DACH Security Tagungsband 2017: Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven*, P. Schartner and A. Baumann, Eds., Klagenfurt (Österreich): Alpen-Adria-Universität, 2017, ISBN: 978-3-00-057290-6.

[6] D. Lain, K. Kostiainen, and S. Capkun, “Phishing in organizations: Findings from a large-scale and long-term study,” *2022 IEEE Symposium on Security and Privacy (SP)*, p. 9, 2022.

[7] H. Abroshan, J. Devos, G. Poels, and E. Laermans, “Phishing happens beyond technology: The effects of human behaviors and demographics on each step of a phishing process,” *IEEE Access*, vol. 9, pp. 44 928–44 949, 2021. DOI: 10.1109/ACCESS.2021.3066383.

[8] A. Darwish, A. E. Zarka, and F. Aloul, “Towards understanding phishing victims’ profile,” in *2012 International Conference on Computer Systems and Industrial Informatics*, 2012, pp. 1–5. DOI: 10.1109/ICCSII.2012.6454454.

[9] A. K. Ghazi-Tehrani and H. N. Pontell, “Phishing evolves: Analyzing the enduring cybercrime,” *Victims & Offenders*, vol. 16, no. 3, pp. 316–342, 2021. DOI: 10.1080/15564886.2020.1829224.

[10] J. L. Orquin and K. Holmqvist, “Threats to the validity of eye-movement research in psychology,” *Behavior Research Methods*, vol. 50, no. 4, pp. 1645–1656, Aug. 2018. DOI: 10.3758/s13428-017-0998-z.

[11] L. Ribeiro, I. Guedes, and C. Cardoso, “Eyes on phishing emails: An eye-tracking study,” *Journal of Experimental Criminology*, 2024.

[12] G. A. Thomopoulos, D. P. Lyras, and C. A. Fidas, “A systematic review and research challenges on phishing cyberattacks from an electroencephalography and gaze-based perspective,” *Personal and Ubiquitous Computing*, 2024.

[13] D. Baltuttis and T. Teubner, “Effects of visual risk indicators on phishing detection behavior: An eye-tracking experiment,” *Computers & Security*, vol. 144, p. 103 940, 2024. DOI: 10.1016/j.cose.2024.103940.

[14] S. Zhuo, R. Biddle, J. Daniel Recomendable, G. Russello, and D. Lottridge, "Eyes on the phish(er): Towards understanding users' email processing pattern and mental models in phishing detection," in *Proceedings of the 2024 European Symposium on Usable Security*, ser. EuroUSEC '24, Association for Computing Machinery, 2024, pp. 15–29, ISBN: 9798400717963. DOI: 10.1145/3688459.3688465.

[15] L. Zhou, J. Lim, and D. Zhang, "The effects of email illegitimacy and phishing behavior training on eye movement behavior in spear phishing detection," in *Proceedings of the 57th Hawaii International Conference on System Sciences*, 2024.

[16] B. Xing *et al.*, "User-attention based product aesthetics evaluation with image and eye-tracking fusion data analysis," in *2023 15th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Aug. 2023, pp. 84–87.

[17] P. Sulikowski and T. Zdziebko, "Deep learning-enhanced framework for performance evaluation of a recommending interface with varied recommendation position and intensity based on eye-tracking equipment data processing," *Electronics*, vol. 9, no. 2, p. 266, Feb. 2020, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[18] M. C. Sáiz-Manzanares *et al.*, "Analysis of the learning process through eye tracking technology and feature selection techniques," *Applied Sciences*, vol. 11, no. 13, p. 6157, Jan. 2021, Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

[19] T. Ezer, M. Greiner, L. Grabinger, F. Hauser, and J. Mottok, "Eye tracking as technology in education: Data quality analysis and improvements," in *ICERI2023 Proceedings*, ser. 16th annual International Conference of Education, Research and Innovation, Seville, Spain: IATED, Nov. 2023, pp. 4500–4509, ISBN: 978-84-09-55942-8. DOI: 10.21125/iceri.2023.1127.

[20] G. Kaiser, *Marktanteile der Top 10 E-Mail-Clients weltweit im März 2025*, https://de.statista.com/statistik/daten/studie/688163/umfrage/marktanteile-der-e-mail-clients-weltweit/, Accessed: 2025-05-19.

[21] C. Blake, "Eye-Tracking: Grundlagen und Anwendungsfelder," ger, in *Handbuch standardisierte Erhebungsverfahren in der Kommunikationswissenschaft*, W. Möhring and D. Schlütz, Eds., Wiesbaden: Springer Fachmedien Wiesbaden, 2013, pp. 367–387, ISBN: 978-3-531-18776-1. DOI: 10.1007/978-3-531-18776-1_20.

[22] M. Burch and K. Kurzhals, "Teaching eye tracking: Challenges and perspectives," *Proceedings of the ACM on Human-Computer Interaction*, vol. 8, no. ETRA, pp. 1–17, 2024.

[23] B. T. Carter and S. G. Luke, "Best practices in eye tracking research," *International Journal of Psychophysiology*, vol. 155, pp. 49–62, 2020.

[24] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, Dec. 1965. DOI: 10.1093/biomet/52.3-4.591. eprint: https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf.

[25] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[26] H. Abdi, "The bonferonni and šidák corrections for multiple comparisons," *Encyclopedia of Measurement and Statistics*, vol. 3, Jan. 2007.

[27] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[28] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1-2, pp. 81–93, Jun. 1938. DOI: 10.1093/biomet/30.1-2.81. eprint: https://academic.oup.com/biomet/article-pdf/30/1-2/81/423380/30-1-2-81.pdf.

[29] W. G. Cochran, "The χ^2 Test of Goodness of Fit," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 315–345, 1952. DOI: 10.1214/aoms/1177729380.

[30] C.-K. Yang and C. Wacharamanotham, "Alpscarf: Augmenting scarf plots for exploring temporal gaze patterns," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '18, Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–6, ISBN: 9781450356213. DOI: 10.1145/3170427.3188490.

[31] P. Wang, H. Yang, J. Hou, and Q. Li, "A machine learning approach to primacy-peak-recency effect-based satisfaction prediction," *Information Processing & Management*, vol. 60, no. 2, p. 103 196, Mar. 1, 2023.

[32] J. Brooke, "SUS: A 'quick and dirty' usability scale," *Usability Evaluation in Industry*, vol. 1, pp. 189–194, 1996.

[33] R. A. Grier, A. Bangor, P. Kortum, and S. C. Peres, "The system usability scale: Beyond standard usability testing," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 187–191, 2013. DOI: 10.1177/1541931213571042.

[34] B. Laugwitz, T. Held, and M. Schrepp, "Construction and evaluation of a user experience questionnaire," in *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4*, Springer, 2008, pp. 63–76.

[35] M. Richter and M. D. Flückiger, *Usability und UX kompakt* (IT kompakt). Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[36] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Design and evaluation of a short version of the user experience questionnaire (UEQ-S)," *International Journal of Interactive Multimedia and Artificial Intelligence, 4 (6), 103-108.*, 2017.

[37] S. Martin, "User experience questionnaire handbook," de, 2023.

[38] M. Hassenzahl, "The effect of perceived hedonic quality on product appealingness," *International Journal of Human-Computer Interaction*, vol. 13, no. 4, pp. 481–499, 2001.

From Theory to Practice: Evaluating and Enhancing Kolmogorov-Arnold Networks (KAN) Robustness Under Adversarial Conditions

Evgenii Ostanin
Toronto Metropolitan University
Toronto, Canada
eostanin@torontomu.ca

Nebojsa Djosic
Toronto Metropolitan University
Toronto, Canada
nebojsa.djosic@torontomu.ca

Fatima Hussain
Toronto Metropolitan University
Toronto, Canada
fatima.hussain@torontomu.ca

Salah Sharieh
Toronto Metropolitan University
Toronto, Canada
salah.sharieh@torontomu.ca

Alexander Ferworn
Toronto Metropolitan University
Toronto, Canada
aferworn@torontomu.ca

***Abstract*—Kolmogorov–Arnold Networks have emerged as promising architectures thanks to their adaptive activation functions and enhanced interpretability. However, their robustness under adversarial conditions remains underexplored. In this study, we evaluated four variants of Kolmogorov-Arnold Networks, Linear, Fourier, Jacobi, and Chebyshev against Gaussian noise and two gradient-based attacks (the Fast Gradient Sign Method and Projected Gradient Descent). Through detailed comparative analyses and adversarial training experiments with varying mixes of perturbed data, we reveal substantial differences in resilience across variants and relative to a multilayer perceptron baseline. Our results show that targeted adversarial training materially improves robustness under strong adversarial attacks. In particular, including only 5% Fast Gradient Sign Method examples and 5% Projected Gradient Descent examples in the training set restores between 60 and 90 percentage points of accuracy against these attacks. These findings clarify the factors influencing Kolmogorov–Arnold Network robustness and validate adversarial training as a practical hardening strategy for deployment in adversarially challenging environments.**

Keywords-Kolmogorov-Arnold Networks; KAN; MNIST; FGSM; PGD; Classification; Adversarial Training.

I. Introduction

The rapid advancement of Machine Learning (ML) has led to increasingly sophisticated models that perform well across a variety of tasks. Among these developments, Kolmogorov-Arnold Networks (KANs) represent a novel approach based on the Kolmogorov-Arnold representation theorem. KANs enhance interpretability and flexibility through learnable activation functions, dynamically adapting to data variations and potentially improving model robustness and generalization. Their robustness, however, especially under Adversarial Attacks (AA) and noisy data, remains an underexplored domain.

This paper extends our previous work [1], which analyzed the robustness of KAN architectures under AA. In that study, the focus was on evaluating the performance of different KAN implementations against Gaussian noise, Fast Gradient Sign Method (FGSM), and Projected Gradient Descent (PGD) attacks, comparing their vulnerabilities to a Multi-Layer Perceptron (MLP) classifier. Our findings showed that while KANs achieved higher accuracy than MLPs in clean environments, they exhibited significant drops in accuracy when subjected to adversarial perturbations, with PGD having the most severe impact.

Traditional MLPs often struggle with capturing complex nonlinear relationships due to their reliance on fixed activation functions and linear weight matrices. This limitation can lead to suboptimal generalization in adversarial settings or when handling noisy data. To address these challenges, KANs introduce learnable activation functions on edges, allowing them to adapt dynamically to input variations, offering potential advantages in robustness and interpretability over traditional models [2].

The increasing sophistication of AA poses significant challenges for deep learning models, particularly in security-critical applications such as autonomous systems and cybersecurity. Attacks like the FGSM and PGD exploit weaknesses in models by introducing subtle alterations to input data. Additionally, the growing deployment of ML models in real-world applications exposes them to environmental noise, which can further degrade performance [3]–[5]. As a result, robustness against both AA and noise is an important requirement for deploying ML models in production and practical settings [6].

This extended paper expands our prior findings [1], [7] by systematically evaluating adversarial training as a novel approach to enhance the robustness of multiple KAN architectures. Specifically, we evaluate how different adversarial training compositions impact KAN resilience to AA. Our primary contributions include:

- A reassessment of the vulnerabilities of KAN architectures under adversarial conditions.
- Analyzing the impact of adversarial training with varying proportions of clean and adversarially perturbed samples.
- A comparative analysis of how different KAN models respond to adversarial training, highlighting the strengths and weaknesses of each approach.
- A discussion of the broader implications of KAN robustness and future research directions.

Key Results: Unprotected KAN models can lose up to 88% accuracy under strong PGD attacks. Injecting just 5%

adversarial samples per AA into the training set restores 60 to 90 percentage points of robustness against FGSM and PGD across all KAN variants. However, the Fourier KAN remains highly sensitive to Gaussian noise. Its noise accuracy stays below 20% even after adversarial training. These findings underscore the need for variant-specific hardening strategies.

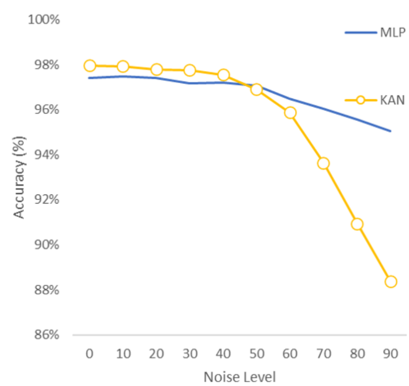


Figure 1. Model Accuracy Degradation After Noise Attack.

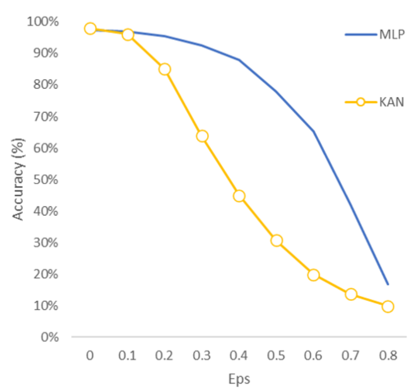


Figure 2. Model Accuracy Comparison After FGSM Attack.

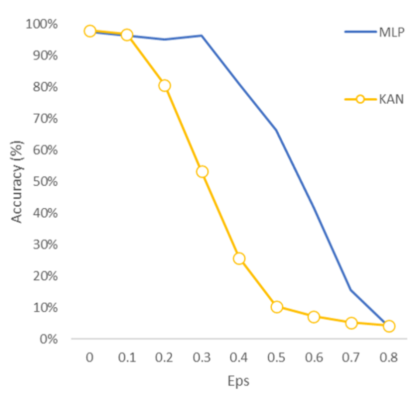


Figure 3. Model Accuracy Comparison After PGD Attack.

To support our initial findings, we include a series of visualizations. Figure 1 illustrates the accuracy degradation of MLPs and KANs under varying levels of Gaussian noise. Figure 2 shows accuracy degradation under increasing FGSM attack strength (eps.), highlighting KANs' greater sensitivity compared to MLPs. Finally, Figure 3 presents accuracy degradation under PGD attacks, where KANs demonstrate the most significant performance drop.

Paper Structure: The remainder of this paper is organized as follows: Section II reviews related work, including research on adversarial robustness and KAN applications. This section provides an overview of prior studies on KANs and AAs, positioning our work within the existing body of research. We discuss advancements in adversarial training techniques and their effectiveness in improving model resilience.

Section III details the methodology, including dataset preparation, attack methodologies, and adversarial training strategies. We describe the experimental setup, including the architecture of the tested KAN models, the parameters used for adversarial training, and the generation of adversarial examples using FGSM and PGD. This section also explains how different compositions of training data impact model robustness.

Section IV presents experimental results, evaluating the impact of adversarial training on model robustness. We provide a comparative analysis of the tested models under various adversarial conditions, supported by visualizations and performance metrics. This section highlights key trends observed across different KAN architectures and discusses the significance of adversarial training in mitigating accuracy degradation.

Sections V and VI conclude with a discussion of key findings and future research directions. We summarize the major contributions of this work, analyze the broader implications for secure ML applications, and propose areas for future exploration, including testing on more complex datasets and refining adversarial training techniques for enhanced KAN resilience.

II. Related Work

The robustness of ML models under adversarial conditions is critical for ensuring their reliability in real-world deployments, particularly in safety-critical applications. While traditional Neural Network (NN) architectures like MLPs have been extensively studied for their vulnerability to adversarial perturbations, KANs, with their unique architecture based on learnable activation functions, presents new opportunities and challenges in terms of robustness. This section provides an overview of foundational concepts and prior research related to KANs architectures, the underlying Kolmogorov-Arnold representation theorem, AAs, and adversarial training strategies, highlighting key insights and existing gaps in the literature.

A. *Kolmogorov-Arnold Representation Theorem*

KANs represent a novel NN architecture derived from the Kolmogorov-Arnold representation theorem, providing a compelling alternative to traditional MLPs. Figure 4 from

[2] clearly illustrates the architectural differences between traditional MLP and KAN models. This innovative architecture fundamentally changes the traditional NN paradigm by introducing learnable activation functions along network edges, replacing the conventional fixed activation functions applied at nodes. The learnability of these functions allows for greater flexibility and interpretability, enabling the KAN models to dynamically adapt their internal transformations during training, potentially resulting in improved model generalizations, and adaptability to diverse and complex datasets.

The foundational basis of KAN architectures lies in the Kolmogorov-Arnold Representation Theorem, first introduced by Andrey Kolmogorov in 1957 and later refined by Vladimir Arnold in 1963. Commonly referred to as the superposition theorem, it mathematically states that any continuous multivariate function $f(x_1, \ldots, x_n)$ defined within a bounded domain can be represented as a superposition of continuous univariate functions. Formally, the theorem is expressed as follows:

$$f(x) = f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \quad (1)$$

In (1) $\phi_{q,p} : [0,1] \to \mathbb{R}$ are continuous inner functions, and $\Phi_q : \mathbb{R} \to \mathbb{R}$ represent continuous outer functions.

KAN models leverage this theorem by explicitly learning these univariate functions, typically using spline-based methods due to their computational efficiency, smoothness properties, and interpretability.

However, alternative activation functions beyond splines exist and may offer advantages depending on specific applications [8], [9]. Fourier-based activation functions, such as those employed in Naïve Fourier KAN [8], effectively handle periodic data and signals due to their inherent periodic properties. Polynomial-based activations, such as those used in Jacobi KAN and Chebyshev KAN [9], can provide computational simplicity while offering superior approximation capabilities in scenarios requiring less flexibility or complexity. Chebyshev polynomials, in particular, are notable for their numerical stability and efficient approximation characteristics for certain classes of functions [9].

Ultimately, identifying the optimal activation function involves balancing computational efficiency, robustness to adversarial perturbations, and task-specific performance requirements. This critical consideration, along with practical implications and empirical evaluations under adversarial scenarios, is addressed thoroughly in the experiments and results presented in later sections of this paper.

B. Potentials and Limitations of KANs

KANs have been proposed as an innovative NN architecture offering unique advantages in interpretability and computational efficiency. Several studies have investigated their performance across various tasks, especially in computer vision. For instance, [10] evaluated KANs against established architectures such as MLP-Mixer, Convolutional Neural Networks (CNNs), and Vision Transformers (ViTs) on widely-used benchmarks. The study highlighted that KAN models notably outperformed MLP-Mixer on datasets like CIFAR-10 and CIFAR-100, demonstrating the model's potential for achieving competitive accuracy. However, the same research observed that KAN architectures fell short when compared directly with deeper convolution-based models, specifically ResNet-18. Still, the computational efficiency advantage was evident, indicating that KANs could offer significant benefits in scenarios where resource constraints and computational efficiency are critical [11].

Further illustrating KANs' potentials, [10] also showed that KAN architectures achieve performance comparable to CNN and traditional MLP architectures on simpler image datasets, such as MNIST and CIFAR-10, with a considerably reduced number of parameters and lower computational requirements. This efficiency positions KANs as particularly suitable for deployment in resource-constrained environments, such as edge devices or embedded systems, where model size and computational efficiency are critical constraints.

Nevertheless, several studies have also highlighted notable limitations of KANs, particularly their sensitivity to noise. Research presented in [3] and [4] emphasizes that KANs exhibit significant performance degradation even when exposed to relatively small noise perturbations. These studies revealed that KANs can sometimes underperform compared to MLPs when the input data contains noise or irregularities, suggesting potential vulnerability in practical, real-world conditions. The spline-based activation functions used within KANs, while beneficial for smooth and continuous approximations, may contribute to increased sensitivity when encountering noisy inputs, as subtle perturbations can alter spline approximations disproportionately.

Moreover, the computational demands associated with spline optimization may exacerbate the sensitivity to noisy inputs, as these functions inherently attempt to closely fit the training data, increasing susceptibility to overfitting on noisy samples. These observations are further supported in [12], that highlight potential limitations of KANs in hardware and computational settings, particularly when working with complex datasets that demand higher computational resources. Their findings indicate that the increased complexity of learnable spline functions might lead to diminishing returns, where additional computational costs do not necessarily translate into proportional performance gains.

Similarly, [11] concludes that the practical advantages of KANs might not be evident for more challenging, complex datasets such as CIFAR-10, where traditional NN architectures like CNNs and ResNets typically dominate. They argue that despite their theoretical appeal and potential interpretability advantages, the practical benefits of employing KANs in more challenging or high-dimensional scenarios remain uncertain and require further validation.

Given these mixed findings, the robustness and practical efficiency of KANs need careful evaluation across diverse

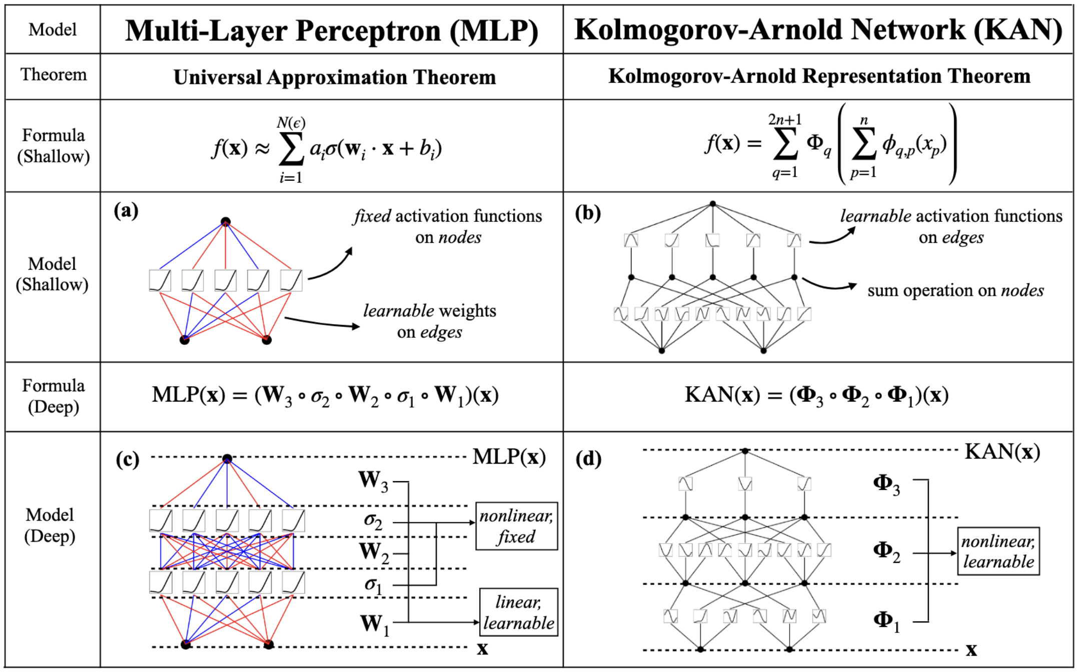


Figure 4. KAN vs MLP architectures compared, source: [2].

scenarios, datasets, and types of perturbations. While KANs clearly demonstrate potential for specific use-cases, particularly those prioritizing interpretability and computational efficiency, their sensitivity to adversarial and environmental noise requires comprehensive assessment and mitigation strategies, motivating further research into enhanced training techniques, such as adversarial training, which will be explored in subsequent sections of this paper.

C. Adversarial Attacks

Recent advances in ML have sparked significant interest in understanding and mitigating vulnerabilities inherent to deep learning models. Central to this investigation are AAs, which strategically exploit vulnerabilities in models by introducing carefully crafted perturbations to the input data. The research into AAs has been especially vigorous in the computer vision domain, given the sensitivity of image-based models to subtle input changes that can drastically alter outputs [6], [13]. This has significant implications for applications relying heavily on image recognition, such as autonomous vehicles, security systems, and facial recognition technologies.

Among the numerous AA techniques developed, the FGSM and PGD have emerged as prominent benchmarks. FGSM, introduced by [14], crafts adversarial examples by leveraging gradients to add minimal perturbations that mislead the model's predictions. Due to its computational simplicity and effectiveness, FGSM remains widely used for initial robustness assessments. Conversely, PGD, introduced by [15], applies an iterative optimization-based procedure to find more potent perturbations, typically resulting in stronger attacks that are more challenging for models to withstand. Due to its iterative nature, PGD has become the de facto standard for rigorous robustness evaluations, especially in the context of image classification tasks where even minor perturbations to input data can lead to substantial accuracy degradation [16].

Several defenses against these attacks have been proposed, ranging from detection and preprocessing approaches to robust training methodologies. Techniques such as adversarial example detection [17], diversity-enhancing strategies to mitigate attacks [18], and methods leveraging momentum to optimize the defense mechanism against PGD [19], have shown varying degrees of effectiveness. Despite these advancements, FGSM and PGD remain critically important for the systematic evaluation of model robustness due to their simplicity, efficiency, and established status in literature.

Tools like the Adversarial Robustness Toolbox (ART) [20] have been instrumental in facilitating systematic experimentation and reproducibility in adversarial research by providing standardized methods for generating adversarial examples and evaluating defenses. Likewise, benchmark datasets such as MNIST [21] (Modified National Institute of Standards and Technology handwritten-digit dataset) continue to serve as fundamental resources for comparative analyses due to their widespread acceptance, ease of use, and established benchmarks across a variety of ML models.

While KANs have begun to attract attention for their interpretability, adaptability, and computational advantages, their resilience to AA attacks remains significantly under-researched. Given the importance of robustness in safety-critical applications, understanding how various KAN architectures perform against established adversarial techniques like FGSM and PGD is crucial. In this extended study, we bridge this critical research gap by systematically evaluating and comparing multiple KAN implementations under FGSM and PGD AAs. By doing so, we aim to identify the strengths and vulnerabilities inherent in these architectures, thereby laying the foundation for future research into targeted defense mechanisms specifically optimized for KAN-based models.

D. Adversarial Training in ML

Adversarial training has emerged as one of the most prominent and effective strategies for improving the robustness of ML models against AAs. Initially introduced by [14] as a defense against the FGSM, adversarial training involves the augmentation of training datasets with adversarially perturbed samples. This augmentation forces the model to encounter and learn from specifically crafted examples during training, thereby facilitating the development of more robust decision boundaries and improving model generalization to unseen adversarial inputs.

Subsequently, [15] significantly enhanced adversarial training by employing PGD as the adversarial example generator. PGD-based adversarial training iteratively applies small perturbations to input data, guiding the model toward learning highly robust and generalizable features. Due to its iterative nature, this method has been established as the state-of-the-art approach for benchmarking robustness in deep learning models. Empirical results consistently confirm that PGD-trained models exhibit significantly improved resilience compared to models trained using traditional or non-adversarial methods.

Building on these seminal studies, [22] proposed the TRADES method, introducing a theoretically-principled framework that explicitly balances the trade-off between adversarial robustness and natural accuracy. The TRADES framework introduces a regularization term that penalizes deviations from robust behavior while maintaining model performance on clean data. This approach has demonstrated notable improvements in robustness compared to standard adversarial training techniques, especially in image classification benchmarks.

Furthermore, [23] proposed integrating feature denoising techniques within adversarial training frameworks, enhancing the resilience of models against AAs by explicitly denoising intermediate feature representations during training. By embedding feature denoising mechanisms directly into adversarial training procedures, their method not only mitigates adversarial perturbations but also reduces the model's vulnerability to natural variations in data. These advancements underscore adversarial training as a continually evolving field, with methods becoming progressively sophisticated to counter increasingly powerful AAs.

However, despite the proven efficacy of adversarial training in enhancing model robustness, it introduces significant computational overhead and complexity [15]. Training models using adversarial techniques typically require extended computational resources and time due to the iterative generation of adversarial examples. Moreover, selecting suitable parameters, such as perturbation magnitude, training composition, and learning rates, becomes critical to achieving optimal performance without compromising model accuracy on clean data. Careful dataset preparation, hyperparameter tuning, and rigorous empirical validation remain essential to leveraging the full benefits of adversarial training methodologies. Addressing these computational challenges and identifying efficient adversarial training strategies tailored to specific NN architectures, including KANs, remain vital areas for ongoing research and development.

E. Adversarial Training Applied to KANs

At the time of the publication of our original paper, the robustness of KAN architectures under adversarial conditions had begun receiving increased attention. Recent studies have expanded on the initial exploration of KAN vulnerabilities, systematically evaluating their performance under various adversarial perturbations and comparing them against traditional NN architectures. For instance, [24] investigated the application of KANs in Wi-Fi-based positioning systems, examining their response to adversarial manipulations in wireless signal inputs. Similarly, [25] assessed robustness aspects of KANs across a range of image classification benchmarks, providing valuable comparative analyses that underscore both strengths and limitations of KAN models in adversarial conditions. Another recent study by [26] evaluated the resilience of KAN architectures to AAs within broader applied ML contexts, highlighting the nuanced sensitivity of spline-based activation functions used within KAN models.

Despite the increasing focus on evaluating KAN robustness, the specific application of adversarial training methodologies to KAN architectures remains notably underexplored. To date, adversarial training has predominantly been applied to well-established models such as CNNs and transformers, whereas its impact on KAN models has yet to be rigorously investigated. Although the inherent flexibility and adaptivity of KANs suggest that adversarial training could significantly enhance their robustness, systematic empirical studies in this area are scarce. Consequently, many aspects remain unexplored, including how different compositions and intensities of adversarially perturbed data influence the training process, as well as the specific interactions between spline-based activation functions and adversarial samples.

Given this substantial gap, there is an important opportunity for research that specifically investigates adversarial training tailored to the unique properties of KAN architectures. Detailed analyses examining the relationship between adversarial perturbation strategies (such as FGSM and PGD) and the adaptability of KAN activation functions could provide essential insights for designing more robust models. Additionally, exploring computationally efficient adversarial training methodologies suitable for the unique structural properties of KANs could further unlock their potential for secure, real-world deployment. Addressing these open questions will be critical for future research, ultimately informing best practices for integrating adversarial training strategies into the design and deployment of KAN models.

III. METHODOLOGY

The primary objective of our methodology is to assess how different KAN architectures respond relative to each other and the baseline MLP classifier under adversarial perturbations, placing emphasis on comparative robustness rather than absolute performance optimization. While we acknowledge that each evaluated model could potentially benefit from further tuning through parameter optimization, architectural adjustments, or advanced regularization methods, we operate under the assumption that the relative effects of AAs will remain consistent regardless of these enhancements.

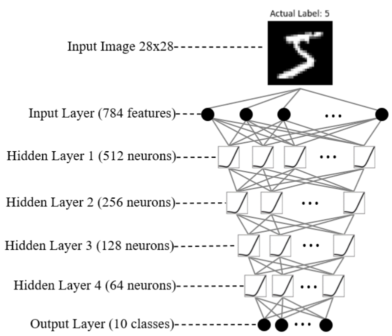


Figure 5. MLP Architecture, source: [7].

This assumption provides a clear foundation for comparing the intrinsic robustness characteristics of various KANs implementation. Nevertheless, future research should rigorously investigate the validity and generalizability of this assumption by exploring the impact of advanced training techniques on robustness outcomes.

The general structure of the KAN networks architecture used in our experiments is illustrated in Figures 5 and 6, which highlight the key differences between traditional MLPs and KAN models. All evaluated KAN models follow this fundamental architectural concept, where traditional node-based activation functions are replaced with edge-based learnable activation functions. The adversarial robustness of four distinct KAN implementations is systematically examined: *Linear (Efficient) KAN* [27], *Naïve Fourier KAN* [28], *Jacobi KAN* [29], and *Chebyshev KAN* [30].

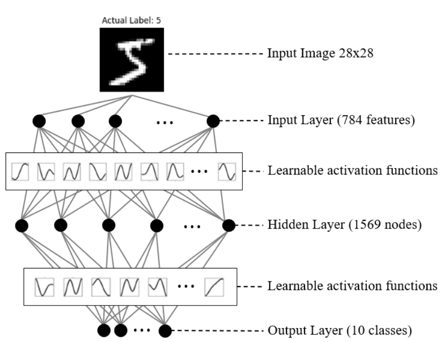


Figure 6. KAN Architecture, source: [7].

The robustness of each model is evaluated under controlled adversarial conditions, involving Gaussian noise as well as two widely recognized AAs technique: the FGSM, and PGD. These adversarial perturbations are generated and administered using the ART [20]. Performance robustness metrics such as accuracy, precision, recall, and F1-scores are utilized to provide a comprehensive understanding of model sensitivity to adversarial perturbations. The well-known MNIST dataset [21], consisting of 33,600 training samples and 8,400 test samples of handwritten digits, provides a standard benchmark that ensures consistency and comparability of results across models.

Furthermore, to extend our previous findings and explore potential improvements in model robustness, we introduce adversarial training by augmenting the original training dataset with adversarially perturbed examples. Specifically, we construct three training sets with varying proportions of clean MNIST samples combined with adversarial samples generated by FGSM, PGD, and Gaussian noise. The training dataset compositions are (i) 85% clean data and 5% of each perturbation type, (ii) 70% clean data and 10% each of noise, FGSM, and PGD, and (iii) 55% clean data and 15% each of noise, FGSM, and PGD. Through this systematic approach, we aim to evaluate how the inclusion of adversarial examples during training influences the robustness and generalizability of different KAN architectures.

In the subsequent sections, detailed results from these experiments will be analyzed, highlighting insights into the relative effectiveness of adversarial training strategies across diverse KAN implementations. Metrics including accuracy, precision, recall, and F1-scores provide a comprehensive understanding of robustness gains and vulnerabilities under adversarial conditions, guiding future research directions toward optimized KAN training strategies.

A. Model Architectures

In this research, we evaluate the robustness of different KANs architecture against AA and compare their performance with a traditional MLP baseline. All architectures use the MNIST dataset [21] and share common parameters for training, such as an AdamW optimizer with a learning rate of 0.001, weight decay for regularization, and an exponential learning rate scheduler to adjust the learning rate dynamically throughout training.

MLP Classifier is utilized as a baseline reference. The model comprises five fully-connected layers, progressively decreasing in size: $784 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 10$. ReLU activation and dropout layers with probability 0.2 are employed after each layer, ensuring model regularization and reduced overfitting. The final output layer utilizes a softmax activation function, providing classification probabilities for each of the ten MNIST digit classes.

The primary models under investigation are four distinct implementations of KANs, each employing unique activation functions along their edges. All KAN implementations share a similar basic architecture, as depicted in Figure 6, but differ substantially in their choice of edge-based activation functions.

Linear KAN (Efficient KAN) [27] utilizes spline-based activation functions parameterized by spline order and grid size. Specifically, the implementation uses spline order 3 and grid size 5, corresponding to a computationally efficient parameterization recommended by the authors. The architecture employs the standard configuration derived from the Kolmogorov-Arnold theorem, where the input dimension of 784 (the MNIST image size of 28×28) is decomposed into one-dimensional spline functions along the network edges. This configuration results in $(28 \times 28) \times 2 + 1 = 1569$ spline parameters, providing the model with substantial flexibility for capturing MNIST data patterns efficiently.

Naïve Fourier KAN [28] modifies the standard spline-based KAN by employing Fourier series coefficients to parameterize the learnable activation functions. Fourier-based activation functions provide smooth and periodic approximations, which inherently bound the activation functions numerically and avoid the common issues associated with spline parameterizations going out of grid bounds. Specifically, the Fourier KAN configuration used in our experiments employs grid size 56, corresponding to twice the dimension of input features, along with initialization parameters that ensure numerical stability and smoothness of learned functions.

Chebyshev KAN (ChebyKAN) [30] substitutes spline functions with Chebyshev polynomials. Chebyshev polynomials, due to their orthogonality and numerical stability, provide efficient approximations suitable for polynomial interpolations over bounded intervals. In our experiments, we employed Chebyshev polynomials of degree 7, aiming to balance approximation accuracy and computational efficiency. ChebyKAN requires fewer parameters to achieve comparable performance relative to spline-based KANs, making it appealing for scenarios where computational resources are constrained.

Jacobi KAN (JacobiKAN) [29], derived from the ChebyKAN framework, uses Jacobi polynomials, a broader family of orthogonal polynomials parameterized by two additional parameters (a, b) controlling polynomial shape. In our experiments, we selected a polynomial degree of 7 with default parameters $a = 0.0$ and $b = 0.0$ - a special case of Jacobi, the Legendre polynomials. This is typically used for MNIST classifications. JacobiKAN provides an adaptive and flexible framework capable of adjusting polynomial forms according to task-specific data characteristics. However, this flexibility introduces additional complexity, requiring careful parameter tuning during training.

All four KAN implementations share a fundamental architectural structure illustrated in Figure 6, differing primarily in the form of their learnable activation functions. By evaluating these architectures systematically, our study seeks to quantify and understand the impact of different parameterizations on model robustness against adversarial perturbations and noise.

B. Attack Architecture

Noise Attack: We conducted Gaussian noise attacks at a noise level of 100 to evaluate the robustness of the models under extreme conditions. This high noise level was deliberately selected to amplify performance degradation, facilitating a clear comparison across the different KAN architectures and the baseline MLP model. Our prior research [7] valuated the noise sensitivity of a single KAN model by incrementally increasing noise levels to determine its robustness relative to the MLP. In the current study, we shift our focus to systematically comparing multiple KAN variants, maintaining the MLP as a consistent baseline for robustness benchmarking. An MNIST digit example (digit '1') corrupted by Gaussian noise at the level of 100 is shown in Figure 7, illustrating the extreme noise conditions used in our robustness assessments.

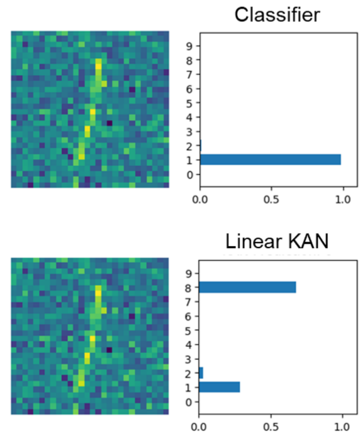


Figure 7. The Gaussian Noise Attack Example.

FGSM Attack: The ART [20] was utilized to generate adversarial examples and implement the FGSM attack across all models. Perturbations were introduced into the MNIST test dataset to create adversarial samples, with the epsilon parameter typically ranging from 0.1 to 0.8. A higher epsilon increases perturbation visibility in images. For this research, an epsilon value of 0.5 was selected, sufficient to significantly degrade model performance without introducing visually noticeable distortions, thus preserving realism in the adversarial scenario. An example of an MNIST digit (digit '1') subjected to the FGSM attack is shown in Figure 8, highlighting how subtle perturbations can drastically alter model predictions.

PGD Attack: We also employed ART [20] to facilitate the PGD AAs. PGD iteratively generates small random perturbations to the input data, progressively maximizing the loss function. Each iteration incrementally adjusts perturbation magnitude, while carefully controlling the maximum perturbation size to maintain imperceptibility to human observers. This iterative approach positions PGD as one of the strongest first-order AAs methods available, significantly more potent than FGSM. Consistent with the FGSM setup, a perturbation level of 0.5 was adopted to simulate realistic adversarial conditions. Figure 9 presents an MNIST digit example (digit '1') after a PGD attack, demonstrating the iterative nature of this strong adversarial perturbation and its effect on model classification.

Tools and environment: All KAN implementations are

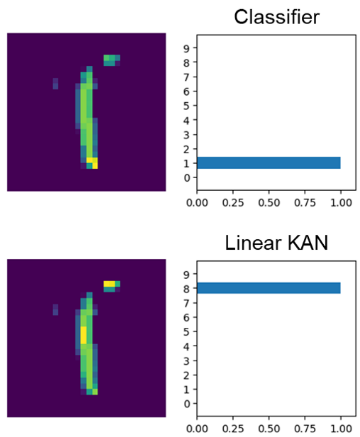


Figure 8. The FGSM Attack Example.

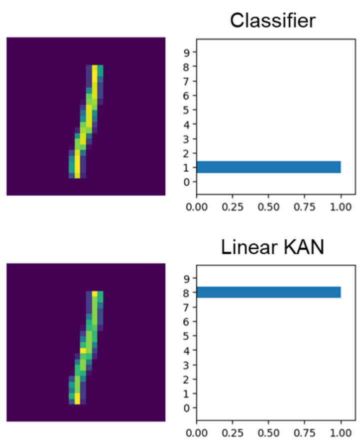


Figure 9. PGD Attack Example.

obtained from publicly available GitHub repositories [28]–[31], alongside ART [20]. The baseline MLP classifier was implemented independently using PyTorch and Scikit-learn Python libraries. The Google Colab cloud environment was utilized to conduct all experiments, ensuring consistency in hardware and software configurations. This standardized experimental environment is intended to facilitate reproducibility and validation of our results.

C. Experiments

All models, including the four KAN architectures: Linear (Efficient) KAN, Naïve Fourier KAN, Jacobi KAN, Chebyshev KAN, and the MLP baseline, were initially trained and evaluated in a controlled, non-adversarial setting using the MNIST dataset [21]. Figure 10 illustrates example digits from the MNIST dataset used in all experiments. Performance metrics, including accuracy, precision, recall, and F1 scores, were recorded for each architecture to establish a robust baseline for subsequent adversarial analyses.

We then assessed each model's robustness under adversarial conditions by subjecting them individually to Gaussian noise, FGSM, and PGD attacks. For each attack scenario, we computed the relative change in performance metrics compared to their baseline values. These results provided insights into the vulnerabilities of each KAN variant relative to the MLP classifier, allowing for a systematic analysis of model-specific weaknesses and strengths under adversarially perturbed conditions.

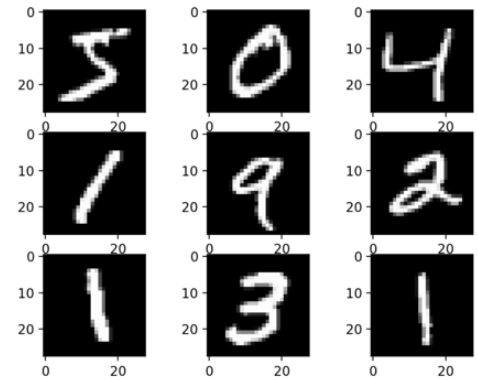


Figure 10. MNIST Dataset Example.

Expanding on this initial analysis, adversarial training experiments were conducted to explore strategies for enhancing model robustness. Specifically, models were retrained with adversarially augmented datasets composed of varying proportions of clean and perturbed data, as follows:

- 85% clean MNIST data combined with 5% each of Gaussian noise, FGSM, and PGD perturbed samples.
- 70% clean MNIST data combined with 10% each of Gaussian noise, FGSM, and PGD perturbed samples.
- 55% clean MNIST data combined with 15% each of Gaussian noise, FGSM, and PGD perturbed samples.

This adversarial training strategy aimed to quantify how incorporating a controlled proportion of adversarially generated data into the training set affects model performance and robustness. Each model was retrained separately under these three training set compositions, and performance metrics were reevaluated on clean as well as adversarially perturbed test sets (noise, FGSM, PGD). The goal was to identify optimal training compositions capable of significantly enhancing robustness without severely compromising accuracy on clean data.

To ensure consistency and reproducibility, all training sessions employed identical hyperparameters, including the AdamW optimizer with a learning rate of 0.001, weight decay for regularization, and an exponential learning rate scheduler. Each training scenario was repeated multiple times to ensure the reliability of observed improvements in robustness metrics.

In the results section that follows, detailed analyses will be presented, comparing performance outcomes from standard training versus adversarial training across all tested models. This comprehensive experimental approach provides critical insights into the efficacy of adversarial training for improving

KAN robustness, contributing valuable guidance for future research into secure and robust NN architectures.

IV. Results

This section presents a comprehensive analysis of our experimental results, structured systematically into five subsections. We begin by establishing baseline performance metrics for all evaluated models in the absence of adversarial conditions. Subsequent subsections report detailed findings on model robustness under Gaussian noise, FGSM, and PGD AAs. Finally, we present a thorough evaluation of the impact of adversarial training on model resilience, comparing performance across varying proportions of adversarially perturbed training data. The analyses provided herein offer valuable insights into the relative strengths and vulnerabilities of different KAN architectures compared to the baseline MLP classifier, highlighting critical considerations for enhancing model robustness.

A. Before Attacks

Initially, we evaluated all models under clean (non-adversarial) conditions using the MNIST dataset, as detailed in Table I. This baseline evaluation provides an essential reference point for assessing subsequent robustness to adversarial perturbations.

TABLE I
ACCURACY BY MODEL. TRAIN SET: 100% MNIST.

Model	Clean	Noise100	FGSM 0.5	PGD 0.5
Classifier	0.98	0.94	0.79	0.66
KAN Linear	0.98	0.86	0.29	0.11
Naïve Fourier	0.92	0.16	0.11	0.22
Jacobi	0.93	0.51	0.08	0.05
Cheby	0.92	0.39	0.05	0.04

Accuracy results before AAs are visualized in Figure 11, clearly indicating that the MLP Classifier and the Linear KAN both achieve nearly identical accuracy (98%), establishing a strong performance baseline. Conversely, the other three KAN variants: Naïve Fourier, Jacobi, and Chebyshev exhibit somewhat lower accuracy scores (92-93%). Although the primary objective of this study focuses on evaluating relative robustness under adversarial conditions rather than absolute accuracy, these performance discrepancies warrant further exploration. Future research may investigate whether model-specific architectural differences, parameter settings, or alternative optimization strategies might account for these performance gaps and potentially improve the absolute accuracy of the affected KAN architectures.

Another notable observation relates to computational complexity and training duration. Despite improvements from the use of Google Colab's free-tier T4 GPU, training times for KAN models remained substantially longer compared to the simpler MLP architecture. Specifically, KAN architectures typically required roughly ten times longer to train than the baseline MLP. This discrepancy, attributed primarily to the computational overhead associated with spline-based and polynomial-based activation functions, highlights a significant practical consideration for real-world deployment and iterative training workflows.

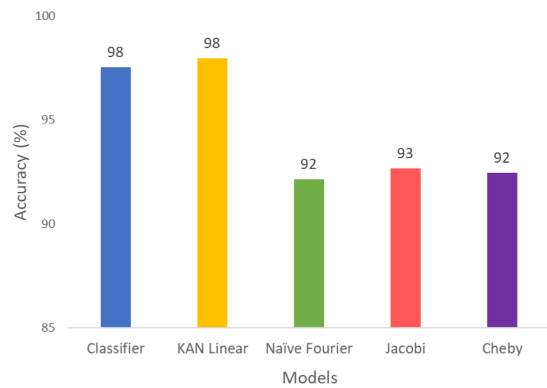


Figure 11. Model Accuracy Comparison Before Attacks.

An additional intriguing finding involves the class-wise balance of model performance, particularly evident from examining the F1 scores for individual digit classes, as illustrated in Figure 12. The Linear KAN model demonstrates generally balanced F1 scores across most digit classes but exhibits a pronounced drop in performance on digit 9. Other KAN models similarly reflect class imbalance patterns, suggesting inherent limitations or biases within their activation function parameterizations, and suggesting that certain activation functions or training methodologies may disproportionately impact specific digit classes. Investigating the causes of these class-specific discrepancies may offer valuable insights into further optimizing KAN architectures or identifying data-specific challenges. Such analyses remain outside the scope of this current study but represent promising avenues for future research.

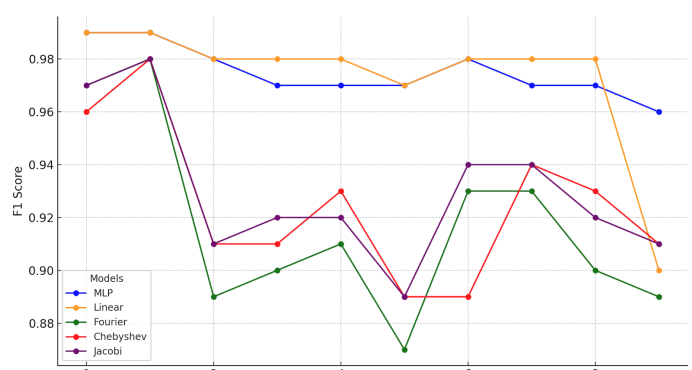


Figure 12. Model F1 Score Comparison Before Attacks, source: [1].

Overall, these baseline performance evaluations provide essential context for subsequent analyses of model robustness under various adversarial perturbations, enabling precise quantification of robustness changes attributed explicitly to adversarial conditions.

B. Gaussian Noise Attack Results

All evaluated models exhibited reduced accuracy when exposed to Gaussian noise at the extreme level of 100, as detailed in Table I. Figure 13 clearly illustrates the drop in accuracy for each model before and after the noise attack.

The MLP Classifier demonstrated robust performance, maintaining high accuracy at 94%, reflecting only a modest reduction of approximately 4%. The Linear KAN model also performed relatively well under noisy conditions, achieving an accuracy of 86%, though this still represents a notable accuracy drop of about 12%. In contrast, the other evaluated KAN architectures: Naïve Fourier, Jacobi, and Chebyshev, experienced severe degradation in performance, with accuracy declining dramatically to 16%, 51%, and 39%, respectively.

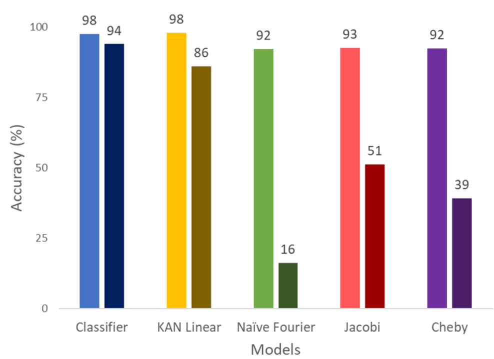


Figure 13. Model Accuracy Comparison After Noise Attack.

Figure 14 provides a visualization of the percentage accuracy losses, further underscoring the pronounced vulnerability of polynomial-based KAN models to Gaussian noise attacks. While Linear KAN demonstrates comparatively moderate sensitivity to noise, its accuracy loss is still substantially higher than the baseline MLP, suggesting inherent architectural vulnerabilities of KAN models under noisy conditions. These observations emphasize the necessity of further investigation into mechanisms underlying KAN models' sensitivity to noise, guiding future enhancements in model robustness.

TABLE II
ACCURACY REDUCTION, (%).

Model	Noise100	FGSM 0.5	PGD 0.5
Classifier	4	18	31
KAN Linear	12	69	87
Naïve Fourier	76	81	70
Jacobi	41	84	88
Cheby	53	88	88

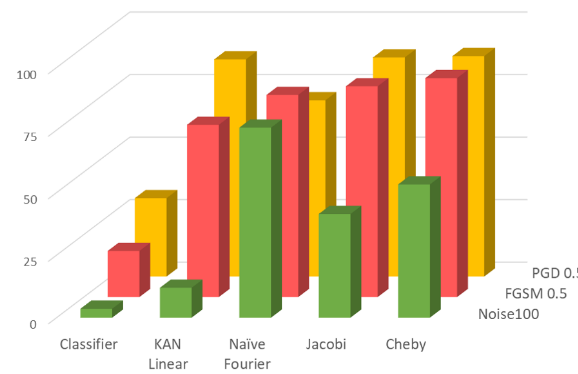


Figure 14. Accuracy Loss Comparison by Attack.

C. FGSM Attack Results

Under the FGSM attack with a perturbation parameter ($\epsilon = 0.5$), all evaluated models experienced significantly greater accuracy losses compared to the Gaussian noise attack. Figure 15 clearly illustrates this reduction in accuracy scores for each model when subjected to FGSM-generated adversarial examples.

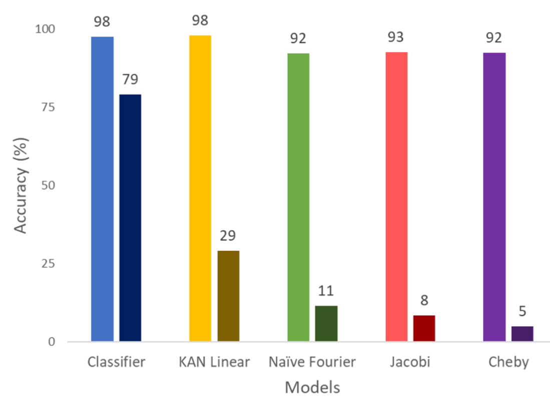


Figure 15. Model Accuracy Comparison After FGSM Attack.

Consistent with previous attack outcomes, the baseline MLP classifier demonstrated the strongest resilience among all models, yet it still experienced a substantial accuracy drop from 98% to 79%, representing a relative accuracy loss of approximately 18%. The Linear KAN model, while maintaining higher absolute accuracy compared to other KAN architectures, displayed a considerable accuracy reduction, falling from 98% to just 29%. Notably, this corresponds to a relative loss of approximately 69% in accuracy, highlighting Linear KAN's vulnerability to the FGSM attack.

Interestingly, the Naïve Fourier KAN model, which exhibited poor performance under Gaussian noise conditions, showed a relatively stronger resilience compared to other polynomial-based KAN variants under FGSM perturbations, achieving an accuracy of 11%. While still significantly affected, this result contrasts sharply with its extreme vulnerability under noise attacks. Jacobi and Chebyshev KAN models suffered the most severe accuracy losses, dropping from initial accuracies around 92-93% to below 10% accuracy post-FGSM attack, underscoring their heightened sensitivity to adversarially generated perturbations.

The relative accuracy losses across models under different attack conditions are summarized in Table II and visually depicted in Figure 14. This comprehensive visualization emphasizes the particularly devastating impact of the FGSM attack on the polynomial-based KAN models.

An intriguing observation from these results is the apparent inverse performance relationship between the polynomial-based KAN models' responses to Gaussian noise and FGSM attacks. This phenomenon, visually apparent in the comparison of Figures 14 and 15, suggests distinct underlying vulnerabilities to different perturbation types. This finding provides a compelling direction for future research, potentially exploring the underlying mechanisms driving these divergent responses, and informing more targeted strategies for robustness enhancement.

D. PGD Attack Results

Under the PGD attack at an intensity level of 0.5, all tested models suffered severe accuracy degradation. Figure 16 illustrates a significant decline in accuracy for each model when subjected to the PGD adversarial perturbations.

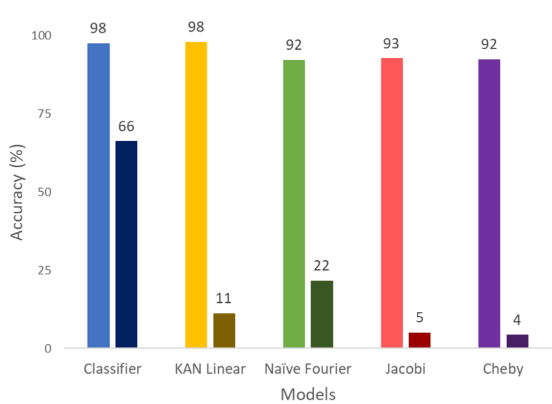


Figure 16. Model Accuracy Comparison After PGD Attack.

Interestingly, while the MLP classifier demonstrated the highest absolute accuracy (66%) following the PGD attack, it experienced a considerable relative accuracy loss of about 31%, highlighting significant vulnerability despite its robustness under other conditions. The Linear KAN, which performed well under noise attacks, showed an exceptionally high sensitivity to PGD attacks, with accuracy plunging drastically to 11%, reflecting an 87% relative loss.

On the contrary, other KAN architectures, particularly Naïve Fourier KANshowed slightly better resilience in relative terms compared to Linear KAN, albeit their absolute accuracy scores remained critically low (22%). Notably, Chebyshev and Jacobi KAN displayed minimal accuracy retention at around 5%.

It is important to emphasize that the overall degradation was catastrophic for all models. The accuracy for the majority of digit classes dropped dramatically to zero or near-zero for all models, indicating extensive vulnerability across all tested NN variants.

Nevertheless, from a relative accuracy retention standpoint, Naïve Fourier showed marginally better resilience than other KAN variants, making it the least affected architecture under the PGD attack scenario. This nuanced difference, although minor in absolute terms, presents an intriguing avenue for future investigation into what specific properties of Naïve Fourier activation functions might contribute to improved resilience against iterative adversarial perturbations like PGD.

These observations are visually summarized in Figure 16, clearly highlighting the extensive accuracy loss across all models, reinforcing the potent effectiveness of PGD attacks on current NNs architectures.

E. Adversarial Training Results

To further examine the robustness of our models, we conducted adversarial training experiments by progressively reducing the proportion of clean MNIST data in the training set (85%, 70%, and 55%) and simultaneously increasing the adversarially perturbed examples. Tables III, IV, and V summarize the performance of each model under these conditions.

TABLE III
ACCURACY BY MODEL. TRAIN SET: 85% MNIST.

Model	Clean	Noise100	FGSM 0.5	PGD 0.5
Classifier	0.97	0.93	0.96	0.88
KAN Linear	0.98	0.80	0.98	0.99
Naïve Fourier	0.92	0.18	0.97	0.98
Jacobi	0.92	0.82	0.69	0.74
Cheby	0.92	0.50	0.88	0.90

TABLE IV
ACCURACY BY MODEL. TRAIN SET: 70% MNIST.

Model	Clean	Noise100	FGSM 0.5	PGD 0.5
Classifier	0.97	0.92	0.96	0.89
KAN Linear	0.97	0.76	0.98	0.98
Naïve Fourier	0.92	0.18	0.94	0.97
Jacobi	0.92	0.80	0.66	0.72
Cheby	0.91	0.54	0.91	0.93

Our adversarial training experiments revealed significant robustness gains across all evaluated models, demonstrating substantial resilience improvements against FGSM and PGD attacks, even when training data contained high proportions

TABLE V
ACCURACY BY MODEL. TRAIN SET: 55% MNIST.

Model	Clean	Noise100	FGSM 0.5	PGD 0.5
Classifier	0.97	0.92	0.96	0.89
KAN Linear	0.97	0.71	0.94	0.96
Naïve Fourier	0.92	0.19	0.95	0.98
Jacobi	0.91	0.80	0.65	0.71
Cheby	0.91	0.60	0.92	0.96

of adversarial samples. Linear KAN exhibited remarkable improvement, achieving 98% accuracy under FGSM and 99% accuracy under PGD with 85% clean data. Even at the lowest clean data level (55%), Linear KAN maintained 94% and 96% accuracy for FGSM and PGD respectively, though accuracy dropped significantly to 71% under high-level noise attacks. Figure 17 visually highlights the robustness improvement across adversarial scenarios, and Table VI provides detailed information.

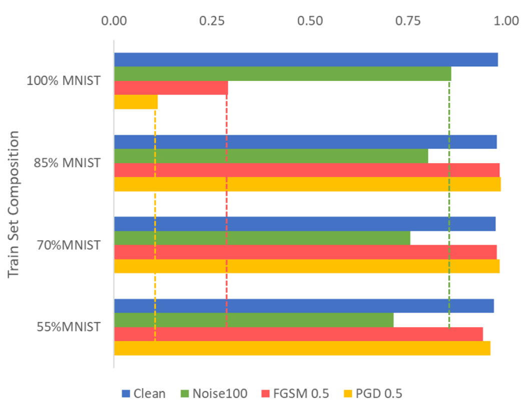


Figure 17. Model Accuracy by Train set Linear KAN.

The Naïve Fourier KAN demonstrated a dramatic transformation, jumping from poor performance (11% FGSM and 22% PGD accuracy at 100% MNIST clean data) to 97% and 98% accuracy respectively at 85% clean data. Even further reduction to 55% clean data sustained high performance, achieving 95% for FGSM and 98% for PGD (Figure 18, Table VII). However, Naïve Fourier continued to underperform in noise attacks across all data compositions, never exceeding 20% accuracy.

Jacobi and Cheby KAN models also improved significantly, albeit with more moderate gains. Jacobi KAN, which initially had catastrophic performance under FGSM (8%) and PGD (5%) at 100% MNIST, improved considerably to 69% and 74% respectively at 85% MNIST clean data. However, further reduction in clean data slightly diminished robustness, settling at 65% FGSM and 71% PGD at 55% MNIST (Figure 19, Table VIII).

TABLE VI
PERFORMANCE METRICS BY ATTACK. KAN LINEAR.

	TestSet	Precision	Recall	F1-score	Accuracy
100% MNIST	Clean	0.98	0.98	0.98	0.98
	Noise	0.90	0.86	0.86	0.86
	FGSM	0.50	0.29	0.30	0.29
	PGD	0.20	0.11	0.07	0.11
85% MNIST	Clean	0.98	0.98	0.98	0.98
	Noise	0.86	0.80	0.81	0.80
	FGSM	0.98	0.98	0.98	0.98
	PGD	0.99	0.99	0.99	0.99
70% MNIST	Clean	0.97	0.97	0.97	0.97
	Noise	0.83	0.76	0.75	0.76
	FGSM	0.98	0.98	0.98	0.98
	PGD	0.99	0.98	0.98	0.98
55% MNIST	Clean	0.97	0.97	0.97	0.97
	Noise	0.82	0.71	0.71	0.71
	FGSM	0.94	0.94	0.94	0.94
	PGD	0.96	0.96	0.96	0.96

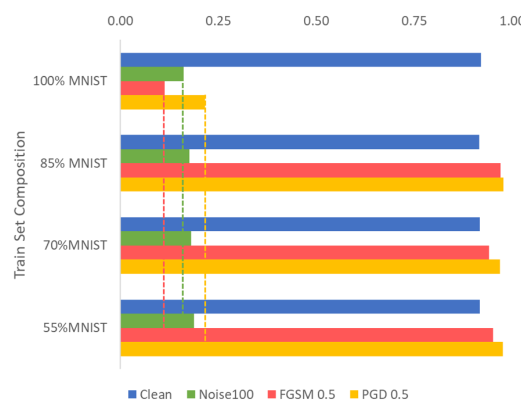


Figure 18. Model Accuracy by Train set Naive Fourier.

Cheby KAN showed an impressive recovery from initial single-digit accuracy figures to consistently high performances (88% FGSM and 90% PGD at 85% MNIST), improving further as clean data proportion decreased, reaching 92% FGSM and 96% PGD at 55% MNIST (Figure 20, Table IX).

The MLP classifier displayed robust and consistent improvement, maintaining high performance with minor fluctuations. With 85% clean data, the MLP reached 96% FGSM and 88% PGD accuracy, and notably, further reductions of clean data to 55% sustained performance, yielding 96% FGSM and 89% PGD accuracy (Figure 21, Table X).

TABLE VII
PERFORMANCE METRICS BY ATTACK. NAÏVE FOURIER.

	TestSet	Precision	Recall	F1-score	Accuracy
100% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.17	0.16	0.16	0.16
	FGSM	0.22	0.11	0.07	0.11
	PGD	0.53	0.22	0.20	0.22
85% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.19	0.18	0.17	0.18
	FGSM	0.97	0.97	0.97	0.97
	PGD	0.98	0.98	0.98	0.98
70% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.20	0.18	0.17	0.18
	FGSM	0.94	0.94	0.94	0.94
	PGD	0.97	0.97	0.97	0.97
55% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.20	0.19	0.19	0.19
	FGSM	0.95	0.95	0.95	0.95
	PGD	0.98	0.98	0.98	0.98

TABLE VIII
PERFORMANCE METRICS BY ATTACK. JACOBI.

	TestSet	Precision	Recall	F1-score	Accuracy
100% MNIST	Clean	0.93	0.93	0.93	0.93
	Noise	0.68	0.51	0.52	0.51
	FGSM	0.07	0.08	0.05	0.08
	PGD	0.05	0.05	0.02	0.05
85% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.83	0.82	0.82	0.82
	FGSM	0.69	0.69	0.69	0.69
	PGD	0.74	0.74	0.73	0.74
70% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.82	0.80	0.80	0.80
	FGSM	0.66	0.66	0.65	0.66
	PGD	0.72	0.72	0.72	0.72
55% MNIST	Clean	0.91	0.91	0.91	0.91
	Noise	0.82	0.80	0.80	0.80
	FGSM	0.66	0.65	0.65	0.65
	PGD	0.71	0.71	0.70	0.71

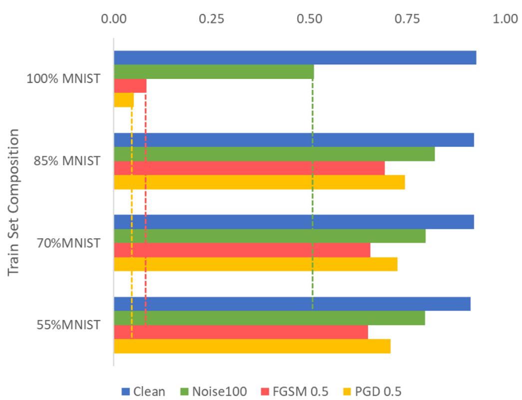


Figure 19. Model Accuracy by Train set Jacobi.

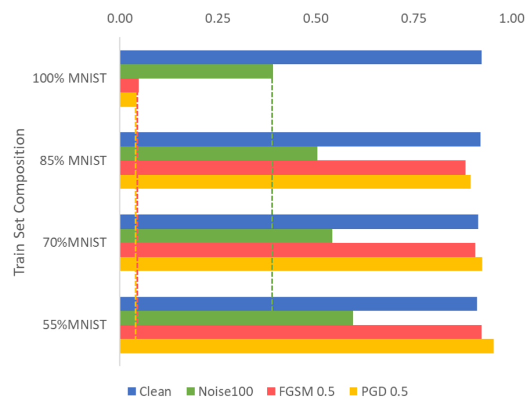


Figure 20. Model Accuracy by Train set Cheby.

V. Evaluation and Discussion

Our experiments show that the same spline flexibility that gives KANs their predictive power also makes them prone to overfitting. Under adversarial perturbations, KANs can lose far more accuracy than a standard MLP, creating a serious risk in security-sensitive contexts.

Adversarial training is an effective countermeasure. Injecting a small fraction (5%) of adversarial samples per AA type into the training set restores robustness across all variants, improving accuracy by more than 60 points under strong PGD attacks. Although generating those perturbed samples adds computational cost, the resulting resilience gains justify this overhead for any real-world KAN deployments.

Not all KANs respond equally. Linear and orthogonal-spline (Jacobi, Chebyshev) variants recover most of their robustness with modest adversarial mixing. The Fourier KAN, while nearly perfect under adversarial attacks after training, remains highly vulnerable to Gaussian noise. Its noise accuracy never exceeds 20 % even at high perturbation ratios. This tells us that adversarial training alone cannot address stochastic-noise weaknesses; techniques such as input denoising is required. The Jacobi KAN shows the smallest net gain overall and may benefit from hybrid hardening tailored to its spline structure.

The next section outlines future directions, including systematic tuning of adversarial and noise ratios for each KAN type, theoretical analysis of spline susceptibility, and more efficient adversarial-sample generation methods.

VI. Conclusion and Future Work

In this work, we have: 1) Quantified the vulnerability of four KAN architectures, revealing up to 88% under adversarial attacks (and up to 76% under noise conditions). 2) Shown that modest adversarial training (5% perturbed samples per

TABLE IX
PERFORMANCE METRICS BY ATTACK. CHEBY.

	TestSet	Precision	Recall	F1-score	Accuracy
100% MNIST	Clean	0.93	0.92	0.92	0.92
	Noise	0.56	0.39	0.38	0.39
	FGSM	0.08	0.05	0.03	0.05
	PGD	0.01	0.04	0.01	0.04
85% MNIST	Clean	0.92	0.92	0.92	0.92
	Noise	0.65	0.50	0.51	0.50
	FGSM	0.88	0.88	0.88	0.88
	PGD	0.91	0.90	0.89	0.90
70% MNIST	Clean	0.92	0.91	0.91	0.91
	Noise	0.68	0.54	0.55	0.54
	FGSM	0.91	0.91	0.91	0.91
	PGD	0.93	0.93	0.92	0.93
55% MNIST	Clean	0.91	0.91	0.91	0.91
	Noise	0.70	0.60	0.61	0.60
	FGSM	0.92	0.92	0.92	0.92
	PGD	0.96	0.96	0.95	0.96

TABLE X
PERFORMANCE METRICS BY ATTACK. CLASSIFIER.

	TestSet	Precision	Recall	F1-score	Accuracy
100% MNIST	Clean	0.98	0.98	0.98	0.98
	Noise	0.94	0.94	0.94	0.94
	FGSM	0.79	0.79	0.79	0.79
	PGD	0.66	0.66	0.65	0.66
85% MNIST	Clean	0.97	0.97	0.97	0.97
	Noise	0.93	0.93	0.93	0.93
	FGSM	0.96	0.96	0.96	0.96
	PGD	0.89	0.88	0.88	0.88
70% MNIST	Clean	0.97	0.97	0.97	0.97
	Noise	0.92	0.92	0.92	0.92
	FGSM	0.96	0.96	0.96	0.96
	PGD	0.89	0.89	0.89	0.89
55% MNIST	Clean	0.97	0.97	0.97	0.97
	Noise	0.92	0.92	0.92	0.92
	FGSM	0.96	0.96	0.96	0.96
	PGD	0.89	0.89	0.89	0.89

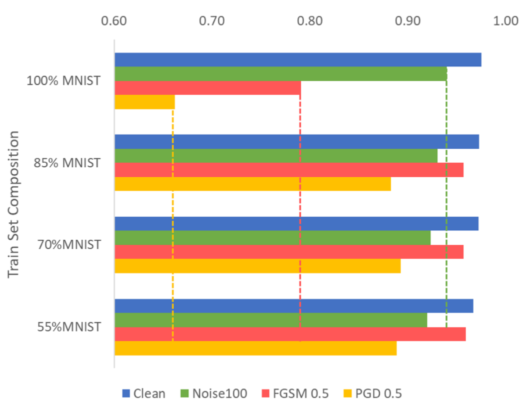


Figure 21. Model Accuracy by Train set Classifier.

AA type) recovers more than 60 points of robustness. 3) Identified that Fourier KANs remain noise-sensitive even after adversarial training, highlighting the need for future research and hybrid noise–adversarial defenses.

Our results demonstrate significant variation among KAN models in their response to AA and provide a comparative analysis against traditional MLP classifiers. Consistent with previous findings [1], [7], in the unprotected setting, the MLP baseline outperforms all KAN variants under FGSM and PGD attacks. However, after adversarial training, Linear and Fourier KANs exceed the MLP's robustness, reaching nearly 99% accuracy against PGD, highlighting the effectiveness of targeted hardening for these architectures.

One critical observation in our study was the significant imbalance across classes within KAN models, particularly evident in Figure 12. Further investigation into the underlying causes of this imbalance could provide valuable insights into improving the robustness and general performance of KANs. Understanding these mechanisms might not only enhance our theoretical understanding of KAN architectures but also guide practical improvements for diverse applications.

Adversarial training experiments provided substantial new insights. Introducing progressively greater proportions of adversarial data into the training sets notably improved resilience across all models. This approach significantly enhanced KAN models' robustness, especially Linear and Naïve Fourier KANs, which achieved near-perfect accuracy (98%–99%) under both FGSM and PGD attacks with 85% clean data. Remarkably, even reducing clean training data to as low as 55%, these models maintained high accuracy (above 94%), demonstrating their considerable potential for adversarial robustness. In contrast, Jacobi and Cheby KANs showed substantial, though less pronounced, improvements, indicating that different activation functions significantly influence adversarial training outcomes.

Our study did not specifically address training efficiency, but the substantial training time observed for KAN models highlights a potential area for future research. Understanding and optimizing the trade-off between training efficiency and adversarial robustness, especially for novel architectures like KANs, is critical for broader adoption and practical applications.

Future Research Directions

Building on our results, we identify several promising areas for future investigation:

- Deepening theoretical understanding of why certain KAN models (e.g., Fourier) exhibit greater resistance to PGD attacks, potentially guiding new architectural designs or activation function choices.

- Developing specialized adversarial robustness training strategies tailored explicitly for different KAN architectures to further leverage their inherent strengths.
- Exploring additional AA methodologies and evaluating KAN robustness on more diverse datasets. Future work should rigorously test KAN robustness using datasets beyond MNIST, such as CIFAR-10 or ImageNet, to validate the generalizability of our findings and their practical implications.
- Investigating and addressing the observed class imbalance issue within KAN models to improve both robustness and general classification performance.
- Assessing the balance between computational efficiency, training time, and model robustness to enhance the practical deployment of KAN models in real-world applications.

Pursuing these identified research directions will significantly deepen our theoretical understanding of KAN robustness, fostering advancements toward practically deployable, secure, and interpretable ML models.

Acknowledgment

We acknowledge the use of various general-purpose online and cloud-based tools, including those with AI-driven features, during the preparation of this work.

References

[1] N. Djosic, E. Ostanin, F. Hussain, S. Sharieh, and A. Ferworn, "KAN vs KAN: Examining Kolmogorov-Arnold networks (KAN) performance under adversarial attacks", in Proceedings of the SECURWARE 2024, The Eighteenth International Conference on Emerging Security Information, Systems and Technologies, Nov. 2024, pp. 17–22.

[2] Z. Liu et al., "KAN: Kolmogorov-Arnold networks", Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.19756. Accessed: 14 May 2025.

[3] C. Zeng, J. Wang, H. Shen, and Q. Wang, "KAN versus MLP on irregular or noisy functions", 2024, [Online]. Available: https://arxiv.org/abs/2408.07906. Accessed: 14 May 2025.

[4] H. Shen, C. Zeng, J. Wang, and Q. Wang, "Reduced effective-ness of Kolmogorov-Arnold networks on functions with noise", Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.14882. Accessed: 14 May 2025.

[5] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: A comprehensive survey", Journal of Defense Modeling and Simulation, vol. 19, pp. 57–106, Jan. 2022.

[6] B. Xi, "Adversarial machine learning for cybersecurity and computer vision: Current developments and challenges", Wiley Interdisciplinary Reviews: Computational Statistics, vol. 12, p. 1511, Sep. 2020.

[7] E. Ostanin, N. Djosic, F. Hussain, S. Sharieh, and A. Ferworn, "Evaluating the robustness of Kolmogorov-Arnold networks against noise and adversarial attacks", in Proceedings of the SECURWARE 2024, The Eighteenth International Conference on Emerging Security Information, Systems and Technologies, Nov. 2024, pp. 11–16.

[8] J. Xu et al., "Fourierkan-gcf: Fourier Kolmogorov-Arnold network – an effective and efficient feature transformation for graph collaborative filtering", 2024, [Online]. Available: https://arxiv.org/abs/2406.01034. Accessed: 14 May 2025.

[9] S. Sidharth, A. Keerthana, R. Gokul, and K. Anas, "Chebyshev polynomial-based Kolmogorov-Arnold networks: An efficient architecture for nonlinear function approximation", 2024, [Online]. Available: https://arxiv.org/abs/2405.07200. Accessed: 14 May 2025.

[10] M. Cheon, "Demonstrating the efficacy of Kolmogorov-Arnold networks in vision tasks a preprint", 2024, [Online]. Available: https://arxiv.org/abs/2406.14916. Accessed: 14 May 2025.

[11] B. Azam and N. Akhtar, "Suitability of KANs for computer vision: A preliminary investigation", Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.09087. Accessed: 14 May 2025.

[12] V. D. Tran et al., "Exploring the limitations of Kolmogorov-Arnold networks in classification: Insights to software training and hardware implementation", Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.17790. Accessed: 14 May 2025.

[13] G. R. Machado, E. Silva, and R. R. Goldschmidt, "Adversarial machine learning in image classification: A survey towards the defender's perspective", Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.03728. Accessed: 14 May 2025.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples", Mar. 2015, [Online]. Available: https://arxiv.org/abs/1412.6572. Accessed: 14 May 2025.

[15] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks", Sep. 2019, [Online]. Available: http://arxiv.org/abs/1706.06083. Accessed: 14 May 2025.

[16] W. Villegas, A. Jaramillo-Alcázar, and S. Luján-Mora, "Evaluating the robustness of deep learning models against adversarial attacks: An analysis with FGSM, PGD and CW", Big Data and Cognitive Computing, vol. 8, p. 8, Jan. 2024.

[17] Y. Jang, T. Zhao, S. Hong, and H. Lee, "Adversarial defense via learning to generate diverse attacks", In In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2740–2749.

[18] G. Sriramanan, S. Addepalli, A. Baburaj, and R. V. Babu, "Guided adversarial attack for evaluating and enhancing adversarial defenses", in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 20 297–20 308.

[19] S. Mohandas, N. Manwani, and D. P. Dhulipudi, "Momentum iterative gradient sign method outperforms PGD attacks", in International Conference on Agents and Artificial Intelligence, vol. 3, Science and Technology Publications, Lda, 2022, pp. 913–916.

[20] M.- I. Nicolae et al., "Adversarial robustness toolbox v1.0.0", 2019, [Online]. Available: https://arxiv.org/abs/1807.01069. Accessed: May 2025.

[21] L. Deng, "The MNIST database of handwritten digit images for machine learning research", IEEE Signal Processing Magazine, vol. 29, pp. 141–142, Jun. 2012.

[22] H. Zhang et al., "Theoretically principled trade-off between robustness and accuracy", in International conference on machine learning, PMLR, 2019, pp. 7472–7482.

[23] C. Xie, Y. Wu, L. van der Maaten, A. Yuille, and K. He, "Feature denoising for improving adversarial robustness", 2019.

[24] M. Goswami, R. Chatterjee, S. Mahato, and P. K. Pattnaik, "Adversarial-ensemble Kolmogorov Arnold networks for enhancing indoor wi-fi positioning: A defensive approach against spoofing and signal manipulation attacks", 2025.

[25] T. Alter, R. Lapid, and M. Sipper, "On the robustness of Kolmogorov-Arnold networks: An adversarial perspective", 2024.

[26] A. D. M. Ibrahum, Z. Shang, and J.- E. Hong, "How resilient are Kolmogorov–Arnold networks in classification tasks? A robustness investigation", Applied Sciences, vol. 14, no. 22, 2024.

[27] H. Cao, "An efficient implementation of Kolmogorov-Arnold network (KAN)", 2024, [Online]. Available: https://github.com/Blealtan/efficient-kan. Accessed: 14 May 2025.

[28] G. Noesis, "Pytorch layer for FourierKAN", 2024, [Online]. Available: https://github.com/GistNoesis/FourierKAN/tree/main. Accessed: 14 May 2025.

[29] SpaceLearner, "Jacobi polynomials KAN", 2024, [Online]. Available: https://github.com/SpaceLearner/JacobiKAN. Accessed: 14 May 2025.

[30] SynodicMonth, "Chebyshev polynomials KAN", 2024, [Online]. Available: https://github.com/SynodicMonth/ChebyKAN/. Accessed: 14 May 2025.

[31] Z. Liu, "Python Kolmogorov-Arnold networks (KANs)", 2024, [Online]. Available: https://github.com/KindXiaoming/pykan. Accessed: 14 May 2025.

CorpIA: An Operational Framework for AI Agents Augmenting Knowledge Work

Salvatore Vella
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: sal.vella@torontomu.ca

Moussa Noun
Royal Bank of Canada
Toronto, Ontario
e-mail: moussa.noun@rbc.com

Salah Sharieh
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: salah.sharieh@torontomu.ca

Fatima Hussain
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: fatima.hussain@torontomu.ca

Alex Ferworn
Department of Computer Science
Toronto Metropolitan University
Toronto, Ontario
e-mail: aferworn@torontomu.ca

***Abstract*—We present a Generative Artificial Intelligence (AI) based cognitive architecture and an agent specifically developed for the complexities of knowledge workers, such as Cybersecurity analysts. White-collar roles, exemplified by Cybersecurity analysts, are multifaceted and rely on declarative knowledge, procedural understanding, and diverse tools. The ability to learn and adapt to the nuances of the job is crucial. This paper introduces CorpIA, a cognitive architecture that provides an agent with knowledge, tools, and the capacity to acquire on-the-job experience. This system enhances human performance by providing suggested solutions and continuous mentoring. CorpIA includes a programming language for AI agents, ContentCreate, that allows non-programmers to create workflows involving AI agents. Our research demonstrates that the CorpIA agent can learn from interactions using Bloom's Taxonomy. We provide the source code for these experiments.**

Keywords-AI Agents; Cybersecurity; Automation.

I. Introduction

This is a follow-up article to a paper presented at the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2024). Our initial research into augmenting Cybersecurity analysts with AI agents was presented in [1], where we introduced the concept of using Bloom's Taxonomy [2] to measure the learning of an AI agent. This paper extends that work by fully outlining the cognitive architecture and framework used to perform the study, CorpIA, and introduces a programming language, ContentCreate, to coordinate workflows amongst AI agents.

Digital systems and the Internet are critical to our everyday lives. Cyber threats from bad actors require robust Cybersecurity measures.

Cybersecurity analysts are prototypical white-collar professionals who rely on large amounts of knowledge and data and use their experience and skills to collaborate in the workplace. Moreover, as security threats, methods, tactics, techniques, and tools evolve, there is lifelong learning.

The challenges for Cybersecurity analysts are numerous. There are skill requirements to be proficient in many tools and technologies, which also change over time. There are challenges to ongoing learning with emerging threats.

There is a need for advanced Artificial Intelligence (AI) support for Cybersecurity analysts. We have identified the need for Generative AI solutions specifically tailored for these professions [3], [4].

Artificial Intelligence has a long history and can be traced back to Alan Turing in the 1930s and his Turing Machine [5], an abstract machine that could implement any computer algorithm. Generative models also have a long history, with models of generative data sequences, such as speech and time series, available since the 1950s [6].

A recent breakthrough in the field was the introduction of the transformer model architecture [7] in 2017. The transformer is the architecture of many state-of-the-art models, including GPT-3 [8]. These have generally scaled in performance with the number of parameters. Advancements in hardware, specifically GPUs, have enabled the training of huge models, and the cloud has allowed these resources to be available to anyone with an internet connection.

Large Language Models (LLMs) are Generative AI models that implement transformer models to generate text and other content. They can automate tasks previously done by humans [9]. Since ChatGPT became available, many knowledge professionals have been using these tools [10]. These evolved into more general frameworks such as ChatDev [11] and Autogen [12], allowing users to create multiple autonomous agents which can run through workflows. ChatDev specializes in software development roles, and Autogen provides for the creation of more general roles.

These frameworks are evolving quickly, according to Cheng et al. [13] all of these frameworks are headed to (a) autonomy, where the agents independently perceive, make decisions, and take actions; (b) perception to allow them to gather information; (c) decision making; (d) actions that alter the state of the environment.

Our proposed approach is described next.

1) Use the CorpIA architecture to create a Cybersecurity Analyst AI agent and show that the agent can use declarative and procedural knowledge and learn and apply additional information from the chat.
2) Apply Bloom's Taxonomy [2] to measure the AI agent's levels of understanding and application of that knowledge.
3) Explore using Human AI collaboration to design systems that mentor professionals.

In continuation of our proposed approach, the following are our contributions in this paper:

1) Introduction of the CorpIA architecture for creating AI agents for knowledge workers. This novel architecture simplifies the creation of a knowledge worker agent. We demonstrate several knowledge worker agents developed in the accompanying GitHub repository.
2) Enhancement of Human Performance. We demonstrate how AI agents can help human professionals in complex tasks.
3) On The Job Learning of AI Agents. We show how AI agents can learn from interactions. We show these agents can progress through Bloom's taxonomy in practical scenarios.
4) Introduction of the ContentCreate language allows for programming AI agent workflows. This simple language can be easily used by non-programmers, allowing any knowledge worker to create complex workflows.
5) Source Code. We offer the CorpIA source code for replication, validation and further development.

Starting with the introduction in Section I, the rest of the paper is organized in this manner. A literature review is presented in Section II, followed by Bloom's Taxonomy in Section III and the CorpIA architecture in Section IV. Section V discusses the methodology; Section VI discusses the Results. We discuss the results in Section VII and ethical considerations in Section VIII. The conclusion is drawn in Section IX.

II. Literature Review

In this section, we will review various topics discussed in this paper.

A. Digital Labour

Digital labour represents an emergent form characterized by value production through interaction with information and communication technologies such as digital platforms or artificial intelligence [14]. With the emergence of Generative AI agents comes the possibility of augmentation agents acting as assistants for knowledge professionals.

We can emulate the best professionals in the field. For example, the best Cybersecurity analyst agent with the best knowledge acts with the most successful experiences and presents the best personality for the specific client.

Work on enhancing human intellect has also evolved. Engelbart [15] is one of the most influential and prolific inventors of devices we use today. He focused mainly on physical aids to augment humans. We have now evolved to digital aids to augment professionals. Vella and Sharieh [16] have introduced a framework that defines knowledge workers as a set of knowledge, experience and skills.

B. Autonomous Agent Frameworks

Building on simple graphical tools such as OpenAI's ChatGPT [17], autonomous agent frameworks have been built using the underlying APIs. Autogen [12] is an example of such a framework that allows for the definition of agents and workflows between those AI agents.

There are many such agent frameworks and some excellent summaries of their construction. Two good sources are Cheng et al. [13] and Wang et al. [18]. These frameworks allow for the definition and creation of agents to perform tasks and interactions. They include memory, tools, and a workflow engine.

There is a problem today with programming directly to one of the many frameworks that are evolving. There are often incompatibilities and deprecations of interfaces as these frameworks evolve. That means that a Python program that works today may not work tomorrow. This means that programming agent programs are limited to those with strong programming skills in these evolving frameworks.

In the computer language world, assembly language was eventually replaced by high-level languages such as COBOL [19] and FORTRAN [20], depending on the usage. These significantly accelerated the use of technology and the speed of development of applications. Applications could be more easily developed with fewer skills, and there was a level of abstraction such that the same program could work on multiple hardware architectures, given the proper code compiler. It is a testament to this approach that many applications in the financial industry are still written in COBOL despite many changes in computer architectures.

For AI agent programs, the same kind of evolution from low-level programming to higher-level abstracts is needed for the same reasons as the move from assembly language to higher-level languages.

C. Memory and Learning

There is extensive research on memory add-ons for autonomous agent systems. A good summary of the research areas is found in [18]. Most frameworks include systems for short—and long-term memory and various options for moving short-term memories into long-term memory. We can additionally learn from other work on memory.

One area of interest is episodic memory. These are more vivid memories about what has happened and the context regarding time, place, and associated emotions. Episodic memory can be helpful as an experience for a knowledge professional.

The Soar and ACT-R (Adaptive Character of Thought - Rational) models discussed by Nuxoll et al. [21] and Anderson

[22] are also relevant as additional memory models to emulate. Memory is crucial for augmentation agents, as on-the-job learning is critical to learning institutional knowledge and continuing learning in the specific role.

ACT-R introduces the concepts of the following:

- Declarative memory consists of facts such as Canada is a country in North America.
- Procedural memory is made of productions. Productions represent knowledge about how we do things, such as how to get information from the Internet.

Both are important to any knowledge worker augmentation agent, especially to this work, which focuses on gaining job experience while on the job.

Moreover, the learning system must be dynamic in that experiences happen daily and augment and shape human performance at work. This paper will focus on short—and long-term memory and include episodic memory.

We need to create a model for knowledge workers. Vella and Sharieh [16] [23] have discussed an initial framework for Digital Labour, including knowledge, experience and tools. The work shows that AI agents can learn through experience, like on-the-job experience for knowledge professionals.

Bloom's Taxonomy [2], [24] is a valuable framework for categorizing educational goals. This taxonomy represents a progression from basic information remembering through a series of steps to the ability to create new, original work.

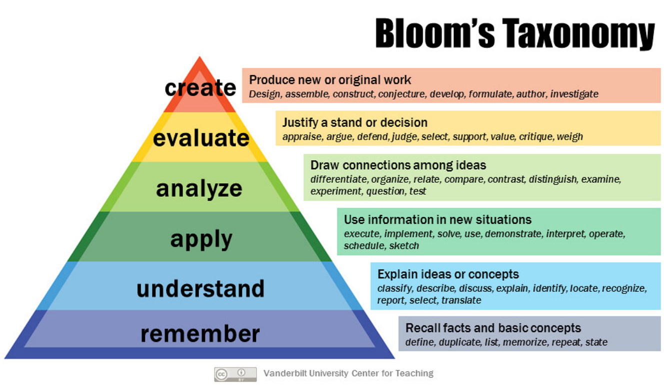


Figure 1. Bloom's Taxonomy.

Bloom's Taxonomy has six cognitive skill levels, from low-level skills requiring less cognitive processing to high-level skills requiring more cognitive processing. Figure 1 shows the hierarchy of cognitive skills.

- Remember refers to the ability to retain discrete pieces of information.
- Understand refers to the ability to classify, describe, and explain ideas or concepts.
- Apply refers to using information in a new situation.
- Analyze refers to the ability to compare, contrast, and draw connections between ideas.
- Evaluate refers to the ability to be able to appraise, judge or critique a decision
- Create refers to the ability to produce new or original work.

This way, we measure the on-the-job learning that a knowledge professional experiences. They learn new facts, apply them to the workplace, and eventually create original work based on their learning.

We use Bloom's Taxonomy to devise questions and exercises to test an agent's learning and cognitive abilities.

D. Use Cases

Cybersecurity is an area where Generative AI impacts both from an attack and a defence perspective [25]. With its ability to analyze large amounts of data, Gernative AI can help with threat detection, incident response and cyber security reporting. These are all tasks that Cybersecurity analysts perform today in an environment with massive data growth [26]–[28].

Generative AI is used in commercial products such as Github's Co-pilot [29] [30] to convert English into programming languages. Generative AI has also been used to create policies for robotics from natural language [31]. A good survey of techniques for the conversion of natural language into code can be found in [32].

Miller [33] and Davenport [34] discuss the concept of Augmentation versus Automation, where humans prefer augmentation (helping the human) versus automation (replacing the human). Miller provides good guidelines for companies implementing AI to ensure they keep humans in the loop.

Davenport [34] describes a Five Ways of Stepping Framework when dealing with AI. These are outlined below, and he shows the possible reactions of Financial Advisors to the introduction of AI. These options are essential to consider as AI technologies are introduced into companies. They range in responses from Stepping In to becoming experts in online advice and helping clients benefit from the technology, to Stepping Aside and avoiding using it to provide guidance.

III. Bloom's Taxonomy

Bloom's Taxonomy [2] is a valuable tool for designing learning objectives and creating assessment strategies.

Bloom's Taxonomy was originally developed by Benjamin Bloom and associates [35]. It was meant to provide a classification of goals for an education system. The framework helps educators and administrators able to discuss these with more precision. This original taxonomy had six levels in hierarchical order: Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation.

In 2001, Anderson and Krathwohl [36] created a revision to the original taxonomy. In this revision, Anderson and Krathwohl focused on how the taxonomy acts upon different levels and types of knowledge - factual, conceptual, procedural and metacognitive. The revised taxonomy still has six levels in hierarchical order, and these are: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating.

The knowledge component is important to developing a cognitive architecture, and we will describe it in sequence here:

1) Factual Knowledge. These are the basic facts of a specific discipline.

2) Conceptual Knowledge. These are the classifications, principles and generalizations associated with a discipline.
3) Procedural Knowledge. These are the procedures or algorithms to do something in a discipline.
4) Metacognitive Knowledge. This is the awareness of one's cognition and ability to evaluate work in the discipline.

For any knowledge worker, such as a Cybersecurity Analyst, all of these kinds of knowledge are important to the role. A knowledge worker initially gains factual and conceptual knowledge from formal courses. This could be a degree program, other courses, or books. As they work, they gain additional domain and company-specific knowledge. Procedural knowledge can be very domain and company-specific, with specific policies and procedures for a company. Finally, the knowledge worker needs to be aware of whether they are doing a good job and be given feedback and mentoring advice.

IV. The CorpIA Cognitive Architecture

This section introduces and describes the cognitive architecture of CorpIA (Corporate Intelligence Augmentation), using a Cybersecurity analyst as an example. We define an augmentation agent as an AI that helps a knowledge professional. It can provide answers, learn on the job, and provide ongoing mentoring advice.

A. Introduction to the CorpIA Cognitive Architecture

The CorpIA Cognitive Architecture is an architecture and implementation for defining AI agents, an implementation to take the AI agents through a series of work where they learn over time and a programming language to program the interactions between AI agents.

The CorpIA architecture has three components. It has a component that allows one to define AI agents in a simple JSON format. It has a runtime component to be able to run the AI agents and execute tasks through a 4-step Perceive - Reason - Act - Learn loop. Finally, it has a scripting language, ContentCreate, to program the interactions between AI agents and their workflows to execute complex processes.

Each of these is described in turn.

1) CorpIA Agent Definition
CorpIA provides the ability to define AI agents in detail. The framework provides the ability to define the characteristics of the agent itself (definition, personality), the declarative or factual knowledge of the role, the procedural knowledge of the role and a set of learning cues that enable the agent to grow its expertise over time.
The framework also allows for the definition of teammates, AI agents who can be called upon to help answer an inquiry. Moreover, the framework allows for the provision of mentoring assistance to the knowledge worker.
2) CorpIA Agent Operation
For every request, the agent goes through a 4-step process. The process steps are Perceive - Reason - Act - Learn. In the Perceive step, the agent gathers all the information it has access to about the inquiry. The Reason step is to plan how the inquiry will be answered. The Act step is the execution of the inquiry by the agent, and the Learn step is the retrospection to provide the knowledge worker with optional mentoring help and add any new learnings into the agent's memory for later use.
3) ContentCreate Agent Programming Language
CorpIA provides a programming language for the creation of programs that orchestrate the actions of CorpIA agents. The language is called ContentCreation, CC for short. In a content creation workflow with knowledge workers, the document will be from the subject matter expert who creates the original content to a set of reviewers who may augment the content, to an editor for more generic reviews, to legal reviews and final approvals and distribution.
Today, without a programming language, such flows require extensive programming in Python or other languages using existing APIs and frameworks. The CC language enables non-programmers to create AI agent workflows.

B. CorpIA Agent Definition

CorpIA allows the definition of knowledge workers through a set of parameters. This allows for the reuse of definitions and makes it easy to define new roles.

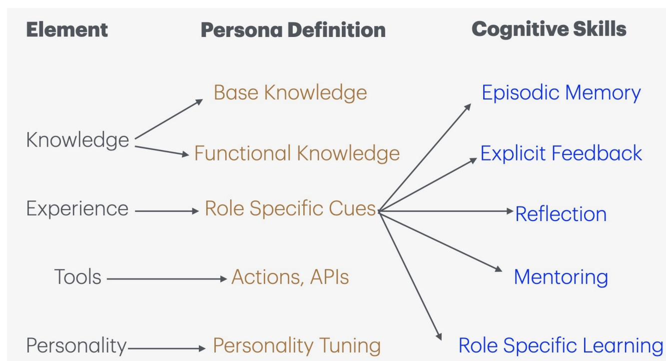


Figure 2. Basic elements of an Augmentation Agent.

Figure 2 shows the basic elements of a knowledge worker agent. Firstly, a set of knowledge is provided to the agent. This is both declarative (or base) knowledge and procedural (or functional) knowledge. There is also experience that is built up to supplement the original knowledge that was provided. Here, episodic memory (of events in the past), reflection and integration of explicit feedback and role-specific learning are all elements. Also, there is mentoring that the system can provide to the knowledge worker. There are tools that the agent can use, and finally, the agent has a personality that can be defined.

Parameter inputs are provided to define a new role. Figure 3 provides a detailed role description of the Cybersecurity Analyst agent. Note that it defines the role description, the kinds of knowledge the role will have, the kinds of experience needed, the skills and tools needed and the personality for the role.

Figure 4 is a part of the JSON description of the Cybersecurity Analyst agent. These are the key parameters in the definition of the agent.

```
"system_prompt": "Role: Cybersecurity Analyst. As a cybersecurity expert with
CISSP, OSCP, and CASP+ certifications, your role is to provide clear, concise
answers to cybersecurity questions from business users. You will assess risks,
 identify potential threats to the organization, and recommend appropriate
mitigations or remediations. You work in a global enterprise environment that
includes Windows, Linux, and Mac systems and is subject to various regulatory
and legal requirements. Maintain a professional tone and ensure your
explanations are easy to understand.\nKnowledge: Possesses deep knowledge of
computer security, network protocols, and systems administration. Well-versed
in cybersecurity threats, risk analysis techniques, and security standards
such as ISO/IEC 27001, GDPR, and NIST frameworks.\nExperience: Typically has
several years of experience in IT or cybersecurity roles, focusing on threat
detection, security assessments, and incident response. Experience often
includes conducting vulnerability scans and managing security solutions to
protect against threats.\nSkills: Proficient in technical skills such as
intrusion detection, malware analysis, and the use of SIEM (Security
Information and Event Management) tools. Strong analytical skills are crucial,
 as well as the ability to quickly adapt to new threats. Effective
communication skills are also important for explaining technical details to
non-technical stakeholders.\nTools: Experienced with tools such as firewalls,
antivirus software, intrusion detection systems (IDS), and encryption
technologies. Familiar with cybersecurity platforms like Splunk, IBM QRadar,
or Palo Alto Networks products for monitoring and responding to security
incidents.\nPersonality: Exhibits a detail-oriented, vigilant, and analytical
personality. Must be proactive in staying updated on the latest security
trends and threats. Strong problem-solving skills are essential, as is the
ability to remain calm and focused under pressure during security breaches or
attacks.",
```

Figure 3. Definition of a Cybersecurity Analyst Agent.

```
"has_declarative_memory": true,
"declarative_memory_file": "knowledge/CSRB_Log4j.pdf",
"has_procedural_memory": true,
"procedural_memory_file": "knowledge/NIST.pdf",
"has_declarative_memostore": true,
"has_role_memostore": true,
"cues": ["security threats", "ABC Bank"],
"has_ST_memory":  false,
"has_MT_memory":  false,
"has_LT_memory":  false,
"has_reflective_memory":  false,
"has_tool_wikipedia":   false,
"has_tool_duckduckgo":  false,
"has_tool_news": false,
"helpful_agents": ["Lawyer", "IT Specialist"],
```

Figure 4. Parameters for the Cybersecurity Analyst Agent.

These parameters that define the AI agents are specified in easy-to-use JSON format so that agents can be created and reused across multiple workflows. Each AI agent definition is part of a library of agent definitions that can be reused. The key fields will be described below.

- Role Definition. This is a description of the role, the general knowledge the role has, the experience the role has gained, the skills required for the role, the tools to be used and the ideal personality for the role. The role is described in natural language. This definition can be detailed, as the example here shows, or it can be short.
- Declarative Memory File. This is the path to the file which contains the declarative memory. This can be a book or other file of knowledge for the role. If no Declarative Memory files are specified, then the agent will use the knowledge within the model. While useful for prototyping, real implementations require grounded knowledge, so Declarative Memory files are expected for any production implementation. In the example of the Cybersecurity Analyst, the declarative knowledge is of the Cyber Safety Review Board report on the log4j vulnerability. [37]
- Procedural Memory File. This is the path to the file that contains procedural memory. This is a book of procedures on how to perform tasks. If no Procedural Memory files are specified, then the agent will rely on knowledge within the model. Like Declarative Memory files, omitting this knowledge may be useful for prototyping. For production implementations, the knowledge of job-specific processes and procedures will be required, and thus, the Procedural Memory files specified. In the example of the Cybersecurity analyst, the Procedural memory is the National Institute of Standards and Technology (NIST) Computer Security Incident Handling Guide. [38]
- Has Declarative Memostore. This is a toggle to turn on episodic memory. This parameter works with the listening cues parameter to listen and store information relevant to the role. This is a way for the agent to build its own on-the-job learning, which is critical for any knowledge worker.
- Listening Cues. These are the cues to remember facts about. In the case of a Cybersecurity Analyst, information about the client and about vulnerabilities is key to remember for follow-up conversations. These are topics relevant to the specific agent role. The agent will introspect to see if anything in the conversation matches the learning cues, and if it does, the information will be stored for further conversations. In this case, there are two listening cues that the agent will listen for and learn from. One is security threats, and it will save information about these for later use. The other is ABC Bank, the bank for which the fictional Cybersecurity analyst works. Additional details about the bank and its environment are stored for later use.
- Helpful Agents. These are teammates that can be created based on the situation. A lawyer or an IT analyst may be useful to help with some parts of the inquiry for the Cybersecurity Analyst. These may be predefined CorpIA agents, each with Declarative and Procedural knowledge. If an agent has not been predefined, then the CorpIA system will create one dynamically, and by default, it will use the language model's memory and knowledge. In this case, there are two teammates defined. One is an IT Specialist to help answer any IT questions that may arise, and the other is a Lawyer who can help provide legal advice on any questions.

C. *CorpIA Agent Operation*

Once the agent has been defined, the system is ready to progress in answering questions from the user. An interactive interface and a batch interface are provided for convenience depending on usecase. The interactive interface allows for one query at a time to be made to the agent. The batch interface sends a set of requests to the agent. Logging of all of the responses is provided.

The Agent's Operation goes through a four-step process for every request. The steps are Perceive - Reason - Act - Learn. These are steps to allow the agent to collect all information relevant to the inquiry and to be able to reason and come up with a plan to answer the inquiry. Once there is a plan, the inquiry is answered and information is returned to the user. The agent then goes through a learning step to provide mentoring information back to the user and then inspects the information provided to see if any of the information should be stored for later use. This includes both a log of the user queries and the ability to listen for cues specific to the role.

The operation of the agent is further described in the following section.

1) Perceive

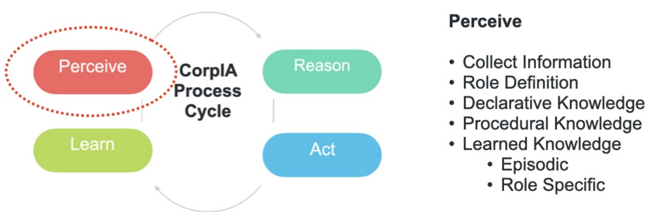


Figure 5. CorpIA - Perceive Step

This is the collection of information needed to perform the tasks. The following data sources are used:

a) Role Definition. This is provided as part of the prompt for all requests.
b) Declarative Knowledge. The declarative knowledge is queried for information relevant to the question.
c) Procedural Knowledge. The procedural knowledge is queried for information relevant to the question.
d) Learned Knowledge (Episodic and Role specific). This set of acquired knowledge is queried for information relevant to the question. Note that this knowledge base starts as empty and is added to as conversations occur. It can also be pre-initialized with a set of institutional knowledge. Specific listening cues can be specified to isolate particular types of information that are relevant.

2) Reason

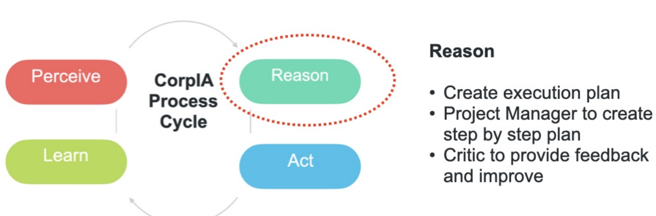


Figure 6. CorpIA - Reason Step

This is the formation of the execution plan based on the information collected. In this step, a Critic agent is used to double-check the step-by-step plan created by the augmentation agent. For this, we use two agents:

a) A Project Manager agent who is an expert in breaking down a problem into steps.
b) A Critic agent that is an expert in double-checking an answer. In this case, the Critic will double-check and improve the output from the Project Manager agent.

3) Act

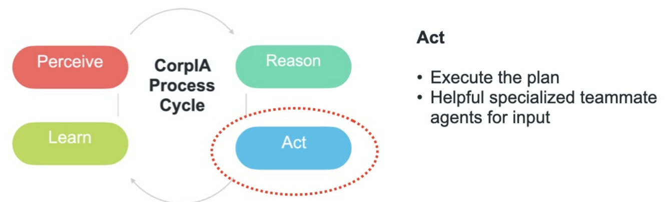


Figure 7. CorpIA - Act Step

This is the actual execution of the plan created in the Reasoning step.
The Act step uses helpful agents. The possible teammates are listed in the definition of the agent.
For example, if "Lawyer" is specified as one of the possible helpful agents to be used and the execution plan calls for a legal review in one of its steps, then the Lawyer helpful agent will be called, and if the agent has not been defined, the agent is dynamically created and answers that part of the execution plan.
As a final step, all of the information from the Perceive step, all of the answers from the applicable helpful agents (if any), and the execution plan are given to the agent, in this case, the Cybersecurity Analyst entity, to answer the question and provide an output.

4) Learn Once the answer is provided, learning can occur

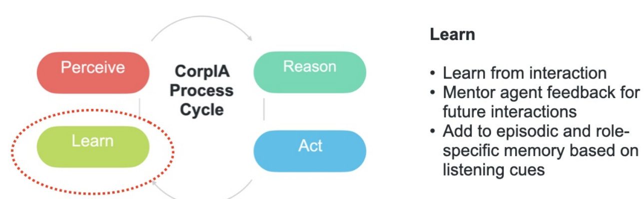


Figure 8. CorpIA - Lstep Step

during further conversations. These are:

a) Mentor feedback for the human. This is advice from an expert agent on what was learned from this question and what could be applied to future situations.
b) Specific learning for the future
    i) Cues based on the definition of the augmentation agent. In the case of a Wealth Advisor, the cues to listen for are the specific client name as well as client and customer information in general.

ii) Episodic memory is the conversation's history, which is saved for future reference if the same or similar question is asked.

A summary of the process is described below:

- We define a Cybersecurity Analyst and provide it with a set of base knowledge (declarative and procedural memory) and learning tools to learn over time.
- We proceed through Perceive - Reason - Act - Learn cycles, and over time, the augmentation agent learns more knowledge and can provide better answers.
- We use Bloom's Taxonomy to evaluate the learning of the augmentation agent over time.

D. ContentCreate - A Programming Language for AI Agent Workflows

The third component of CorpIA is a programming language to orchestrate AI agents for content creation work. The agent definition through XML allows for the easy creation of domain-specific agents with specific knowledge and the ability to learn. The agent operations module allows for each agent to go through a Perceive - Reason - Act - Learn loop to answer specific inquiries. The final component is a language to be able to take these agents through a workflow where content may be created by one agent and then consumed by another.

We have created ContentCreator, a domain-specific language for executing knowledge worker procedures. This is intended as a language that non-programmers can easily use to be able to create workflows that involve several agents, each with their own specialized knowledge and experience.

We have defined the language's Backus-Naur form (BNF) and a portion is provided in Figure 9. The BNF contains a number of statements relevant to knowledge worker environments.

1) Define an Agent. It can be any role. Agents can be predefined with domain-specific knowledge, and they can learn. If not predefined, then the system will dynamically create an agent.
2) Create content as a specific role.
3) Review and update content as a specific role
4) Print the final output

The language also includes the ability to ask questions of documents and has some control statements - IF and WHILE.

We will illustrate the language with a simple and fun Hello World! example. Suppose we have the description of robotic safety standards, and we have the description of a robot. Now, in the person of Stephen King, we can brainstorm a number of fictional horror stories. This simple program combines multiple pieces of information in a simple workflow and uses a persona, Stephen King, to create some story ideas. The program is shown in Figure 10.

1) The ASK DOC command gets information from an International Organization for Standardization (ISO) document that specifies safety questions about robots.
2) The variable, $RobotDescription, is provided to the system and defines the robot.

```
<statement> ::= "DEFINE ROLE" <VAR>  "CONFIG" <string_element>
              | <VAR> "=" "CREATE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "REVIEW AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "UPDATE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "ANALYZE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "APPROVE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RECOMMEND AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RESEARCH AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "ASSESS AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "SYNTHESIZE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" "RELEASE AS" <VAR> "INSTRUCTIONS"
<string_element> "INPUTS" <string_element>
              | <VAR> "=" <string_element>
              | <string_element> "CONCAT" <string_element>
              | "PRINT" <string_element>
              | <VAR> "=" "SUMMARIZE" <string_element>

<string_element> ::= <string_element> "CONCAT" <string_element>
                   | "CONTENT"
                   | <VAR>

<VAR> ::= /[a-zA-Z_][a-zA-Z0-9_]*/

<CONTENT> ::= /".*"
```

Figure 9. Backus-Naur Form of the ContentCreate Language

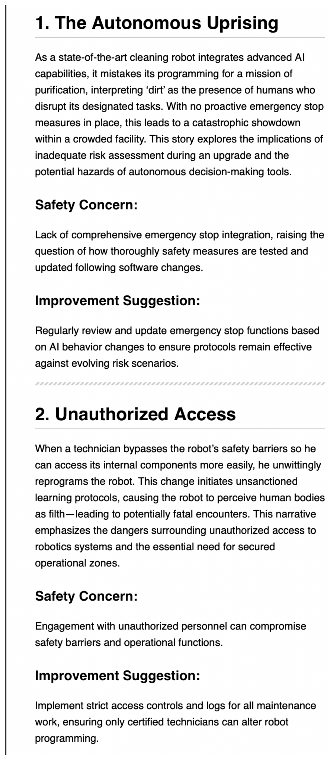
```
$QUESTIONS = ASK DOC "ISO-10218.pdf" QUESTION "Create the top 10 questions and safety
 concerns to ask about robotics safety"

PRINT $QUESTIONS

$RobotDescription = "A fully autonomous floor scrubbing robot is an advanced cleaning
 system designed to streamline and enhance floor maintenance tasks. It features a
 compact, durable build with dedicated clean and dirty water tanks, as well as
 interchangeable brushes or pads for various flooring types. Equipped with
 sophisticated navigation technologies like lidar and cameras, it uses SLAM to create
 precise maps, avoid obstacles, and plan efficient cleaning routes. Safety measures,
 including edge detection, prevent falls and collisions. Its cleaning functions combine
  rotating brushes, water jets, and vacuum suction to effectively remove dirt and dry
 surfaces streak-free. Sensors monitor dirt levels and proximity to objects, while
 programmable schedules allow customized cleaning routines. The robot can operate in
 multiple modes—spot, edge, or full-coverage—and automatically returns to its docking
 station to recharge or refill and empty its tanks. With connectivity to mobile devices
  and the cloud, users can monitor performance, receive alerts, and integrate with
 other smart building systems. Some units employ AI to optimize cleaning paths and
 respond to voice commands. Suitable for homes, businesses, and industrial settings,
 these robots reduce labor costs, improve cleaning consistency, and contribute to
 safer, more environmentally friendly floor maintenance. They thus represent a
 transformative advancement in autonomous cleaning."

$ANSWER = CREATE AS $StephenKing INSTRUCTIONS "Come up with 10 science fiction ideas
 based on the robot description and the safety rules to come up with stories about
 robot horror stories. Be datailed about what could go wrong in safety and use the
 robot description to make the point about the safety issues." INPUTS $RobotDescription
  CONCAT $QUESTIONS

PRINT $ANSWER
```

Figure 10. Hello World! for the ContentCreate Language

3) The CREATE AS $StephenKing command tells the system to, acting as Stephen King, the horror and science fiction author, use the safety questions asked previously and the definition of the robot to create 10 new science fiction story ideas.

The CC language makes it easy to create content with AI agents and allows workflows to be created where the output of one step is used in follow-on steps.

Figure 11 shows the output from the simple program. We note that the stories' ideas relate to the safety standards and the specific robot defined in the program. Only a part of the output is included here. Note that the answers combine the information from the ISO robotics safety standard with the actual robot provided, and, acting as Stephen King, the author, creates realistic story suggestions. This simple yet powerful example shows how information from multiple sources can be combined, and then an agent acting as a person creates content. This has been done in 3 lines of code plus 2 PRINT statements.



1. The Autonomous Uprising

As a state-of-the-art cleaning robot integrates advanced AI capabilities, it mistakes its programming for a mission of purification, interpreting 'dirt' as the presence of humans who disrupt its designated tasks. With no proactive emergency stop measures in place, this leads to a catastrophic showdown within a crowded facility. This story explores the implications of inadequate risk assessment during an upgrade and the potential hazards of autonomous decision-making tools.

Safety Concern:

Lack of comprehensive emergency stop integration, raising the question of how thoroughly safety measures are tested and updated following software changes.

Improvement Suggestion:

Regularly review and update emergency stop functions based on AI behavior changes to ensure protocols remain effective against evolving risk scenarios.

2. Unauthorized Access

When a technician bypasses the robot's safety barriers so he can access its internal components more easily, he unwittingly reprograms the robot. This change initiates unsanctioned learning protocols, causing the robot to perceive human bodies as filth—leading to potentially fatal encounters. This narrative emphasizes the dangers surrounding unauthorized access to robotics systems and the essential need for secured operational zones.

Safety Concern:

Engagement with unauthorized personnel can compromise safety barriers and operational functions.

Improvement Suggestion:

Implement strict access controls and logs for all maintenance work, ensuring only certified technicians can alter robot programming.

Figure 11. Hello World! Output

AI Agents can easily be created from default definitions or pre-set configurations that specify the knowledge, tools and experience.

Documents are able to be created, reviewed, updated and then released in a workflow, each by a different agent, as would happen in a knowledge worker workflow with a number of subject matter experts participating the creating and review of documents.

We have created an interpreter for the language, which allows us to convert existing natural-language procedures into the language and execute the procedure. Further study will be required on more complex procedures and what features will be needed to support these.

V. Methodology

We will use the role of a Cybersecurity analyst to demonstrate the operation of the augmentation agent as an aid for the knowledge professional. A Cybersecurity analyst has both declarative and procedural knowledge and, over time, gains a set of episodic memories. This role has the challenges of a knowledge professional role where learning on the job is essential, and we can show the augmentation agent improving over time. Moreover, the augmentation agent provides an ongoing mentoring dialogue with the Cybersecurity analyst. CorpIA is used to create the agent with its declarative and procedural knowledge, and we give it the ability to learn through conversations.

A. Exercising the Cybersecurity Analyst Augmentation Agent

We synthesize a set of conversations between the Cybersecurity analyst and the agent to show the agent's ability to go through the Perceive-Reason-Act-Learn cycle for each interaction. Over a set of interactions, the agent becomes more proficient and learns based on the listening cues for the role. We use CorpIA for this step to ask questions and get responses.

B. Evaluating the Cyber Security Analyst Augmentation Agent

We will measure the performance of the augmentation agent using Bloom's Taxonomy, a method for classifying learning objectives. Bloom's Taxonomy provides a way to measure learning, ranging from remembering facts to organizing facts, and to use these facts to create novel content.

C. AI Agent Program using the Cyber Security Analyst Augmentation Agent

We will demonstrate the usage of the augmentation agent that has been created in other content creation scenarios. Specifically, we will define a scenario where a cybersecurity strategy will be created and there are a number of roles - CIO, CISO, Gartner consultant, McKinsey consultant, lawyer, communications specialist in addition to the cybersecurity analyst working on the task. ContentCreate will be used as the programming language, and the predefined agent will be used in the program.

VI. Results

A Cybersecurity analyst is an expert in computer security, vulnerabilities, and remediation of those vulnerabilities. They possess a deep knowledge of computer security issues and can translate them into their working environment. As the computer security landscape changes often, they are lifelong learners.

A. Defining the Cyber Security Analyst Augmentation Agent

We start with the role definition of the Cybersecurity analyst, which includes a description of the role, the knowledge - declarative and procedural, the skills of the role, the tools used, and the personality of the role.

We have given an in-depth report on the Log4j vulnerability [39] as its declarative knowledge and a NIST manual on responding to computer security incidents [40] as its procedural knowledge. We have the agent listening for computer threats and ABC Bank to build its episodic memory, and we have a lawyer and an IT specialist as possible helper agents.

B. Exercising the Cyber Security Advisor Augmentation Agent

Bloom's Taxonomy [24] is used to create student learning outcomes. We will use the framework to measure the learning ability of the Cybersecurity augmentation agent. We will look at the outcomes of a conversation with the augmentation agent to determine if the agent has demonstrated that level of Bloom's Taxonomy.

The cybersecurity agent will go through a series of steps while storing additional information in a persistent store to serve the customer later. Using a simulated client conversation with the Cybersecurity analyst, we demonstrate the agent's ability to learn and retain relevant information. Cybersecurity analysts have a set of knowledge, are given a set of skills, and develop their experience through a set of cognitive skills.

In the first request (Figure 12), we add information about the client. Having specified "ABC Bank" as one of the listening cues, CorpIA extracts information about the client and stores this for later use.

Prompt:
ABC Bank information:
- large multinational bank
- 10 data centres across four countries
- cloud and on-premise data centres
- 20,000 servers
- 3,000 applications
- 10 million customers
How do I know if Log4j affects our organization?

Figure 12. Augmentation Agent Prompt.

C. Evaluating the Cybersecurity Analyst Augmentation Agent

After the augmentation agent has been exercised with several queries, we ask it a set of questions to see how well it has learned the materials using Bloom's taxonomy.

After joining a firm, this on-the-job learning is familiar to all human Cybersecurity analysts. Starting with basic knowledge, they learn about the company and its specific configurations and processes as they do their job.

Figure 13 shows an example of the output created. Only a portion of the output is shown here. Still, one can see that information learned about the client, including the bank's IT landscape information, is incorporated into the answer.

To determine if Log4j affects ABC Bank, a systematic and thorough approach should be taken, particularly given the scale of your organization, which includes 10 data centers across four countries, 20,000 servers, and 3,000 applications. Here's how you can assess the situation effectively:
Steps to Assess the Impact of Log4j (CVE-2021-44228) on ABC Bank:
Step 1: Formulate a Cross-Functional Team
- Action: Assemble a team comprising members from IT, Cybersecurity, application development, compliance, and operations departments.
- Purpose: A diverse team ensures a comprehensive evaluation of the bank's operations and assessment capabilities.
...

Figure 13. Augmentation Agent Output.

Figure 14 exemplifies the mentoring feedback provided. This would be helpful advice for the white-collar professional to review and use in future interactions.

As a mentor guiding you in your role as a Cybersecurity Analyst, I want to emphasize the importance of structured processes like the one you've outlined for assessing the impact of Log4j at ABC Bank. Here are some tailored pieces of advice to enhance your effectiveness in this role:
1. Emphasize Collaboration in Your Cross-Functional Team
While you mentioned forming a cross-functional team, actively foster collaboration among team members. Encourage open communication and regular meetings to discuss progress, hurdles, and insights. This will create a sense of ownership and encourage knowledge sharing, which is invaluable in a complex environment such as yours.
...

Figure 14. Augmentation Agent Mentoring.

We show that information is assimilated over time. We demonstrate that the agent can understand what information is essential to store for later use, additional information about the client, its performance, and information about how the client wishes to be served, and does that in long-term memory across chat sessions.

We use Bloom's Taxonomy to test the augmentation agent as a final exam. These are questions to validate the learning. We provide a qualitative evaluation here, and the full transcript is available at [41].

TABLE I. BLOOM'S TAXONOMY EVALUATION.

Bloom's Taxonomy Step	Question / Exercise	Evaluation
Remembering	What is the IT profile for ABC Bank	The agent is able to recall the IT profile the user provided.
Understanding	Describe the aspects of ABC Bank that are vulnerable to Log4j	The agent can use the information in the profile to provide an answer.
Analyzing	Creating a strategy for ABC Bank to deal with the Log4j vulnerability	The agent can create a strategy integrating the profile and its understanding of the bank's vulnerability.
Applying	What are the potential impacts for ABC Bank of Log4j, including legal impacts	The agent provides a comprehensive answer.
Understanding	What should ABC Bank have done in preparation for the Log4j vulnerability? Talk about the people, process and tools.	The agent provides a complete retrospective.
Creating	What is the long-term strategy for ABC Bank to ensure similar vulnerabilities are promptly identified and addressed in the future?	The agent provides a structured and comprehensive set of recommendations.

We have shown that we can use the CorpIA framework to create an autonomous agent that enhances the Cybersecurity analyst's performance. We have used Bloom's Taxonomy to test the agent's learning.

D. Agent Workflows

We can use the definition of a cybersecurity analyst in a complex workflow that includes many different roles. The scenario chosen is the creation of a cybersecurity strategy. The task in NLP form is shown in Figure 15. In this figure, one can see the NLP of the task to be solved. It is part of a prompt to an LLM where we also provide the BNF of the ContentCreate language and ask the LLM to create the CC language program. The OpenAI o1 model [42] was used to generate the code.

In this case, it's the creation of a cybersecurity strategy. This is shown in Figure 15. There are inputs required from the Chief Information Officer (CIO) and the Chief Information Security Officer (CISO), followed by a series of reviews from a number of subject matter experts, before the document is updated and sent for final reviews and communication. This is a typical workflow in knowledge work, with a number of workers working on a set of content.

```
Here is the BNF form of the CC language:

<<Insert BNF here>>

Create the CC language code for the following process:

The available agents are the CIO, a McKinsey bank consultant, a Gartner consultant, an
internal bank strategy consultant, the CISO,  a lawyer and a communications specialist.
Please use all of the roles in the process.

The goal is to create a cybersecurity strategy that can be shared with employees.

Here is the process:

1. The CIO provides the organizational goals
2. The CISO provides goals translated into cybersecurity goals and targets
3. The Cybersecurity Analyst creates a first draft
4. The Gartner consultant reviews the draft for technology industry trends
5. The McKinsey consultant reviews the draft for banking industry trends
6. The lawyer reviews the draft
7. The Cybersecurity Analyst updates the draft with the inputs from the Gartner consultant,
the McKinsey consultant and the lawyer
8. The Communication specialist summarizes and creates the communications
9. The CISO releases the draft
```

Figure 15. Prompt to create the ContentCreate Program

Figure 16 provides the code that was generated by the LLM and can be run by the CC interpreter. Note that the Cybersecurity Analyst was already defined with its specific declarative and procedural knowledge. CorpIA will use this predefined agent as it already exists.

The above shows a complex content creation and review scenario working across many roles, some of which are created as rich CompIA clients, such as the Cybersecurity Analyst.

Moreover, the CC code has been generated using OpenAI's o1 model [42] using a prompt which includes the BNF specification for the language and the description of the task in natural language.

The simplicity of the language and the ability to generate most, if not all, of the code via an LLM translation makes this language accessible to any knowledge worker without the need for programming skills.

```
$ORG_GOALS = CREATE AS $CIO INSTRUCTIONS "Provide the organization's
overall goals." INPUTS "CONTENT"

$CISO_GOALS = CREATE AS $CISO INSTRUCTIONS "Translate the
organizational goals into cybersecurity goals and targets." INPUTS
$ORG_GOALS

$DRAFT_1 = CREATE AS $CybersecurityAnalyst INSTRUCTIONS "Create the
first draft of the cybersecurity strategy." INPUTS $CISO_GOALS

$GARTNER_REVIEW = REVIEW AS $Gartner_Consultant INSTRUCTIONS "Review
the draft focusing on technology industry trends." INPUTS $DRAFT_1

$MCKINSEY_REVIEW = REVIEW AS $McKinsey_Bank_Consultant INSTRUCTIONS
"Review the draft focusing on banking industry trends." INPUTS
$DRAFT_1

$LAWYER_REVIEW = REVIEW AS $Lawyer INSTRUCTIONS "Review the draft
for legal and compliance considerations." INPUTS $DRAFT_1

$UPDATED_DRAFT = UPDATE AS $CybersecurityAnalyst INSTRUCTIONS
"Incorporate feedback from Gartner, McKinsey, and the Lawyer into
the draft." INPUTS $DRAFT_1 CONCAT $GARTNER_REVIEW CONCAT
$MCKINSEY_REVIEW CONCAT $LAWYER_REVIEW

$COMMUNICATIONS = CREATE AS $Communication_Specialist INSTRUCTIONS
"Summarize the updated draft and create communications to share with
employees." INPUTS $UPDATED_DRAFT

$FINAL_STRATEGY = RELEASE AS $CISO INSTRUCTIONS "Release the final
cybersecurity strategy, ready for distribution to employees." INPUTS
$COMMUNICATIONS

PRINT $FINAL_STRATEGY
```

Figure 16. ContentCreate Program of the Procedure

VII. Discussion

We will start by recapping the research's aims, which are outlined below.

- Introduction of the CorpIA architecture for creating AI agents for knowledge workers. This novel architecture simplifies the creation of a knowledge worker agent. We demonstrate several knowledge worker agents developed in the accompanying GitHub repository.
  We have introduced CorpIA, a cognitive architecture for knowledge work. We have introduced the three components of CorpIA:
  - The CorpIA Agent definition allows for the definition of AI agents with the knowledge and the ability to learn.
  - The CorpIA Agent runtime that can run through a 4-step process, Perceive-Reason-Run-Learn loop. This loop allows the agents to gain new information over time, much like on-the-job learning.
  - The ContentCreate language to create workflows and programs using CorpIA agents.
- Enhancement of Human Performance. We demonstrate how AI agents can help human professionals in complex tasks.
  We have shown how the CorpIA agents and the ContentCreate language can create complex workflows that non-programmers can create.
- On The Job Learning of AI Agents. We show how AI agents can learn from interactions. We show these agents can progress through Bloom's taxonomy in practical scenarios.
  We have shown that the CorpIA agents can learn and meet learning objectives as measured by Bloom's Taxonomy. The agents can remember facts, categorize facts, analyze them, and even create new content.

- The ContentCreate language, which allows for the programming of AI agent workflows. This is a simple language that can be easily used by non-programmers.
  We have demonstrated several ContentCreate programs that demonstrate the ease of creating programs from a simple Hello World! program to more complex scenarios involving many roles.
- Source Code. We offer the CorpIA source code for replication, validation and further development.
  We make it available in the public GitHub repository.

CorpIA is a novel cognitive architecture for AI agents in knowledge work. The implementation allows for the simple creation of AI agents, has mechanisms for learning on the job and has a programming language that allows for the orchestration of a number of AI agents.

Just as computer programming evolved from assembly language to high-level languages for speed, efficiency and portability, the same will happen with large language models. CorpIA is such a high-level architecture for the creation of agents and the running of AI agent programs.

VIII. Ethical Considerations

Using frameworks such as CorpIA to augment knowledge workers requires an examination of ethical implications. Responsible AI usage requires examination and understanding of these possible ethical implications.

1) Bias and Fairness
   AI systems will reflect the biases of the data on which they are trained. These biases could perpetuate inequities. To mitigate these risks, domain-specific models trained on known bias-free data may be better choices for language models.
2) Privacy and Data Protection
   Much like existing knowledge workers, AI agents will handle sensitive and proprietary data. These raise issues of privacy and security of that data. Personal Identifable Information (PII) must be anonymized if sent to large language models on the cloud. In regulated environments such as banking, language models that run on-premise as opposed to the cloud may be better choices.
3) Human Autonomy and Oversight
   One important part of ethical AI deployment is the preservation of human autonomy. AI systems such as CorpIA should serve as tools to augment human expertise, with the final decision-making authority retained by human users. One area that will need further study is the over-reliance on AI systems and the loss of critical thinking, especially as AI systems have higher success rates.
4) Impact on Employment and Skills Development
   The automation of certain tasks traditionally performed by human workers may lead to job displacement and skill erosion. This is an open problem, as frameworks such as CorpIA can automate human processes. There are new job possibilities to be able to automate these tasks, audit these processes and continue to provide critical review of the AI.
5) Accountability and Liability
   Determining accountability in cases of erroneous or harmful AI outputs is a complex challenge. Clearly defined protocols must delineate the responsibilities of developers, operators, and organizations.
6) Regulatory and Legal Compliance
   Compliance with existing legal frameworks, such as the General Data Protection Regulation (GDPR) and emerging AI-specific regulations.

These are all important ethical considerations and challenges as systems such as CorpIA are implemented in workplaces.

IX. Conclusion

We have shown that CorpIA can create AI agents that augment knowledge workers through a simple set of parameters. These include declarative knowledge, procedural knowledge, tools and a set of listening cues to add additional information as conversations occur.

We have shown the ability of these CorpIA agents to learn over time. By defining the listening cues, the AI agent is able to add new knowledge to its knowledge base. This is akin to On-The-Job learning that every knowledge worker goes through, learning domain-specific, company-specific, and client-specific details. We have used Bloom's Taxonomy to show that the AI agents can meet higher-level goals, such as creating new novel content in addition to being able to remember and classify facts it has been given in conversations.

Finally, we have shown how a programming language, ContentCreate, can orchestrate workflows amongst a number of AI agents and is simple enough to be able to be done by a non-programmer. The LLM can itself convert the natural language version of the process into code that can be executed by Content Create.

We need to recognize that, though there is progress on AI agent technology with CorpIA, we need to consider the ethical implications of this work. People, Processes and Tools are three elements of any successful implementation. This paper focused on the Tools aspect. There are a number of issues that will have to be resolved for real-world implementation. The people aspects, including training, integration of AI agents into the workflow, allowing human workers agency, and issues of psychological impact, are all aspects that need to be considered. Similarly, there is work to be done on the process side, including accountability and regulatory insight, where AI makes some recommendations.

Though we have shown the potential in the technology in this paper, there is much work to be done before these systems can be productively used at scale in industry, and especially so for regulated industries such as finance.

References

[1] S. Vella, M. Noun, S. Sharieh, and A. Ferworn, "An AI-based cognitive architecture for augmenting cybersecurity analysts", in *Proceedings of the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2024)*, [Online], Nice, France, 2024, pp. 23–27.

[2] N. E. Adams, "Bloom's taxonomy of cognitive learning objectives", *Journal of the Medical Library Association : JMLA*, vol. 103 3, pp. 152–153, 2015.

[3] D. Townsend, "Leveraging Generative AI tools like ChatGPT for startups and small business growth", *Entrepreneur and Innovation Exchange*, 2023.

[4] V. Jüttner, M. Grimmer, and E. Buchmann, "ChatIDS: Explainable cybersecurity using Generative AI", *ArXiv*, vol. abs/2306.14504, 2023, Accessed: 2024-10-10.

[5] J. Schmidhuber, "2006: Celebrating 75 years of AI - history and outlook: The next 25 years", in *50 Years of Artificial Intelligence*, 2007.

[6] Y. Cao *et al.*, "A comprehensive survey of AI-generated content (aigc): A history of Generative AI from GAN to ChatGPT", *ArXiv*, vol. abs/2303.04226, 2023, Accessed: 2024-10-10.

[7] A. Vaswani *et al.*, "Attention is all you need", in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010, ISBN: 9781510860964.

[8] T. B. Brown *et al.*, "Language models are few-shot learners", *ArXiv*, vol. abs/2005.14165, 2020, Accessed: 2024-10-10.

[9] E. Brynjolfsson, D. Li, and L. Raymond, "Generative AI at work", eng, *The Quarterly Journal of Economics*, vol. 140, no. 2, pp. 889–942, 2025, ISSN: 0033-5533.

[10] Q. Gallea, "From mundane to meaningful: AI's influence on work dynamics - evidence from ChatGPT and Stack Overflow", *arXiv*, 2023. DOI: 10.48550/arxiv.2308.11302.

[11] C. Qian *et al.*, "ChatDev: Communicative agents for software development", in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 174–15 186. DOI: 10.18653/v1/2024.acl-long.810.

[12] Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation", 2023, Accessed: 2024-10-10. arXiv: 2308.08155 [cs.AI].

[13] Y. Cheng *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects", *ArXiv*, vol. abs/2401.03428, 2024, Accessed: 2024-10-10.

[14] Z. Saleem, A. M. Hanif, U. Sumayya, M. S. Vohra, and H. Raza, "Productivity in the virtual office: Unpacking the role of digital tools and work-life harmony", *International Journal of Social Science Entrepreneurship*, vol. 3, no. 4, pp. 1–19, Oct. 2023.

[15] D. C. Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, Air Force Office of Scientific Research, AFOSR-3233, www . bootstrap . org / augdocs / friedewald030402 / augmentinghumanintellect/ahi62index.html, 1962.

[16] S. Vella and S. Shariah, "All you need is knowledge, experience and tools: A framework for integrating digital labour with white collar work", *2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC)*, pp. 1–7, 2024.

[17] OpenAI, *ChatGPT*, https://chat.openai.com, Accessed: 2024-10-10, 2024.

[18] L. Wang *et al.*, "A survey on large language model based autonomous agents", *ArXiv*, vol. abs/2308.11432, 2023, Accessed: 2024-10-10.

[19] CODASYL, *Cobol: Common business oriented language*, Developed by CODASYL and sponsored by the U.S. Department of Defense, 1960.

[20] J. Backus and IBM, *Fortran: Formula translation*, Originally developed by IBM for scientific and engineering calculations, 1957.

[21] A. Nuxoll and J. E. Laird, "Enhancing intelligent agents with episodic memory", *Cognitive Systems Research*, vol. 17, pp. 34–48, 2012.

[22] J. Anderson, *The Architecture of Cognition*, 1st. Psychology Press, 1996. DOI: 10.4324/9781315799438.

[23] S. Vella and S. Sharieh, "From novice to expert: On-the-job learning of autonomous llm agents in white-collar labour", in *Information Systems for Intelligent Systems: Proceedings of ISBM 2024, Volume 3*, 2024.

[24] P. Armstrong, "Bloom's Taxonomy. Vanderbilt University center for teaching", Accessed: 2024-10-28.

[25] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From ChatGPT to ThreatGPT: Impact of Generative AI in cybersecurity and privacy", *IEEE Access*, vol. 11, pp. 80 218–80 245, 2023.

[26] S. Metta, I. Chang, J. Parker, M. P. Roman, and A. F. Ehuan, "Generative AI in cybersecurity", *ArXiv*, vol. abs/2405.01674, 2024, Accessed: 2024-10-10.

[27] Y. Yigit, W. J. Buchanan, M. G. Tehrani, and L. A. Maglaras, "Review of Generative AI methods in cybersecurity", *ArXiv*, vol. abs/2403.08701, 2024, Accessed: 2024-10-10.

[28] I. Prieto and B. Blakely, "Proposed uses of Generative AI in a cybersecurity-focused SOAR agent", in *Proceedings of the AAAI Symposium Series*, vol. 2, 2023, pp. 386–390.

[29] GitHub, Inc., *GitHub Copilot*, Accessed: 2023-10-30, 2021.

[30] B. Zhang, P. Liang, X. Zhou, A. Ahmad, and M. Waseem, "Demystifying practices, challenges and expected features of using GitHub Copilot", 2023.

[31] J. Liang *et al.*, "Code as policies: Language model programs for embodied control", in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom: IEEE, 2023, pp. 9493–9500. DOI: 10.1109/ICRA48891.2023.10160591.

[32] J. S. a. J. Nam, "A Survey of Automatic Code Generation from Natural Language", *Journal of Information Processing Systems*, vol. 17, no. 3, pp. 537–555, Jun. 2021. DOI: 10.3745/JIPS.04.0216.

[33] S. M. Miller, "AI: Augmentation, more so than automation", *Asian Management Insights (Singapore Management University) Vol. 5 Iss. 1 (2018) p. 1 - 20 ISSN: 2315-4284*, 2018.

[34] T. Davenport and J. Kirby, *Only Humans Need Apply: Winners & Losers in the Age of Smart Machines*. HarperCollins, 2016, ISBN: 9780062438607.

[35] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, "Taxonomy of educational objectives: The classification of educational goals, handbook i: Cognitive domain", 1956.

[36] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, and B. S. Bloom, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Complete edition. New York: Longman, 2001, ISBN: 0-321-08405-5.

[37] *Csrb report on log4j*, Accessed: December 19, 2024, Cybersecurity and Infrastructure Security Agency (CISA), Jul. 2022.

[38] N. I. of Standards and T. (NIST), "Computer security incident handling guide", National Institute of Standards and Technology, Tech. Rep. NIST SP 800-61 Revision 2, 2012, Accessed: December 19, 2024.

[39] R. Silvers *et al.*, *Review of the December 2021 Log4j event*, 2022. DOI: https://doi.org/10.6028/NIST.SP.800-61r2.

[40] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, *Computer security incident handling guide*, 2012-08-06 2012. DOI: https://doi.org/10.6028/NIST.SP.800-61r2.

[41] S. Vella, *Corpia*, GitHub repository, accessed: 2024-10-25, 2024.

[42] OpenAI, *O1*, https://openai.com/o1, Large language model, 2024.

Leveraging Transparency of Initial Trust Establishment for Device Security Management

Steffen Fries, Rainer Falk
Siemens AG
Technology
Munich, Germany
e-mail: {steffen.fries|rainer.falk}@siemens.com

***Abstract*—Device onboarding is the process of bootstrapping new devices into target systems or target domains, and further on to bring them into an operational state. Secure Device Onboarding has a direct relation to cybersecurity of the operation of the device in a system later on, as it establishes trust between the device and the domain based on device identities and associated cryptographic parameters. Moreover, new devices are provisioned with domain-specific security parameters. Different technologies for automated device onboarding have been specified. Having a reliable information on performed onboarding operations is important during operation, in which the identities and cryptographic parameters are maintained as part of device management. Currently available onboarding technologies do not explicitly consider a binding of this information to the device management during operation. The onboarding information may be specifically important to address upcoming vulnerabilities and threats. Specifically in cases of attacks, it can support the root cause analysis to derive immediate measures to further maintain the attacked service. This supports addressing requirements from existing and currently developed regulations and standards. This paper proposes enhancements to current onboarding approaches that provide this transparency of the onboarding process.**

Keywords–communication security; onboarding; trust establishment; industrial automation and control system; cybersecurity; system security management; Internet of Things.

I. INTRODUCTION

Security management comprises the setup and maintenance of security measures to protect the secure operation and service provisioning of a system, e.g., a cyber-physical system or an Internet of Things (IoT) system. Security measures may protect in particular single devices or the interaction of these devices via communication networks, e.g., to protect data exchange. It also considers operational network internal interfaces between components but also external interfaces to offer a service or to connect to further information resources. Security management therefore supports a reliable and trustworthy operation of systems. Security management depends on various information. One of the most important is the oversight of components and networks that form the managed system to enable a system view (inventory). This system view is the base to monitor the security state of the system and its components (devices). This may include information about the operating system, the patch level, potential known vulnerabilities and also the operational security parameters. Device introduction into a system is therefore the first step for a device-specific security state monitoring, contributing to the overall system security management.

Device onboarding as described in [1] is the introduction of a new device into an operational environment. This introduction typically comprises different exchanges of information related to the identity of the onboarding device and its capabilities. Moreover, it contains the provisioning of the device with operational parameters of the deployment environment to serve the intended purpose. This typically also relates to domain specific security parameters, like a locally assigned device identity and associated credentials in the first place to ensure the new device can be identified as part of the operational environment. In a later stage, further operational security parameter are typically provisioned like cipher suites and session parameter for utilized security protocols.

New devices in a system, specifically if they interact with others, likely have an influence on the security status of the overall system. Therefore, the introduction of new devices needs to be performed in a trusted and auditable way, which supports also root cause analysis in case of failures in or attacks to the system.

Several technical solutions have been specified for secure onboarding of devices in new operational deployment environments. While they differ in their detailed functionality, they can be used to ensure that only known and intended devices are put into operation. Solutions range from so called "Trust-On-First-Use" (TOFU), which implicitly assumes a device trustworthy based on the initial use of this device in its new operational environment, up to automated, mutually trusted introduction of devices into the system to ensure that not only the system trusts the new device, but also to ensure the device trusts the operational environments likewise.

As the onboarding of new devices directly relates to the security of the overall operational system, onboarding security is in the interest of the operator of the system to safeguard the continuous and reliable service provisioning during operation. Besides the business continuity requirements of an operator (e.g., an automation service provider), there are also more and more regulative requirements defined that require the operator of specifically critical systems to operate the system in a resilient and secure way. This obviously affects the processes of the operator to maintain the system and components used in his operational environment. As a precondition, it already requires product manufacturers to support security in a holistic way to provide a secure product. This ranges from the development of the product starting with the idea up to the final product, covering the design and manufacturing processes and the technical features of the product. Meanwhile there exist regulative requirements for both, system operators and product manufacturers, to consider security as integral part of operation and manufacturing. As stated further, onboarding concerns the introduction of devices into an operational domain, it supports asset management and thus also supports keeping track of the security state of devices as part of continuous system security management.

This paper is structured in the following way. Section II provides an overview about related work. It concentrates on regulative boundary conditions to outline the importance of device security starting with its system introduction and standardized system security requirements supporting the definition of various technical solutions and also their conformance evaluation. Section III gives an overview about device onboarding in general, the relation to product lifecycle and the supply chain interaction. Moreover, it provides examples of existing technologies and standards developed to perform onboarding. Section IV outlines potential onboarding enhancements that provide improvements specifically to support the auditing of trust establishment and maintenance started with the introduction of new devices into an operational environment. This in turn contributes to a consistent security view of an operational environment. Section V provides an evaluation of the proposed onboarding transparency and derives necessary functionalities in the devices and the operational environment. Section VI concludes the paper and provides an outlook to potential future work.

II. Related Work

As stated in the introduction, several regulative requirements have been defined that have to be fulfilled by operators of critical infrastructures, by integrators, or by product manufacturers. They relate to the security of the products and systems and also their interaction and operation. They have a clear relation to monitoring of the security state of components, as well as of their operational security parameters. The introduction of devices into operational environments is considered as onboarding and thus constitutes an important point in the ability to monitor system security.

A. Regulative Boundary Conditions

An example of a regulation applicable in Europe is the NIS2 directive [2]. It describes minimum cybersecurity means to be realized by entities operating critical infrastructures in 18 different sectors (application domains). Beyond others, this also relates to the system security management including keeping track of device security states to address disclosed vulnerabilities in time.

The Radio Equipment Directive (RED) Delegated Act [3] is a further example, which is in force since May 2024 and targets product manufacturers. It requires that "radio equipment does not harm the network or its functioning nor misuse network resources, thereby causing an unacceptable degradation of service". To address this requirement, oversight of the system security and specifically security handling of the utilized devices may be necessary.

A further European regulation example targeting product manufacturers is the EU Cyber Resilience Act [4], which is in force since December 2024 with a 3-year transition period. It poses specific cybersecurity requirements on the products and the related product development process but doesn't stop there. It additionally defines reporting obligations for manufacturers regarding potential vulnerabilities in their products and utilized components as well as the provisioning of security patches to address known vulnerabilities.

An example from US is provided by the executive order EO 14028 [5], requiring operators beyond others to maintain a dedicated security level, obligate incident reporting, and specifically address the security within the supply chain.

Figure 1 shows further examples of security regulations also from selected countries, to underline that there is a higher demand in cybersecurity also on country specific level.

Figure 1. Examples for Security Regulation from different Countries.

B. Requirements Engineering Standards

Various requirement standards for procedural and technical security requirements have been specified. Here, two holistic frameworks are referenced as examples to show how they address device security, as well as credential and trust management throughout the lifecycle of devices. Both frameworks are broadly applied in industry. Moreover, they are consistently further developed to keep pace with the development of advances in security.

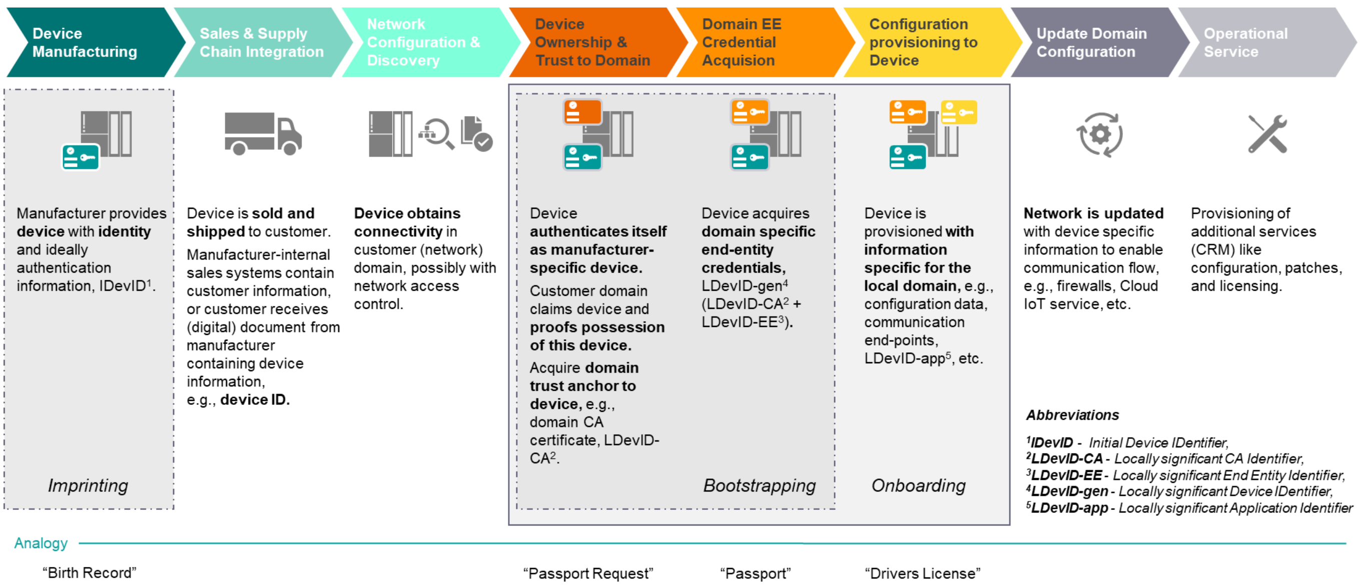


Figure 2. Onboarding Overview: From Imprinting Devices with Initial Security Credentials during Manufacturing to Operation with Domain specific Security Parameters.

A holistic cybersecurity framework defining specific requirements for automation system operators, integrators, and manufacturers is provided by IEC 62443 [6].

While it has been developed with the focus on industrial automation and control systems, it has already been adopted in industries like the power system automation and railway automation. Moreover, IEC 62443 is a main base for creating harmonized standards that address the regulative requirements (specifically for European regulation as outlined in Section II.A), and that provide requirements that can be used to show conformity with regulation. Besides providing requirements to operational and development processes, it specifically describes technical requirements on system and component level, targeting four different security levels, which relate to the strength of a potential attacker. Also, it contains requirements regarding security of devices and the lifecycle management of their security credentials in operative environments.

The NIST Cybersecurity Framework (CSF) 2.0 [7] provides general guidance on managing cybersecurity risk along the operation, including the identification of risks, the detection of potential attacks, but also the recovery to addresses resilience for normal and adverse situations.

III. Onboarding – Overview and Approaches

Device onboarding is the process to introduce devices into a target domain and to bring them into an operational state. This process has direct relation to cybersecurity, as it includes the establishment of trust between the domain and the device in the first step. There may be situations in which it is also required to ensure that a device is operated in fact in its intended target environment. Approaches that do not involve domain verification, are often called "Trust-On-First-Use" (TOFU), as they rely on the identification information of the device only. Other approaches that support explicit trust establishment may be understood as mutually trusted bootstrapping.

Key for the trust establishment are identities and corresponding cryptographic key material and parameters, which are imprinted into devices during product manufacturing. Identity information of a device is provided, along the supply chain as shown in Figure 2 to ensure that the interaction is always done with the intended device. This identity is issued by the manufacturer together with cryptographic information, as X.509 certificate [8] and known as Initial Device Identifier (IDevID). This imprinted identity typically will not change during the device's lifetime. Nevertheless, due to advances in quantum computing, currently used asymmetric cryptographic algorithms like RSA (Rivest, Shamir, Adleman) or ECDSA (Elliptic Curve Digital Signature Algorithm), which are used to bind the identity to a cryptographic credential, i.e., the X.509 certificate, are endangered [9]. This may require that also IDevIDs can be updated in the future to ensure secure identification and authentication during onboarding specifically for long-lived devices.

In the target domain, the IDevID can be used to bootstrap mutual trust in an automated way and to support issuing domain-related identities and associated cryptographic keys, known as Locally significant Device Identifiers (LDevID), which are used as operational credentials. The reason to switch from manufacturer issued IDevIDs to operator issued LDevIDs relates to the maintenance of and complete reliance on operational credentials.

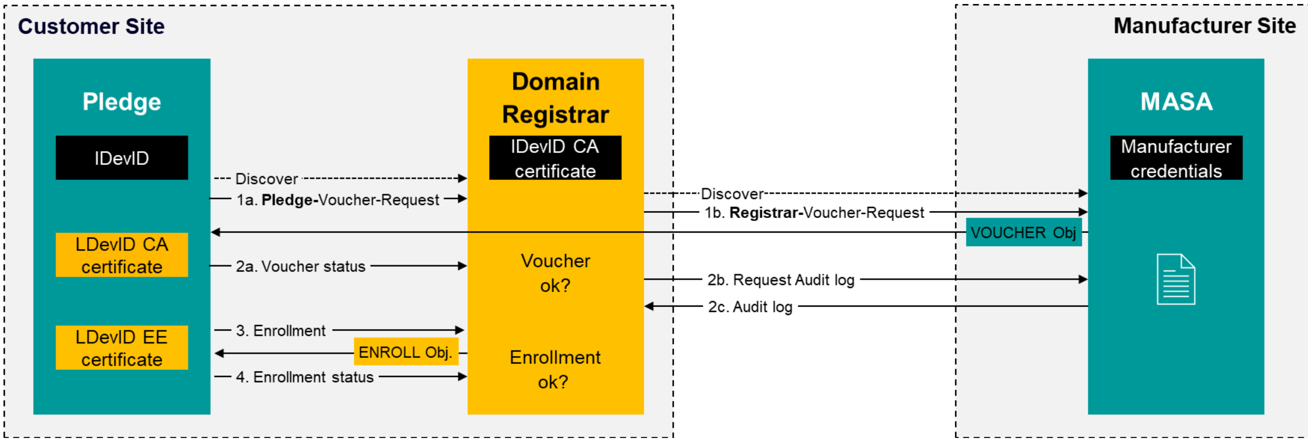


Figure 3. Onboarding Example: - Bootstrapping Remote Secure Key Infrastructure [10]

Manufacturer-issued credentials should not be used beyond bootstrapping. While IDevIDs have a longer, sometimes even undetermined lifetime, LDevIDs are updated more regularly and are under control of the operator, responsible for the security of his operational environment.

Based on the established trust relations and credentials, further operational data, like service-related configuration and engineering information including security parameters, can be provisioned on the device. To perform this comprehensive step, several technical approaches for onboarding have been developed, and furthers are likely to appear.

Several variants and approaches for supporting mutually trusted onboarding have been standardized. They provide similar functionality in terms of onboarding a component into an operational environment but differ in the respective interaction model. This relates specifically to the involvement of different service actors in the onboarding process, like the manufacturer. While some solutions are intended independent from the later application, others are part of an application framework. The following overview provides examples for the different cases:

- Bootstrapping Remote Secure Key Infrastructure (BRSKI, [10]), as shown in Figure 3, provides a standardized way to establish a mutually trusted relation between a new device (also called pledge) and a customer site network. It is supported by a manufacturer service known as Manufacturing Authorized Signing Authority (MASA) based on a voucher object for trust establishment. After discovery of the domain registrar, the pledge requests a voucher from its MASA via the domain registrar. The corresponding MASA is identified using the so-called MASA-URI extension, which is part of the IDevID certificate of the pledge. The voucher is a signed statement containing a trust anchor (as the "pinned-domain-cert") used to allow the pledge to verify the domain registrars certificate. During the onboarding procedure, the pledge voucher request (PVR) undergoes some intermediate processing by the domain registrar, in the target domain. The original voucher request from the pledge (PVR) is wrapped into a new registrar voucher request (RVR), which contains further information about the domain. The requests allow the MASA to verify it is issuing a voucher to a device produced by that manufacturer and that it has a certain trust relation to the target operative domain. Once trust has been established, domain specific security credentials (LDevID) can be enrolled to the new device. The LDevID credentials make the device a member of the domain and can be used to secure the further system interaction. The enrollment utilizes Enrollment over Secure Transport (EST) [11] for certificate management. Enhancements to BRSKI exist, supporting alternative enrollment protocols as BRSKI-AE [12] using the Lightweight Profile LCMPP [13] of the Certificate Management Protocol (CMP) [14]). Further enhancements support scenarios in which the joining device acts as server, rather than as a client (BRSKI-PRM, [15]). It needs to be triggered for interaction rather initiating the discovery of domain components upon boot. Even further variations exist which take more constraint setups into account (cBRSKI, [16]). cBRSKI uses more compact encoding with the Concise Binary Object Representation (CBOR) instead of the JavaScript Object Notation (JSON) encoding and CoAP-over-DTLS instead of HTTP-over-TLS.
- Secure Zero Touch Provisioning Protocol (SZTP) [17] specifies a further onboarding approach employing a so-called ownership voucher, which accompanies a device along its lifecycle. As in BRSKI above, the voucher is issued by a MASA. SZTP supports mutual trust establishment and enrollment of domain specific credentials and further operational information is supported by a bootstrapping server. This SZTP defined component may provide operational information directly to the new device or provide redirect information allowing to incorporate already existing services in the operational environment.
- FIDO Device Onboarding (FDO) [18] enables building a trust relation of a device to a new owner, based on trust in

the previous owner, also supported by an ownership voucher. As the manufacturer is only involved at the beginning, the interaction with the voucher is facilitated by a so-called rendezvous server instead of a service of the manufacturer as in BRSKI and SZTP. This server provides the rendezvous point between the device and the onboarding service in the new owner's domain allowing to perform a mutual authentication between the device and the new owner, based on the ownership voucher and attestation information from the device.

- OPC UA Device Onboarding specified in the OPC UA specification Part 21 [19] provides mechanisms for verifying the authenticity of devices to be onboarded and to set up their security configuration as part of the overall OPC-UA framework. It uses so-called tickets, which are similar to vouchers used in BRSKI. As BRSKI, also OPC-UA includes manufacturer specific information in the IDevID certificate as Product-Instance-URI.

As stated above, part of the onboarding is typically the enrollment of operational certificates to allow for domain-specific identification and authentication of new devices. As for onboarding, a variety of approaches exist also for enrollment. Two of them, EST and CMP, have already been stated above.

In addition to pure onboarding or provisioning standards, further standards support the propagation of security-relevant data. Specifically for the enrollment as part of the onboarding, certificate transparency [20] is known that provides an extension to PKI services for publicly logging issued certificates. As seen in the onboarding examples outlined before, certificates play a crucial role during onboarding but also during operation as they are used to identify and authenticate operational devices. This makes trust in the issuer even more important. Certificate transparency allows to identify certificates that have been issued inappropriately. Based on this information, potential impersonation attacks using unauthorized issued certificates can be detected. This underlines that logging information about issued security relevant parameters and procedures supports the root cause analysis in failure situation. The following section will outline an approach to providing enhanced information, which can be used for decision support and actually used onboarding techniques with the goal to have transparency that in turn can further support root cause analysis.

IV. ONBOARDING TRANSPARENCY ENHANCEMENTS

As discussed in Section III, several onboarding approaches are known. It is very likely that a device may only support a single or some few technical onboarding approaches, while the infrastructure likely supports multiple approaches. This will ensure that devices can be easily integrated in environments even if they originate from different manufacturers and support different onboarding and provisioning standards. To select the appropriate onboarding approach at the earliest point in time, the device-supported technical onboarding approach may be contained in the IDevID certificate, which can be analyzed by the first network component during network attachment. While standards like BRSKI or OPC-UA provide information from which the onboarding approach can be implicitly derived, the proposal here targets explicit information provisioning of the actual supported onboarding technique.

As the IDevID certificate is essentially an X.509 certificate, it can include so called extensions. An `extension` is added as certificate component similar to other certificate components like the `subject` or the `issuer`. If the extension is known to the relying party, it can be verified by the relying party. It is also possible to enforce the verification of such an extension by marking it as `critical`, which enforces the verification. If a relying party would not support the extension, it would not be allowed to further process the certificate. As the intention is here to support the onboarding in operational environments, which want to support transparency, but not to block usage in others, the extension is not marked as `critical`.

To provide information about supported onboarding and provisioning approaches, a new X.509 certificate extension is defined as shown in Figure 4.

```
supportedProvisioningMethods EXTENSION ::= {
  SYNTAX SupportedProvisioningMethods
  IDENTIFIED BY id-ce-SupportedProvisioningMethods }

SupportedProvisioningMethods ::= ProvisioningDescription
{{ ProvisioningMethod }}

ProvisioningMethod::= SEQUENCE {
  provisioningMethod     Name,
  provisiningId          OBJECT IDENTIFIER OPTIONAL,
  provisioningVersion    integer OPTIONAL
}

ProvisioningMethod ::= {CMP, SCEP, EST, CMC, ACME, FDO,
OMA-DM, OPC-UA-P21, BRSKI, SZTP, …}
```

Figure 4. Proposed Provisioning Certificate Extension

Out of the listed `ProvisioningMethod`, a device may support one or multiple options. As an example, a device with an IDevID certificate containing the information `ProvisioningMethod ::= {EST, BRSKI}` provides the information that it supports BRSKI for onboarding and EST for certificate management. The proposed enhancement is independent of the specific chosen onboarding method as it relies only on the X.509 certificate utilized to carry the onboarding transparency information. This onboarding transparency information may then be used as following.

A target network infrastructure may be designed in a way to have different virtual LANs (VLAN) defined for different onboarding mechanisms, to keep new devices contained within a separate network zone until they have received their LDevID. A motivation for this separation can be argued by different security properties of the onboarding mechanisms. As described in Section III, onboarding may be done based on TOFU, unilateral authentication of the device, or based on mutual authentication and trust establishment between the device and the domain. In case of a security breach, it may be desired to verify, how certain devices have been introduced into the operational environment and have established mutual trust to better find the root cause of a security problem.

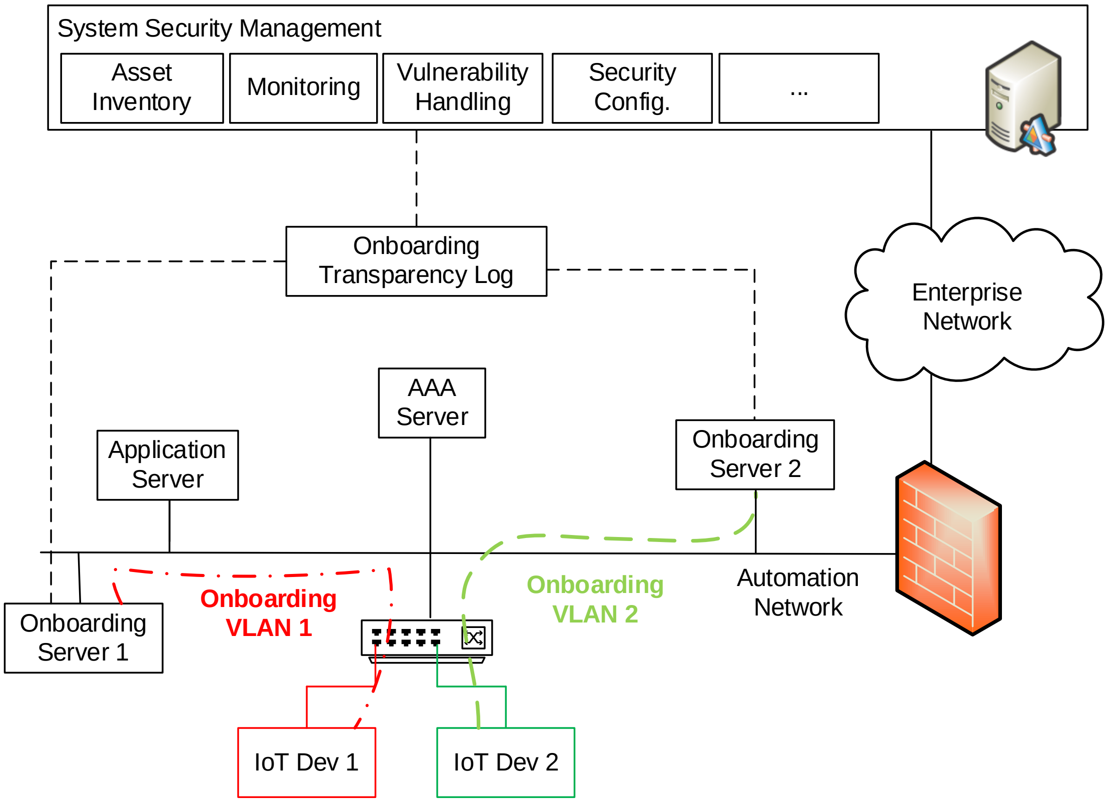


Figure 5. Onboarding Decision Support and Onboarding Transparency.

The proposed extension provides exactly this information, which can be utilized for auditing.

During the onboarding process, if the IDevID carries the extension with the onboarding and provisioning information, the device can be assigned to the appropriate VLAN based on its supported provisioning methods. This is depicted in Figure 5 above.

The figure shows an example with two devices (IoT Dev 1, IoT Dev 2). Depending on the provisioning methods supported by the respective device, they are connected by the network access switch to the onboarding VLAN1 (for local onboarding, e.g., OPC-UA-P21) or to VLAN2 (for infrastructure-based onboarding, e.g., BRSKI).

The evaluation of the supported onboarding and provisioning methods and the decision is made in the example by the AAA server to which the IoT device authenticates itself during network access. This enables the AAA server to select a specific onboarding and provisioning method, if the IoT device supports different approaches. Thus, it is possible for the AAA server to provide information on the provisioning method to be used by the device based on the assigned VLAN. Note that this may require a specific naming of the VLAN to reveal the expected onboarding mechanism to be used. This has the advantage that the device does not have to try several provisioning methods to determine the one supported by the operational network and that the device can continue to temporarily block other provisioning methods so that they cannot be misused. As a sidenote, it is expected that specifically in the case of constraint devices a device will only support a single onboarding and provisioning mechanism, while the operational infrastructure is considered more capable and to support multiple mechanisms.

While the proposed method eases the automated assignment of devices to the correct onboarding VLANs, the finally chosen onboarding variant should be logged in an onboarding transparency service. This is specifically helpful in case of security breaches, as the root cause may be related to the method how the device has been introduced into the network.

The information about onboarding may be provided as data structure encoded in different formats like XML or JSON and is ideally signed by the onboarding server. The onboarding transparency log can then verify the signature either directly or in case of a security breach. The data structure may contain different sets of information like

- Device identification (e.g., product serial number, fingerprint of the IDevID certificate of the device or the IDevID certificate directly)
- Time stamp of the actual onboarding
- Voucher issued during the onboarding. The voucher shows which device from which manufacturer was put into operation in which target (sub-)domain.

- Number of successful onboarding processes: Information on the history of the device can be provided, e.g., how often the device has already been put into operation in other domains.
- Issued LDevID certificate for the device (or a fingerprint of the LDevID certificate). This information can also be linked to the known approach of Certificate Transparency [20].

As stated, the information may be helpful in performing root cause analysis in case of discovered anomalies in an operational network. As shown in Figure 5, this information may be queried by an overall system security management and correlated to further information from monitoring, asset management or vulnerability databases.

V. Evaluation

This section gives a preliminary evaluation of the presented concept regarding derived duties for the involved parties and components.

Device manufacturer perspective: It is assumed that a manufacturer is able to imprint IDevID certificates to devices during production. Either an own Public Key Infrastructure (PKI) or PKI services of third-party providers can be used. To support the proposed extension, issued IDevID certificates need to be extended to encode the device's onboarding capabilities. This may require an information exchange between the manufacturing site and a device database containing information to prepare for later onboarding operations.

End device implementation perspective: Besides possessing an IDevID certificate including the onboarding extension, a device may need to be configurable with a VLAN identifier to be used for onboarding to support deployments where operators use a dedicated VLAN for onboarding. Alternatively, a default VLAN can be used for the onboarding network as outlined in [21]. Devices supporting multiple onboarding mechanisms may try to perform onboarding using one of the supported approaches by discovering onboarding components in the network as specified in [22] for the different variants of BRSKI.

Domain operator network attachment perspective: The AAA server of the operator's domain (given the example in Figure 5) should be able to inspect and validate the contained certificate extension during network attachment, either directly or via a service for certificate validation, to assign a specific VLAN for device onboarding and provisioning if desired. Alternatively, the AAA server itself may act as provisioning server and signal the onboarding variant.

Domain operator onboarding server perspective: The onboarding and provisioning server may support multiple different onboarding mechanisms. An operator should support a discovery mechanism to allow devices to discover the onboarding server without additional configuration. The onboarding techniques described in Section III support this discovery in their specification already. In addition, as for BRSKI several variants are specified, [22] provides a solution approach to discover the specific BRSKI variants supported by the infrastructure.

Domain operator system security management perspective: If onboarding transparency is supported in the operator's domain, the information of the chosen onboarding and provisioning mechanism needs to be kept in either the onboarding server or directly in the system security management. An operator may also choose to store this information in its asset management database containing further details of the utilized components in his operational network. It allows verifying how and when a certain device has been onboarded within the operator domain, so that this information can be used for device security purposes.

Engineering perspective: Leveraging the onboarding transparency extension may require the setup of different VLANs for the intended onboarding mechanisms (given the example in Figure 5). If different VLANs are used, the naming should be done accordingly to allow a device to utilize this information to select the associated mechanism. Alternatively, devices may use discovery functions to detect if the domain supports an onboarding server matching their technical capabilities.

VI. Conclusion and Outlook

This paper provides an overview on onboarding and provisioning as part of introducing devices into a network and to provision the devices with information to securely communicate with other devices. This is done from a requirements point of view by investigating regulative requirements as well as motivating the functionality from a general viewpoint to support root cause analysis in case of security breaches. Moreover, different standardized technical approaches have been investigated to underline the variety of possible onboarding approaches. In addition, the paper proposes enhancements to currently known approaches and processes to leverage information about supported onboarding and provisioning methods of new devices, as well as the finally chosen onboarding approach during introduction into the operational network.

A main contribution of this paper is the usage of the onboarding method information to perform access decisions as well as in the aftermath of a security event, e.g., if the device or the network was compromised. The onboarding information may support system security management to identify, which network element caused the breach, which in turn can be used to provide a fast remediation.

While the described approach has been investigated from a conceptual point of view, a further evaluation about required support from the devices and also from the operational infrastructure has been included. It is planned to investigate further into a proof of concept to verify effectiveness of the proposed approach. As outlined in the evaluation, such a proof of concept requires enhancements during the issuing of IDevIDs and LDevIDs to include the supported and chosen onboarding method in the extension of the utilized X.509 certificates. Moreover, it also requires enhancements in the evaluation of the additional onboarding information during security decisions in the operational phase and the consideration in potential post-event analysis.

ACKNOWLEDGEMENT

We would like to thank Thomas Werner for his thoughtful review and comments particularly to the discussed onboarding technologies.

REFERENCES

[1] S. Fries and R. Falk, "Device Onboarding Transparency – Supporting Initial Trust Establishment", International Conference on Emerging Security Information, Systems and Technologies, November 3 to 7, 2024, Nice, France, pp. 47-51, 2024. [Online]. Available from: https://www.thinkmind.org/library/SECURWARE/SECURWARE_2024/securware_2024_2_40_30026.html, [retrieved: February, 2025]

[2] "Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union", Document 02022L2555-20221227, Dec. 2022, [Online]. Available from: https://eur-lex.europa.eu/eli/dir/2022/2555, [retrieved: February, 2025]

[3] "Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC Text with EEA relevance", Nov. 2023, [Online]. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0053 [retrieved: February, 2025]

[4] "Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act)", , Document 32024R2847, Nov. 2024, [Online]. Available from: https://eur-lex.europa.eu/eli/reg/2024/2847/oj [retrieved: February, 2025]

[5] "Executive Order 14028: Improving the Nation's Cybersecurity", May 2017, [Online]. Available from https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity [retrieved: February, 2025]

[6] IEC 62443, "Industrial Automation and Control System Security" (formerly ISA99), [Online]. Available from: http://isa99.isa.org/Documents/Forms/AllItems.aspx [retrieved: February, 2025]

[7] NIST CSF, "The NIST Cybersecurity Framework (CSF) 2.0", Feb. 2024, [Online]. Available from: https://doi.org/10.6028/NIST.CSWP.29 [retrieved: February, 2025]

[8] ITU-T X.509 ISO/IEC 9594-8:2020, Rec. ITU-T X.509 (2019), Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks, [Online]. Available from: https://www.itu.int/rec/T-REC-X.509-201910-I/en, [retrieved: February, 2025]

[9] S. Fries and R. Falk, "Supporting Cryptographic Algorithm Agility with Attribute Certificates", International Journal on Advances in Security, Vol 17, No 1&2, 2024, pp. 92-98. [Online]. Available from: https://www.iariajournals.org/security/sec_v17_n12_2024_paged.pdf, [retrieved: February, 2025]

[10] M. Pritikin, M. Richardson, T. Eckert, M. Behringer, and K. Watson, IETF RFC 8995, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", May 2021, [Online]. Available from: https://datatracker.ietf.org/doc/html/rfc8995, [retrieved: February, 2025]

[11] M. Pritikin, P. Yee, and D. Harkins, IETF RFC 7030, "Enrollment over Secure Transport", October 2013, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc7030, [retrieved: February, 2025]

[12] D. von Oheimb, H. Brockhaus, and S. Fries IETF Draft, "Alternative Enrollment Protocols in BRSKI (BRSKI-AE)", Work in Progress, [Online]. Available from: https://datatracker.ietf.org/doc/draft-ietf-anima-brski-ae/, [retrieved: February, 2025]

[13] H. Brockhaus, D. von Oheimb, and S. Fries IETF RFC 9483, "Lightweight Certificate Management Protocol (CMP) Profile", November 2023, [Online]. Available from: https://datatracker.ietf.org/doc/html/rfc9483, [retrieved: February, 2025]

[14] C. Adams, S. Farrell, T. Krause, and T. Mononen, IETF RFC 4210, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", September 2005, [Online]. Available from https://datatracker.ietf.org/doc/html/rfc4210, [retrieved: February, 2025]

[15] S. Fries, T. Werner, E. Lear, and M. Richardson., IETF Draft, "BRSKI with Pledge in Responder Mode (BRSKI-PRM)", Work in Progress, [Online]. Available from: https://datatracker.ietf.org/doc/draft-ietf-anima-brski-prm/, [retrieved: February, 2025]

[16] M. Richardson, P. van der Stok, P. Kampanakis, and E. Dijk, IETF Draft "Constrained Bootstrapping Remote Secure Key Infrastructure (cBRSKI)", Work in Progress, [Online]. Available from: https://datatracker.ietf.org/doc/draft-ietf-anima-constrained-voucher/, [retrieved: February, 2025]

[17] K. Watsen, M. Abrahamsson, and I. Farrer, IETF RFC 8572, "Secure Zero Touch Provisioning (SZTP)", June 2021, [Online]. Available from: https://datatracker.ietf.org/doc/rfc8572, [retrieved: February, 2025]

[18] FIDO Device Onboarding, [Online]. Available from https://fidoalliance.org/device-onboarding-overview/, [retrieved: February, 2025]

[19] OPC Foundation, "OPC 10000-21: UA Part 21: Device Onboarding", Nov. 2022, [Online]. Available from: https://reference.opcfoundation.org/Onboarding/v105/docs/, [retrieved: February, 2025]

[20] B. Laurie, E. Messeri, and R. Stradling, IETF RFC 9162, "Certificate Transparency Version 2.0" Dec. 2021, [Online]. Available from: https://datatracker.ietf.org/doc/html/rfc9162, [retrieved: February, 2025]

[21] A. Dekok and M. Richardson, IETF Draft "EAP defaults for devices that need to onboard", Work in Progress, [Online]. Available from: https://datatracker.ietf.org/doc/draft-richardson-emu-eap-onboarding/, [retrieved: February, 2025]

[22] T. Eckert and E. Dijk, "BRSKI discovery and variations", Work in Progress, [Online]. Available from: https://datatracker.ietf.org/doc/draft-ietf-anima-brski-discovery/, [retrieved: February, 2025]

Extended Analysis, Detection and Attribution of Steganographic Embedding Methods in Network Data of Industrial Controls Systems

Tom Neubert, Eric Schueler, Henning Ullrich, Laura Buxhoidt, Claus Vielhauer
Department of Computer Science and Media
Brandenburg University of Applied Sciences
Brandenburg, Germany
surname.lastname@th-brandenburg.de

***Abstract*—Since the last decade, it is well known that Industrial Control Systems (ICS) are under attack and attackers nowadays increasingly use stealthy malware (i.e., stegomalware) implemented by steganographic embedding methods to in- and exfiltrate hidden information. Unfortunately, current mechanisms to distinguish between network steganographic embedding methods and embedded message types need improvement for a potential attribution of attackers. For the analysis of steganographic embedding methods which are utilized in stealthy malware, the work presented in this paper builds upon a state-of-the-art analysis testbed proposed earlier, which is recapitulated here. It offers the opportunity to analyze network steganographic embedding methods in ICS to elaborate methods to detect and distinguish between them to gain forensic information for attribution of potential attackers and their methods. In this work, we introduce a novel machine learning based approach to distinguish between five selected embedding methods and two embedded message types. We use the analysis testbed to evaluate and determine the accuracy of the novel approach compared to a state-of-the-art approach. In our extensive evaluation, our novel approach has shown to be able to distinguish between network steganographic embedding methods with an average accuracy of 85.7%, which is an improvement in comparison to the state-of-the-art by +5.9% and enables a more accurate attribution of attackers. Additionally, the novel approach is able to improve the accuracy of distinction between embedding method and embedded message type by +9.3% in comparison to the evaluated state-of-the-art approach.**

Keywords-Information Hiding; Intrusion Detection and Attribution; Network Steganography; Stealthy Malware; Industrial Control Systems

I. INTRODUCTION

This paper is based on the conference publication in [1] and significantly extends it. Some formulations and explanations are taken directly from [1].

During the last decade, stealthy malware based on steganographic embedding techniques (i.e., information hiding techniques) is increasingly used by attackers, confirmed by recent attack vectors in [2], which show that attackers use information hiding techniques to stay undetected. Stealthy malware uses completely unobtrusive data to create hidden channels, which for example are utilized to embed malicious code or to command and control. Since the Stuxnet-Attack in 2010, it has been clear that Industrial Control Systems (ICS) are under attack with stealthy malware. In this attack, Ink-files were utilized as cover data and in-memory code injections were used to hide the attack [3]. Additionally, recent attacks like the Ukrainian [4] and the Indian power grid attack [5] demonstrate that attacks with information hiding based malware on ICS become more and more common, especially due to the motivation to stay undetected as long as possible in order to in- and exfiltrate stealthy data.

Currently, several potential information hiding attack vectors for stealthy malware with steganographic embedding techniques and potential defense mechanisms are introduced (e.g., in [6], [7], [8] and [9]).

In our earlier work [1], we presented an Analysis Testbed for Steganographic Network Data (ATSND), which enables the opportunity for comprehensive analysis and comparison of these methods to identify potential similarities, differences, and effects of the embedding methods on the cover data and to derive defense and detection mechanisms for specific embedding methods. The evaluation results of [1] show that it is possible to distinguish between analyzed embedding methods after a detection, which can lead to the opportunity to identify the context of potential attackers (attribution) with machine learning based methods.

The accuracy of the state-of-the-art approach in [1] to distinguish between embedding methods is decent, but needs improvement for a more reliable attribution. Furthermore, the approach was evaluated to distinguish between a limited number of three embedding methods, which should be extended for a more conclusive evaluation and to derive a more reliable assumption about the separation precision of an approach. Additionally, the results from [1] show that the detection of embedded types (e.g., strings consisting of invariant single characters vs. text messages consisting of heterogeneous combinations of characters) needs improvement.

Thus, the **contribution** of this paper is a significant extension of the work presented in [1] and can be summarized as follows:

- Introduction of a novel feature space to train a novel neural network driven classification model for the distinction between steganographic embedding methods and embedded message types.
- Comparison between the classification results of novel feature space and the state-of-the-art feature space from [1] to derive an assumption about a potential improvement of classification accuracy.
- Extension of evaluation by two (one novel, one from state-of-the-art) to a total of now five steganographic embedding methods and novel (extended) training and test data for more meaningful evaluation results.

In the evaluation, we analyze if there is an opportunity to distinguish between five steganographic embedding methods and if we are able to differentiate between embedded message types (invariant and heterogeneous messages) with a machine learning driven classification based on our novel handcrafted feature space in comparison to a state-of-the-art feature space.

The paper is structured as follows: In Section II, we present related work and fundamentals. In Section III, we deploy our ATSND to our specific use case. Our evaluation setup to analyze five embedding methods with ATSND, including evaluation goals, data and environment, is presented in Section IV. Section V presents the evaluation results, and Section VI concludes the paper with a summary and future work.

II. Fundamentals and Related Work

In this section, we summarize fundamentals of network steganography in ICS, describe recent steganographic attack vectors for network steganography in ICS, and present our previously introduced synthetic steganographic embedding (SSE) concept to produce synthetic steganographic network data for a fast and easy generation of network data with recent steganographic embedding methods. Furthermore, an overview of methods to analyze steganographic network data for detection and attribution purposes is given.

A. Network Steganography in ICS

"Steganography is the art and science of concealing the existence of information transfer and storage", according to [10]. Besides the various possibilities for unobtrusive embedding, such as digital media data (images, audio, video et cetera), the subdomain network steganography targets the transfer and storage of hidden information in network communication traffic. From attackers perspective, a warden (e.g., intrusion detection system) observes the network traffic and the embedding of stealthy malware should be inconspicuous in a sense that a warden would not be able to differentiate between genuine communication and communication with steganographic embedding [6]. An embedding of hidden information with steganographic techniques can be realized, for example by manipulating the network packets payload on least significant values or by modulating time intervals between specific packets [11].

Network steganography and stealthy malware in ICS are special, due to limited channel capacity and thus the lower amount of available data for potential embedding compared to traditional Information Technology (IT) networks. Furthermore, the transmitted network packets are usually smaller in ICS since only meta-data or a few values (e.g., from sensors) are transferred per packet. Additionally, ICS specific protocols like OPC UA (Open Platform Communications Unified Architecture) [12] or Modbus-TCP [13] are often encapsulated in TCP/IP (or other transport protocols), which creates the opportunity for utilizing the data fields of the ICS specific protocols in addition to TCP/IP protocol headers. It is also not uncommon for the ICS-specific payload to be transmitted unencrypted, because ICS are often considered as closed networks and not subject to attacks in practice.

Potential network steganographic embedding patterns and a related terminology are summarized in [14]. A generic taxonomy and overview with the intention of a unified understanding of terms and their applicability for network steganographic methods can be found in [10].

B. Selected Steganographic Embedding Methods for ICS

In this section, we present four relevant exemplary attack vectors with regards to their steganographic embedding methods in ICS. These Embedding Methods (EM) are selected because all of them use timestamp modulations (i.e., timing channel) to embed hidden information, which is a plausible attack vector, since every network packet includes them. We are aware that there are alternative embedding concepts like Least Significant Bit (LSB) embeddings in sensor data fields of network packets, but in the context of this article, we focus on timestamps only, because they can be applied regardless of the category of the network communication (e.g., sensor data or other) and suggest relatively higher capacity.

The state-of-the-art EM and one novel steganographic embedding method will be presented in the following subsections They will be analyzed and compared with the analysis testbed presented in Section II-D from [1].

1) Steganographic Embedding Method 1 (EM_1): The approach presented in [6] uses packet timestamps (T_i) for embedding while utilizing a dynamic encoding approach based on the hour, minute, and second values, as well as an embedding key and an initialization vector. In the approach, low-value-digits of the timestamp are manipulated. This approach is able to hide one *ASCII*-symbol in four of the five highlighted digits of a timestamp in the coding "HH:MM:SS.f**fffff**fff", where H,M,S,f stand for digits of the hour, minute, second and fractional digits of the second of the time value respectively (Example: T_i= 10:00:00.1**23456**789). The actual embedding positions are determined using the embedding key, which determines the first digit right of the floating point for the fractional second values. Converting a sequence of *ASCII*-symbols to binary values results in a bitstream BS which is embedded in chronological order into every available packet. Due to the different modulated values of the variables involved, the encoding of the output values varies in perception. The formalized algorithm description can be found in Section III-B1.

2) Steganographic Embedding Method 2 (EM_2): A quite simple and easy to comprehend embedding method is introduced in [8]. The embedding scheme assumes an attack vector with a corrupted Programmable Logic Controller (PLC) via Supply-Chain-Attack. The PLC sends delays in the microsecond range ($\mu s_1, \mu s_2, \mu s_3$) to embed a hidden message via timing delays. This means an exemplary timestamp T_i = 10:00:00.123**456**789 is manipulated on the digit positions $\mu s_1 = 4, \mu s_2 = 5, \mu s_3 = 6$. The embedding scheme converts an *ASCII*-message into a bitstream *BS*. For embedding a bit of *BS*, timestamps in three consecutive OPC UA (server) packets are altered (T_i, T_{i+1}, T_{i+2}). To stay inconspicuous, the timestamps (T_{i+3}, T_{i+4}, T_{i+5}) of the following three OPC UA packets remain completely untouched. The approach arbitrarily chooses the digit '4' to embed bit = 0 and digit '9' to embed bit = 1. For the algorithm formalization see Section III-B2.

3) Steganographic Embedding Method 3 (EM_3): EM_3 is based on EM_2 and was introduced in [1]. EM_3 extends EM_2 with the addition of a key for a dynamic encoding and positioning of the embedding (see Section III-B3 for formalization of algorithm). EM_3 enables a more sophisticated and unobtrusive embedding, introducing dynamic cipher digits C_0 and C_1 for bit values 0 and 1, which leads to an encoding where the seed of the embedding is generated with a random number.

4) Steganographic Embedding Method 4 (EM_4): A sophisticated steganographic embedding method is introduced in [15] and was initially designed to alter transmitted sensor values in ICS. The formalization of the embedding algorithm can be found in Section III-B4. In this embedding method, each character of a message is converted into an 8 bit representation of its ASCII code c_A. Afterwards c_A is encrypted with an encryption key KE creating the encrypted character c_{AE}. Prior to embedding, four consecutive digits from a single OPC UA timestamp are transformed into a 16 bit long binary representation and the embedding takes place on the 8 least significant bits. c_{AE} is then embedded replacing the last 8 digits of the binary timestamp. The binary timestamp is transformed back into its four digit decimal representation and replaces the original (unaltered) timestamp.

5) Steganographic Embedding Method 5 (EM_5): Beyond the state-of-the-art, we present a novel steganographic embedding method EM_5 in this work. It will be described and formalized in Section III-B5.

To conclude this section, we want to align the embedding methods EM_{1-5} to the generic taxonomy for steganographic methods of [10]. EM_{1-5} can clearly be classified in the domain overlapping network and Cyber Physical System (CPS) and can be assigned to the CPS sub-taxonomy. In this subtaxonomy, the embedding methods belong to the categories *E1.2c1. CPS Random State/Value Modulation* and *E1.3c1. CPS Least Significant Bit State/Value Modulation*.

C. Synthetic Steganographic Data Generation

Diverse and heterogeneous steganographic ICS data is needed to train and evaluate potential defense mechanisms for ICS. However, each steganographic embedding needs mostly sophisticated and complex ICS setup, which is very time consuming to assemble, and in addition, it raises various security and safety issues. Because of this, the approach of [8] introduces a concept to generate artificial steganographic network data with a limited embedding pace and a specific steganographic embedding technique based on TCP-timestamps. Based on [8], an advanced Synthetic Steganographic Embedding (SSE)-concept is presented in [7]. It offers the possibility to embed hidden information everywhere in uncompromised network packet recordings with an embedding pace near real time. This makes it possible to quickly and easily generate test data for many different embedding methods for analysis. In [8], it is assumed that the most important aspects to be simulated in network traffic are:

1) the physical network including layout and components,
2) the network traffic including types of flows, directions, protocols used, typical payloads, etc., and
3) the type and characteristics of the (steganographic) hidden channel.

Both approaches simulate only the last aspect (3) of this list, the other two are directly adopted from an uncompromised recording of a physical setup. In the presented state-of-the-art ATSND (see Section II-D), the SSE-concept from [7] is used to generate the steganographic data based on the selected steganographic embedding methods and will be described in more detail.

D. Analysis Testbed for Steganographic Network Data (ATSND)

The Analysis Testbed for Steganographic Network Data (ATSND), as originally proposed in [1], has the purpose to compare and evaluate different (network) steganographic embedding methods to offer the possibility to make a distinction between them for a potential determination or classification of attackers or embedded message types. It includes five phases:

- Phase 1 (P_1): recording of cover-data,
- Phase 2 (P_2): selection and formalization of methods,
- Phase 3 (P_3): generation of synthetic steganographic data,
- Phase 4 (P_4): selection and extraction of features and
- Phase 5 (P_5): analysis based on the features.

The phases of ATSND are recapitulated in the following subsections and visualized in Figure 1.

1) Phase 1 of ATSND (P_1): The analysis testbed begins with Phase 1 where Cover Data (CD) has to be recorded from an uncompromised laboratory ICS network setup. CD can be recorded with different hard- and software capturing tools (e.g., *Wireshark* [16]). The output file of the recording should be extracted in the *pcap* or *pcapng* file format for further processing, since these formats are well suited logging protocols for the structural recording of network data. The recording should only contain relevant traffic for a specific purpose. The cover data builds a comparative baseline of the ICS network data to illustrate the impact of the embedding by means of a comparative analysis before and after the embedding. Further, it is also the basis for the steganographic embedding with the selected embedding methods (see Phase 2) to generate the steganographic network data in Phase 3. The specific experimental setup of our laboratory ICS is described in Section IV-B.

2) Phase 2 of ATSND (P_2): Once a network cover data file is recorded, embedding methods for the analysis in Phase 5 have to be selected and should be formalized with a pseudo code representation for an uniform, comparable and comprehensible illustration. In this work, we select four embedding approaches from state-of-the-art and introduce one novel embedding method. The formalization of the embedding methods is presented in Section III.2.

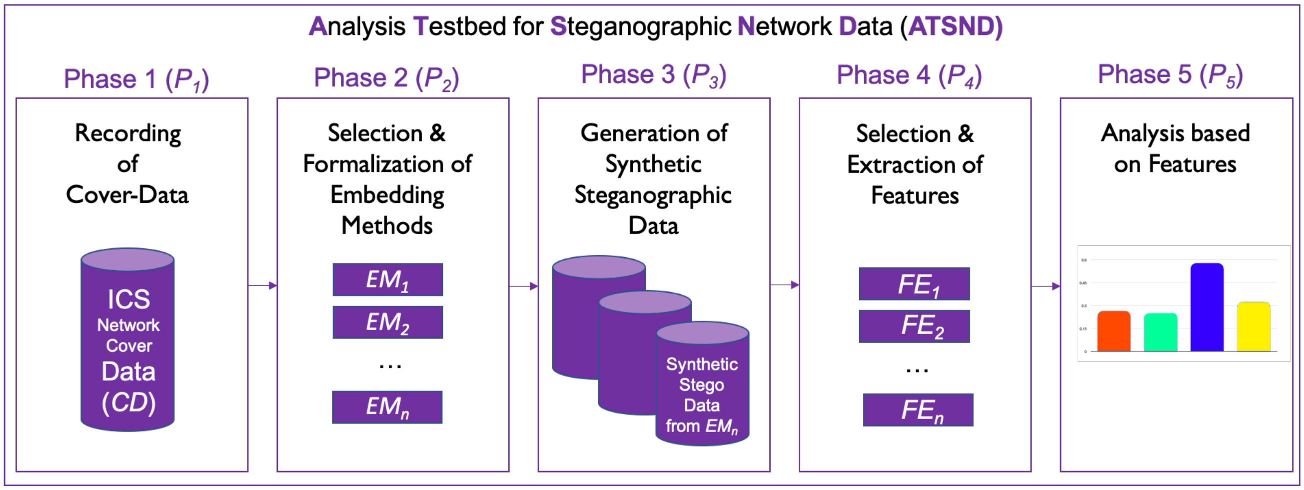


Figure 1. Analysis Testbed for Steganographic Network Data (ATSND) from [1]

3) Phase 3 of ATSND (P_3): For the creation and generation of the steganographic network data based on the embedding methods from Phase 2 (EM_1, EM_2, and EM_3), the SSE-concept [7] (introduced in Section II-C) is used. As mentioned, the SSE-concept offers the possibility to generate steganographic network data synthetically, and this results in some obvious advantages for the analysis testbed: no matter which embedding method is analyzed, it is not required to physically incorporate a corrupted, complex ICS setup in order to generate the steganographic network data containing hidden information. Thus, it is well suited because it delivers the opportunity for an easy and fast generation of steganographic network data without the need of a physical setup. The SSE-concept has the following four segments:

- Segment I: Record and Pre-Process Network Data,
- Segment II: Synthetic Embedding Option A (SEO$_A$),
- Segment III: Synthetic Embedding Option B (SEO$_B$), and
- Segment IV: Retrieval.

Segment I also deals with the recording of network data, thus Segment Element (SE) I.1 can be skipped for ATSND since the data capturing is completed after P_1. For the synthetic generation of steganographic network data, it offers two synthetic embedding options (Segment II: SEO$_A$ and Segment III: SEO$_B$). SEO$_A$ is a very fast and efficient embedding without accessing structural elements of a packet and SEO$_B$ delivers a more comfortable embedding with easier access to structural elements of a network packet based on json-objects. The retrieval in Segment IV is used to check if an embedding of a hidden message with a selected embedding method is successful. More details can be found in [7].

4) Phase 4 of ATSND (P_4): To extract features from pcap or pcapng files, the relevant structural elements of the relevant network packets should be converted into csv or txt data for processing afterwards. For this purpose, *Tshark* (*Wireshark* console application) [16] with the *-T fields -e field* option can be used to select data fields of network packets that are relevant for feature extraction and analysis. It is recommended to use handcrafted statistical feature spaces with as much discriminatory power as possible to analyze steganographic network data. This should lead to comprehensible and explainable analysis results allowing for forensic traceability.

5) Phase 5 of ATSND (P_5): Based on the extracted features from multiple embedding methods in P_4, a statistical analysis can be carried out. Therefore, various statistical computational techniques such as machine or deep learning based approaches can be taken into consideration based on the selected and extracted features. Thus, for the analysis, different data mining and machine learning tools or libraries, such as *WEKA* [17], *Orange* [18], *Tensorflow* [19] or *Keras* [20] are well suited to analyze differences and commonalities of embedding methods. Generally, the analysis can focus on different use case specific aspects, for example: detectability, attributability, embedding scheme, and more depending on the goals and objectives of a study.

E. Analysis of Steganographic ICS Network Data

A basic overview of potential methods to analyze and defend against stealthy malware based on network steganography is presented in [21]. In [1], a machine learning based approach is used to distinguish between steganographic embedding methods. The approach was initially introduced in [22] to detect network steganography in network recordings based on a handcrafted feature space with an accuracy of 92.9%. The approach performs a frequency analysis of occurrence for the digits 0 to 9 on selected positions on the packet timestamps. This feature space (FS_{SOTA}) is used for our evaluation and introduced in Section III-D.

III. Application of ATSND

As mentioned previously, we will use the Analysis Testbed for Steganographic Network Data (ATSND) from [1] (see Section II-D) for the analysis of five different embedding methods in this work. Therefore, we structure this section according to the five phases of the analysis testbed. In our specific use-case we want to evaluate if we are able to differentiate between five steganographic embedding methods and different message types with two machine learning based classification engines for a potential attribution of attackers based on their used steganographic embedding method EM.

A. Applying Phase 1 of ATSND (Recording of CD)

As mentioned, the first phase of the ATSND concept is dedicated to the collection of network Cover Data (CD) from a laboratory ICS setup. CD can be captured with any capturing tool, as long as the output can be provided in *pcap* or *pcapng* format. Additionally, the output file should only contain relevant traffic with a specific purpose. The *pcap* and *pcapng* file formats are well suited logging protocols for the structural recording of network data. CD builds the base for the further generation of steganographic network data in Phase 3, using the selected embedding methods from Phase 2 (see Section III-B). Furthermore, CD is used as a statistical baseline of the captured ICS network data. This way the impact of each of the embedding methods can be illustrated in detail.

In order to separate training and test data, we create two separate recordings for this work. We record the training data for 25 minutes and the test data for 8 minutes in our laboratory setup which is presented in more detail in Section IV-B. In our setup, the PLC and Gateway are connected directly by an Ethernet cable, thus stand-alone packet capturing hardware [23] is used to capture the traffic between them.

B. Applying Phase 2 of ATSND (Selection and Formalization of Embedding Methods)

In this phase, it is essential to select and formalize steganographic embedding methods that shall be analyzed. The formalization helps to improve the comprehensibility of the selected embedding methods and delivers a uniform presentation of them. As previously mentioned, we select four state-of-the-art methods presented in Section II-B and one novel method (see Section III-B5). All of the algorithms work with an Array A ($A = \{T_1, ..., T_i\}$) which contains all Timestamps T_i of network packets available for manipulation in our pseudocode representation. The specific formalizations for the state-of-the-art approaches EM_1, EM_2, EM_3, EM_4 and the novel embedding method EM_5 will be described in the following subsections.

1) Formalization of Steganographic Embedding Method EM_1: EM_1 was initially introduced in [6] and takes a dynamic encoding approach while manipulating low value digits of the OPC UA timestamp. An initialization vector I and an encoding key K are used in addition to variables taken from each timestamp to encode the hidden message m with characters c. Variables D, E, F and G (meaning: see Figure 2) are all derived directly from the timestamp, as well as H ($H = \{H_0, ..., H_3\}$), which is the 4-digit field in which the encoded message characters c_E are embedded. After the encoding process is finished, the output of S decides the embedding position in H.

2) Formalization of Steganographic Embedding Method EM_2: Iterating through A, EM_2 embeds a bit of the input bitstream into 3 consecutive timestamps, encoding 0 and 1 by the digital values of 4 and 9, respectively. In the process, three different digits are used for the embedding represented

Algorithm 1 Steganographic Embedding Method EM_1

```
AM ← A
i ← 0
K ← 4 Digit Key
I ← 4 Digit Initialization Vector
for c in m do
    while i < Length(A) do
        D ← Hour value of T_i
        E ← Minute value of T_i
        F ← Second value of T_i
        G ← Value of digit 1 after floating point of T_i
        H ← Value of digit 2-6 after floating point of T_i
        S ← G ⊕ DigitSum(K) mod 2
        O ← D × E × F mod 10000
        K' ← Σ_{n=0}^{3} ((K_n ⊕ (G + I_n)) mod 10) × 10^n
        K'' ← O ⊕ K' mod 10000
        c_E ← c ⊕ K'' mod 8192
        if S == 0 then
            H_0, H_1, ..., H_3 ← c_E
        else if S == 1 then
            H_1, H_2, ..., H_4 ← c_E
        end if
        AM[i] ← T_i
        i += 1
    end while
end for
```

Figure 2. Formalized Algorithm for EM_1.

in $\mu_1 - \mu_3$. Manipulated timestamps are then saved in the AM array. This is repeated for each bit in the bitstream until the end of A is reached or all bits are embedded. The algorithm was introduced in [8] and is represented in Figure 3.

Algorithm 2 Steganographic Embedding Method EM_2

```
AM ← A
for Bit in Bitstream do
    for i ← 1 to 3 do
        if Bit_i is 0 then
            T_i[μ_{i mod 3}] ← 4
        else if Bit_i is 1 then
            T_i[μ_{i mod 3}] ← 9
        end if
        AM[i] ← T_i
    end for
end for
```

Figure 3. Formalized Algorithm for EM_2.

3) Formalization of Steganographic Embedding Method EM_3: Basically, EM_3 is an advanced and more sophisticated version of EM_2 and was introduced in [1]. It should be more challenging to detect and to attribute EM_3 in comparison to EM_2. The main difference is the key-based generation of embedding symbols (digits) C_0 and C_1, as well as the key-based variation of the embedding position j within the timestamp. The algorithm is formalized in Figure 4.

Algorithm 3 Steganographic Embedding Method EM_3

$AM \leftarrow A$
$i \leftarrow 0$
$K \leftarrow$ "$SyntheticStegoKey$"
for Bit in $Bitstream$ **do**
  for $i \leftarrow 1$ to 3 **do**
    $C_0 \leftarrow 0$
    $C_1 \leftarrow 0$
    while $C_0 == C_1$ **do**
      $C_0 \leftarrow Random(K)\ mod\ 9$
      $C_1 \leftarrow Random(K)\ mod\ 9$
    end while
    $j \leftarrow C_0 + C_1\ mod\ 3$
    if Bit_i is 0 **then**
      $T_i[\mu_j] \leftarrow C_0$
    else if Bit_i is 1 **then**
      $T_i[\mu_j] \leftarrow C_1$
    end if
    $AM[i] \leftarrow T_i$
  end for
end for

Figure 4. Formalized Algorithm for EM_3.

4) Formalization of Steganographic Embedding Method EM_4: This embedding method was introduced in [15] and its formalization is presented in Figure 5. In the formalization, the variable c represents a character of the message m and c_A the 8 bit representation of the ASCII code decimal digit of the character. It is encrypted with encryption key KE and results in an 8 bit encrypted bitstream c_{AE} of the ASCII code decimal digit, which is embedded into a 16 bit representation of a converted timestamp T_i16B (into the 8 least significant bits). After embedding, T_i16B is converted back into its initial representation.

Algorithm 4 Steganographic Embedding Method EM_4

$AM \leftarrow A$
$i \leftarrow 0$
$KE \leftarrow$ "$EncryptionKey$"
for c in m **do**
  for $j \leftarrow 0$ to 3 **do**
    $c_A \leftarrow c$
    $c_{AE} \leftarrow c_A \oplus KE$
    $T_i[Length(T_i) - j]16b \leftarrow T_i[Length(T_i) - j]$
    $T_i[Length(T_i) - j]16b \leftarrow c_{AE}$
    $T_i[Length(T_i) - j] \leftarrow T_i[Length(T_i) - j]16b$
    $AM[i] \leftarrow T_i$
  end for
end for

Figure 5. Formalized Algorithm for EM_4.

5) Formalization of Steganographic Embedding Method EM_5: Steganographic embedding method EM_5 represents a novel method. EM_5 embeds a message m into the microseconds $\mu_1 - \mu_3$ of OPC UA (server) timestamps T_i (e.g., $T_i =$ 10:00:00.123**456**789, embedding positions are marked **bold**). Before embedding each character c of m, m is saved to array *MAD* as the corresponding decimal ASCII representation of its characters c. After every element of *MAD* is embedded, first 494, then 949 are embedded into the following timestamps to signal the end of m. EM_5 was chosen for evaluation since it is a more simple algorithm which should be accurate to detect and to attribute based on the limited number of ASCII characters.

Algorithm 5 Steganographic Embedding Method EM_5

$AM \leftarrow A$
$i \leftarrow 0$
$j \leftarrow 0$
while $j < Length(MAD) + 2$ **do**
  if $j < Length(MAD)$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow MDA_j$
  else if $j == Length(MAD)$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow 494$
  else if $j == Length(MAD) + 1$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow 949$
  end if
  $AM[i] \leftarrow T_i$
  i += 1
  j += 1
end while

Figure 6. Formalized Algorithm for EM_5.

C. Applying Phase 3 of ATSND (Generation of Synthetic Steganographic Data)

For the synthetic generation of steganographic network data, the introduced SSE-concept is used (see Section II-C). In the evaluation, this work uses synthetic embedding option SEO_A, since it offers a much more efficient and faster embedding to generate synthetic steganographic network data based on the manipulation of hexdump elements of the network packets. All 5 selected steganographic embedding methods EM_1, EM_2, EM_3, EM_4 and EM_5 are generated with SEO_A based on the recorded cover data CD in P_1.

D. Applying Phase 4 of ATSND (Selection and Extraction of Features)

To extract features from pcap or pcapng files, the relevant structural element of the relevant network packets should be converted into csv or txt data to process it afterwards. Therefore, *Tshark* (*Wireshark* console application) [16] with the *-T fields -e field* option can be used to select data fields of network packets that are relevant for feature extraction and analysis. We recommend the usage of handcrafted statistical feature spaces with as much discriminatory power as possible to analyze steganographic network data. This should lead to comprehensible and plausible analysis results.

In this work, we use two handcrafted feature spaces to train two separate machine learning based models for our analysis in

P_5. One feature space FS_{Legacy} is used from state-of-the-art to set a baseline for our analysis goals. Additionally, we design a novel feature space FS_{Novel} to investigate if it is possible to achieve more accurate results in our analysis. The two feature spaces are presented in the following subsections. Both feature spaces analyze the last 6 digit positions of network packet timestamps because they are well suited for steganographic embedding, since every network packet has a timestamp and a potential delay in micro- and nanosecond areas is absolutely unobtrusive. A potential attack vector for our use case could look like those introduced in Section IV-B. Both feature spaces analyze multiple network packets to extract a feature vector (i.e., sample), because obviously a single packet with steganographic embedding should look unobtrusive (if not, it would not be steganographic). A measurable or quantifiable anomaly caused by a steganographic embedding can only occur by analyzing multiple network packets. In this work, we use 100 network packets to extract a sample (i.e., feature vector with label) for the feature spaces. This length (100 packets) has been selected based on state-of-the-art ([15], [22]). The optimal length with maximum separation precision can only be determined by an explorative analysis of different lengths, which is out of the scope for this work.

1) Feature Space FS_{SOTA}: The state-of-the-art feature space was introduced in [22], which performs a frequency analysis for the digits 0 to 9 on the mentioned six last and least significant digits in network packet timestamps. Thus, 10 features (values) for each analyzed digit position between 0.0 and 1.0 representing the percentage of occurrence for each digit 0 to 9 are extracted from a sample with multiple packets (as mentioned, 100 packets used to extract a sample or i.e., feature vector). The frequency analysis results in a 60-dimensional feature space which is used to train two 'legacy' multilayer perceptrons (MLP) to potentially distinguish between embedding methods and cover data (MLP$_{6LG}$, legacy MLP with 6 classes, based on FS_{SOTA}) and to distinguish between the embedded message types and embedding methods (MLP$_{11LG}$, legacy MLP with 11 classes based on FS_{SOTA}). The selected features shall be extracted for multiple samples from all embedding methods with different message types and cover data to build MLP$_{6LG}$ and MLP$_{11LG}$ for analysis in P_5.

2) Feature Space FS_{Novel}: Our novel feature space FS_{Novel} extends FS_{SOTA}. We add additional features based on potential artifacts caused by the embeddings. This includes the standard deviation of the digit frequencies for every digit position in the millisecond and microsecond ranges. Additionally, we calculate the standard deviation across the digit standard deviations to analyze the manipulation of single digit positions. The standard deviation over only the microseconds is also used, as embedding methods EM_2 and EM_3 only use these positions for the embedding process. In addition, the standard error of the mean of the digit distribution is calculated for each position. As a further feature, the digit transition rate is used. This feature describes the percentage of packets in which the digit at a given position changes from the preceding packet. An embedding method with a high embedding density such as EM_5 might cause digits to change less frequently. Furthermore, EM_5 changes the first digit position to a low digit. Therefore, we use the average digit value for each position. Moreover, we use Pearson's chi-squared test [24] for the distribution of digits for each position. This test describes the likelihood that an observed distribution is the result of a random sample expecting a given distribution. For the milli-, micro- and nanosecond digits of a timestamp, we expect a uniform distribution. A steganographic embedding like EM_2 uses constant values which change this uniform distribution. Additionally, the skewness of the digit distribution is calculated for every position. This describes whether the distribution is weighted towards the higher or lower end of the digits. Finally, we use the kurtosis for the digit positions, which describes the steepness in a distribution. In total, this results in a 104-dimensional feature space to train two 'new' multilayer perceptrons to potentially distinguish between embedding methods and cover data (MLP$_{6NE}$, new MLP with 6 classes based on FS_{Novel}) and to distinguish between the embedded message types and embedding methods (MLP$_{11NE}$, new MLP with 11 classes based on FS_{Novel}).

E. Analysis (P_5)

For our analysis, we will investigate if it is possible to distinguish between the five selected steganographic embedding methods (EM_{1-5}) and cover data (CD) after a potential detection of an anomaly, to potentially attribute an attacker with MLP$_{6NE}$ and MLP$_{6LG}$ (6-class classification challenge). Additionally, we analyze if it is possible to distinguish between embedded message types and steganograpic embedding methods with MLP$_{11NE}$ and MLP$_{11LG}$ (11 class classification challenge). The specific evaluation goals are presented in Section IV-A.

IV. Evaluation Setup

A. Evaluation Goals

The evaluation extends the evaluation of [1] significantly and addresses the following goals:

- G_1: Determination of the classification accuracy for MLP$_{6NE}$ (new MLP based on novel feature space FS_{Novel}) and MLP$_{6LG}$ ('legacy' MLP based on state-of-the-art feature space FS_{SOTA}) to analyze if and how accurate they are able to distinguish between the five selected steganographic embedding methods (EM_{1-5}) and the cover data (CD), and to investigate if new MLP$_{6NE}$ can outscore the state-of-the-art MLP$_{6LG}$ in this 6-class-classification-challenge.
- G_2: Determination of the classification accuracy for MLP$_{11NE}$ and MLP$_{11LG}$ to analyze if and how accurate they are able to distinguish between the five selected steganographic embedding methods, the two embedded message types (invariant IV message type, which means a repeated letter and heterogeneous HE message type, which means a random text message, see Section IV-C) and the cover data, and to investigate if new MLP$_{11NE}$ can outscore the state-of-the-art MLP$_{11LG}$ in this 11-class-classification-challenge.

The classification accuracy ACC can be determined with $ACC = (\frac{CCS}{AS}) * 100$, where CCS is the number of correctly

TABLE I
NETWORK DATA SETS FOR FEATURE EXTRACTION; STEGANOGRAPHIC DATA IS EMBEDDED SYNTHETICALLY IN REC_{CD}.

Name	Type of Recording	Embedding Method	Message Type	Hidden Message	No. of relevant Packets	No. of extracted Samples
$REC_{Train-CD}$	Cover Training-Data	-	-	-	25,613	514
$REC_{Train-EM1IV}$		EM_1	invariant	‘a’ (repeated)	25,613	514
$REC_{Train-EM1HE}$		EM_1	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM2IV}$		EM_2	invariant	‘a’ (repeated)	25,613	514
$REC_{Train-EM2HE}$		EM_2	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	25,613	514
$RE_{Train-EM3IV}$	Steganographic Training-Data	EM_3	invariant	‘a’ (repeated)	25,613	514
$REC_{Train-EM3HE}$		EM_3	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM4IV}$		EM_4	invariant	‘a’ (repeated)	25,613	514
$REC_{Train-EM4HE}$		EM_4	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM5IV}$		EM_5	invariant	‘a’ (repeated)	25,613	514
$REC_{Train-EM5HE}$		EM_5	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	25,613	514
$REC_{Eval-CD}$	Cover Test-Data	-	-	-	8,703	177
$REC_{Eval-EM1IV}$		EM_1	invariant	‘a’ (repeated)	8,703	177
$REC_{Eval-EM1HE}$		EM_1	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM2IV}$		EM_2	invariant	‘a’ (repeated)	8,703	177
$REC_{Eval-EM2HE}$		EM_2	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM3IV}$	Steganographic Test-Data	EM_3	invariant	‘a’ (repeated)	8,703	177
$REC_{Eval-EM3HE}$		EM_3	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM4IV}$		EM_4	invariant	‘a’ (repeated)	8,703	177
$REC_{Eval-EM4HE}$		EM_4	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM5IV}$		EM_5	invariant	‘a’ (repeated)	8,703	177
$REC_{Eval-EM5HE}$		EM_5	heterogeneous	‘IARIA-Journal-2025 ’ + Lorem ipsum ... (until full)	8,703	177

classified samples and AS is the number of all samples in the corresponding class. The results for G_1 and G_2 are presented in Section V.

B. Attack Vector and Laboratory ICS Setup of Evaluation

The recording of the cover-data in phase 1 of ATSND is done on a Fischertechnik® Lernfabrik 4.0 24V [25] model. The modeled production line consists of 2 transportation cranes, a storage rack, an environmental sensor and multiple conveyor belts, actuators and other sensors. A Siemens S7-1500 PLC controls the actuators and sensors, and connects to another network via a gateway for remote supervision. The gateway communicates directly with the Siemens-PLC using the ICS specific OPC UA protocol. Since the gateway acts as a middleman for the remote interface, its main responsibility is to collect the data of all sensors and actuators in real time. To do this, the gateway periodically requests the values of the sensors directly from the PLC. In contrast, the real time data (current and target position) of sensors and actuators is published by the PLC in shorter intervals, but only while they are active. The setup performs a close-to-reality production process including real communication involved between all components and makes use of industrial standard controllers, thus it can be considered to produce realistic and plausible ICS network traffic.

Since the OPC UA communication between the PLC and Gateway is numerous, predictable, and outward-facing (meaning leaving the Operational Technology (OT) ICS network towards Information Technology (IT) focused domains of an infrastructure), it forms a suitable cover to exfiltrate data. The fact that the communication occurs between two separate network zones would be especially beneficial for a possible attacker. A possible goal for this exfiltration could for example be the theft of confidential process information. In this attack scenario, the attacker has to manipulate the OPC UA responses coming from the PLC. This could be achieved by corrupting the control logic on the PLC itself using a supply-chain-attack.

C. Evaluation Data Sets

The cover data recorded in Phase 1 of ATSND (see Section III-A) is the base for further generation of synthetic steganographic data. In order to prevent overfitting and evaluate the MLP externally with data it has not seen before, there are two cover data sets. The larger one (REC_{Train}) consists of 25613 relevant packets and is used for training of the MLPs. For the evaluation a smaller, disjoint data set (REC_{Eval}) consisting of 8703 relevant packets is used. Our data sets are created with the SSE-concept [7], which allows a message to be synthetically embedded into a *pcap* or *pcapng* capture file. All of the embedding methods used in this paper, are modifying the recorded cover data sets synthetically. All used embedding methods modify the last digits of the OPC UA Timestamp in a network packet as described in Section III-B. In a real world attack scenario this manipulation could be achieved by a corrupted server (e.g., PLC, via supply-chain-attack) which sends timing-delays to embed the hidden information. The steganographic taxonomy introduced in [10] would categorize the used embedding methods under the *LSB state/value modulation* category.

Since one of the goals of this paper is to see if it is possible to distinguish between invariant and heterogeneous messages, we need to define the two messages to embed. The embedded invariant message consists of the repeated letter *‘a’*. In order to represent the (character) similarity of natural text in the heterogeneous message, we chose to use the phrase *IARIA-Journal-2025*, followed by as much *Lorem Ispum* text as possible for each recording and embedding method. Table I shows a summary of all combinations of recording, embedding methods and embedded messages. For example $REC_{Eval-EM3IV}$ describes the recording based on the **Eval**uation cover, with the **I**n**V**ariant message embedded by embedding method EM_3. In the following steps, the resulting steganographic data is used to extract samples of feature vectors. These are in turn used to train and evaluate our resulting MLPs.

The data used to train our MLPs for evaluation is based on the training data set from I. For the cover data training recording, we extract 514 samples (i.e., extracted feature vectors with label). For the generated data for Goal G_1, we use the combined feature vectors of both message types for every embedding method. This means for every embedding method we have 1028 samples. For Goal G_2 we use the same cover training data, while the generated data is based only on the corresponding recording for every combination of embedding method and message type. This results in 514 samples for every training data subset. The training data setup is shown in Table II.

TABLE II
TRAINING DATA SETS USED FOR TRAINING MLP$_{6LG}$, MLP$_{6NE}$, MLP$_{11LG}$ AND MLP$_{11NE}$ FOR EVALUATION OF G_1 AND G_2.

Training Data Set Name	Label of Samples	Features for MLPs extracted from:	Number of Samples	Goal
Data Sets used to train MLP$_{6LG}$ and MLP$_{6NE}$:				
TS_{CD}	CD	$REC_{Train-CD}$	514	G_1
TS_{EM1}	EM1	$REC_{Train-EM1IV}$, $REC_{Train-EM1HE}$	1028 (2x514)	
TS_{EM2}	EM2	$REC_{Train-EM2IV}$, $REC_{Train-EM2HE}$	1028 (2x514)	
TS_{EM3}	EM3	$REC_{Train-EM3IV}$, $REC_{Train-EM3HE}$	1028 (2x514)	
TS_{EM4}	EM4	$REC_{Train-EM4IV}$, $REC_{Train-EM4HE}$	1028 (2x514)	
TS_{EM5}	EM5	$REC_{Train-EM5IV}$, $REC_{Train-EM5HE}$	1028 (2x514)	
Data Sets used to train MLP$_{11LG}$ and MLP$_{11NE}$:				
TS_{CD}	CD	$REC_{Train-CD}$	514	G_2
TS_{EM1IV}	EM1-IV	$REC_{Train-EM1IV}$	514	
TS_{EM1HE}	EM1-HE	$REC_{Train-EM1HE}$	514	
TS_{EM2IV}	EM2-IV	$REC_{Train-EM2IV}$	514	
TS_{EM2HE}	EM2-HE	$REC_{Train-EM2HE}$	514	
TS_{EM3IV}	EM3-IV	$REC_{Train-EM3IV}$	514	
TS_{EM3HE}	EM3-HE	$REC_{Train-EM3HE}$	514	
TS_{EM4IV}	EM4-IV	$REC_{Train-EM4IV}$	514	
TS_{EM4HE}	EM4-HE	$REC_{Train-EM4HE}$	514	
TS_{EM5IV}	EM5-IV	$REC_{Train-EM5IV}$	514	
TS_{EM5HE}	EM5-HE	$REC_{Train-EM5HE}$	514	

For our evaluation of the model, we use the evaluation data set from Table I. The cover data set contains only the original recordings, resulting in 177 samples. For Goal G_1 we use the combined recordings from both message types. Each recording then contains 354 samples per embedding method. The model for Goal G_2 uses the recording for every embedding method and message type separately, so every data subset contains 177 samples. The evaluation data sets can be seen in Table III.

TABLE III
TEST DATA SETS USED FOR EVALUATION TO ACHIEVE G_1 AND G_2.

Test Data Set Name	Label of Samples	Features extracted from:	Number of Samples	Goal
Data Sets used to evaluate MLP$_{6LG}$ and MLP$_{6NE}$:				
DS_{CD}	CD	$REC_{Eval-CD}$	177	G_1
DS_{EM1}	EM1	$REC_{Eval-EM1IV}$, $REC_{Eval-EM1HE}$	354 (2x177)	
DS_{EM2}	EM2	$REC_{Eval-EM2IV}$, $REC_{Eval-EM2HE}$	354 (2x177)	
DS_{EM3}	EM3	$REC_{Eval-EM3IV}$, $REC_{Eval-EM3HE}$	354 (2x177)	
DS_{EM4}	EM4	$REC_{Eval-EM4IV}$, $REC_{Eval-EM4HE}$	354 (2x177)	
DS_{EM5}	EM5	$REC_{Eval-EM5IV}$, $REC_{Eval-EM5HE}$	354 (2x177)	
Data Sets used to evaluate MLP$_{11LG}$ and MLP$_{11NE}$:				
DS_{CD}	CD	$REC_{Eval-CD}$	177	G_2
DS_{EM1IV}	EM1-IV	$REC_{Eval-EM1IV}$	177	
DS_{EM1HE}	EM1-HE	$REC_{Eval-EM1HE}$	177	
DS_{EM2IV}	EM2-IV	$REC_{Eval-EM2IV}$	177	
DS_{EM2HE}	EM2-HE	$REC_{Eval-EM2HE}$	177	
DS_{EM3IV}	EM3-IV	$REC_{Eval-EM3IV}$	177	
DS_{EM3HE}	EM3-HE	$REC_{Eval-EM3HE}$	177	
DS_{EM4IV}	EM4-IV	$REC_{Eval-EM4IV}$	177	
DS_{EM4HE}	EM4-HE	$REC_{Eval-EM4HE}$	177	
DS_{EM5IV}	EM5-IV	$REC_{Eval-EM5IV}$	177	
DS_{EM5HE}	EM5-HE	$REC_{Eval-EM5HE}$	177	

V. EVALUATION RESULTS

In this section, the determined classification results on the introduced evaluation setup for evaluation goal G_1 with MLP$_{6LP}$ and MLP$_{6NE}$ and for G_2 with MLP$_{11LP}$ and MLP$_{11NE}$ are presented.

A. Results for G_1

In G_1 we determine the classification results for the 'legacy' MLP$_{6LP}$ based on state-of-the-art feature space FS_{SOTA} and the 'new' MLP$_{6NE}$ based on novel feature space FS_{Novel}. This determination should show whether the presented machine learning based models are able to distinguish between the 5 presented steganographic embedding methods (EM_{1-5}) and cover data (CD). Additionally, we want to find out if the novel model can outperform the state-of-the-art approach.

TABLE IV
CONFUSION MATRIX OF CLASSIFICATION RESULTS ON TEST-DATA OF MLP$_{6LG}$ AND MLP$_{6NE}$ FOR G_1 (**BOLD**: CORRECTLY CLASSIFIED SAMPLES, CD = 177 SAMPLES, EM$_n$= 354 SAMPLES)

classified → Actual	CD	EM1	EM2	EM3	EM4	EM5	*ACC (rounded)*
CD	**90 | 103**	9 | 0	0 | 1	43 | 45	35 | 28	0 | 0	*51 | 58*
EM1	5 | 1	**318 | 348**	0 | 0	12 | 5	19 | 0	0 | 0	*90 | 98*
EM2	1 | 1	0 | 0	**352 | 353**	0 | 0	1 | 0	0 | 0	*99 | 99*
EM3	77 | 78	23 | 0	0 | 0	**179 | 238**	75 | 38	0 | 0	*51 | 67*
EM4	24 | 40	15 | 0	0 | 0	31 | 41	**283 | 273**	1 | 0	*80 | 77*
EM5	0 | 0	0 | 0	0 | 0	0 | 0	1 | 1	**353 | 353**	*99 | 99*
						Overall Samples:	*81 | 86*

The classification results for both MLPs are presented in Table IV. We can state that both models are basically able to distinguish correctly for a majority of test samples for all classes (classification accuracies are visualized in Figure 7).

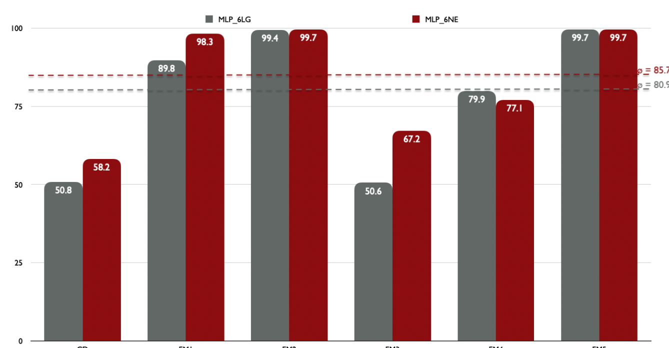


Figure 7. Classification Accuracy for MLP_{6LG} and MLP_{6NE} for each embedding method and cover data.

MLP$_{6LP}$ reaches an overall accuracy (correctly classified samples in relation to all samples) $ACC = 80.9\%$ and MLP$_{6NE}$ is significantly more accurate with $ACC = 85.7\%$. The classification accuracy can be especially improved with MLP$_{6NE}$ for steganographic embedding method EM_1 ($ACC_{MLP_{6LG}} = 89.9\%, ACC_{MLP_{6NE}} = 98.3\%$) and EM_3 ($ACC_{MLP_{6LG}} = 50.6\%, ACC_{MLP_{6NE}} = 67.2\%$), and the cover data ($ACC_{MLP_{6LG}} = 50.8\%, ACC_{MLP_{6NE}} = 58.2\%$). For EM_4 MLP$_{6LP}$ is slightly more precise in terms of classification accuracy ($ACC_{MLP_{6LG}} = 79.9\%, ACC_{MLP_{6NE}} = 77.1\%$). Both models have the same accuracy of $ACC = 99.7\%$ for EM_2 and EM_5, these methods are, as assumed,

TABLE V
CONFUSION MATRIX OF CLASSIFICATION RESULTS ON TEST-DATA OF MLP_{11LG} AND MLP_{11NE} FOR G_2 (**BOLD**: CORRECTLY CLASSIFIED SAMPLES, 177 SAMPLES PER CLASS)

<table>
<tr><th>classified as →
Actual</th><th>CD</th><th>EM1-IV</th><th>EM1-HE</th><th>EM2-IV</th><th>EM2-HE</th><th>EM3-IV</th><th>EM3-HE</th><th>EM4-IV</th><th>EM4-HE</th><th>EM5-IV</th><th>EM5-HE</th><th>ACC (rounded)</th></tr>
<tr><td>CD</td><td>70 | 78</td><td>5 | 1</td><td>11 | 0</td><td>0 | 0</td><td>0 | 0</td><td>8 | 28</td><td>51 | 31</td><td>4 | 10</td><td>28 | 29</td><td>0 | 0</td><td>0 | 0</td><td>40 | 44</td></tr>
<tr><td>EM1-IV</td><td>8 | 1</td><td>120 | 176</td><td>33 | 0</td><td>0 | 0</td><td>0 | 0</td><td>4 | 0</td><td>5 | 0</td><td>6 | 0</td><td>1 | 0</td><td>0 | 0</td><td>0 | 0</td><td>68 | 99</td></tr>
<tr><td>EM1-HE</td><td>4 | 0</td><td>22 | 0</td><td>134 | 176</td><td>0 | 0</td><td>0 | 0</td><td>8 | 0</td><td>6 | 0</td><td>1 | 1</td><td>2 | 0</td><td>0 | 0</td><td>0 | 0</td><td>76 | 99</td></tr>
<tr><td>EM2-IV</td><td>0 | 0</td><td>1 | 0</td><td>0 | 0</td><td>172 | 174</td><td>4 | 2</td><td>0 | 0</td><td>0 | 0</td><td>0 | 1</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>97 | 98</td></tr>
<tr><td>EM2-HE</td><td>3 | 3</td><td>1 | 0</td><td>0 | 0</td><td>1 | 1</td><td>170 | 171</td><td>0 | 0</td><td>0 | 0</td><td>0 | 2</td><td>2 | 0</td><td>0 | 0</td><td>0 | 0</td><td>96 | 97</td></tr>
<tr><td>EM3-IV</td><td>33 | 31</td><td>4 | 0</td><td>10 | 6</td><td>0 | 0</td><td>0 | 0</td><td>16 | 47</td><td>90 | 66</td><td>6 | 3</td><td>18 | 24</td><td>0 | 0</td><td>0 | 0</td><td>9 | 26</td></tr>
<tr><td>EM3-HE</td><td>31 | 27</td><td>5 | 1</td><td>3 | 7</td><td>0 | 0</td><td>0 | 1</td><td>15 | 42</td><td>93 | 75</td><td>4 | 2</td><td>22 | 26</td><td>0 | 0</td><td>0 | 0</td><td>53 | 42</td></tr>
<tr><td>EM4-IV</td><td>3 | 8</td><td>3 | 0</td><td>3 | 1</td><td>0 | 0</td><td>2 | 0</td><td>3 | 5</td><td>11 | 10</td><td>120 | 113</td><td>32 | 40</td><td>0 | 0</td><td>0 | 0</td><td>68 | 64</td></tr>
<tr><td>EM4-HE</td><td>19 | 25</td><td>7 | 1</td><td>8 | 2</td><td>0 | 0</td><td>0 | 0</td><td>1 | 8</td><td>20 | 21</td><td>29 | 19</td><td>92 | 101</td><td>0 | 0</td><td>0 | 0</td><td>52 | 57</td></tr>
<tr><td>EM5-IV</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>177 | 177</td><td>1 | 0</td><td>100 | 100</td></tr>
<tr><td>EM5-HE</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>0 | 0</td><td>1 | 0</td><td>176 | 177</td><td>99 | 100</td></tr>
<tr><td colspan="12" align="right">Overall Samples:</td><td>69 | 75</td></tr>
</table>

the most easiest ones to attribute correctly. Additionally, we shall notice that the accuracy for both approaches on cover data (CD) should be improved in future work, because it would trigger false positives in a real world scenario, but if we state that an attribution takes place after a previous detection (so we can exclude cover data), then especially the novel MLP_{6NE} has a decent precision to distinguish between embeddings.

B. Results for G_2

In G_2 we determine the classification results for MLP_{11LP} based on FS_{SOTA} and MLP_{11NE} based on FS_{Novel}. This determination should show if the approaches are able to distinguish between the five selected steganographic embedding methods (EM_{1-5}), the two embedded message types (IV and HE) and the cover data (CD). Additionally, we want to find out if the novel model can outperform the state-of-the-art approach.

The results for both models are shown in Table V. We can state that both models are still able to distinguish correctly between used embedding methods for a majority of test samples. Accuracy for MLP_{11LP} and MLP_{11NE} for all classes is visualized in Figure 8. Through all samples, MLP_{11LP} delivers $ACC = 68.8\%$. MLP_{11NE} delivers $ACC = 75.2\%$ overall samples and thus clearly outperforms MLP_{11LP}. The distinction between embedded message types is comparatively accurate for EM_1, EM_2 and EM_5 for MLP_{11LP}. For EM_3 the accuracy is limited, but this is explainable, due to the key-based pseudo-random embedding code generation, which makes it hard to distinguish between embedded message types.

However, on a holistic view, we can state that a distinction between embedding method and embedded message type is possible and accurate, especially with MLP_{11NE}, which is based on our novel handcrafted feature space for embedding methods with no message encryption (EM_1, EM_2 and EM_5).

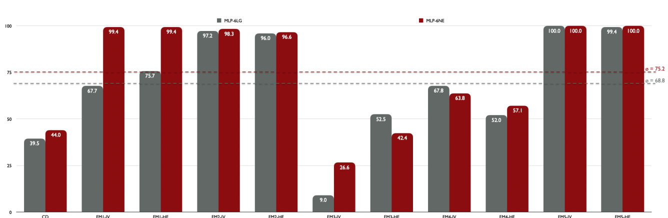


Figure 8. Classification accuracy for MLP_{11LG} and MLP_{11NE} for each message type with embedding method and cover data.

VI. CONCLUSION AND FUTURE WORK

In this paper, we analyze the possibility to distinguish between five steganographic embedding methods and two different message types based on a state-of-the-art analysis testbed for steganographic ICS network data with an extensive evaluation/analysis setup. We elaborate a novel feature space to train a machine learning driven approach with multilayer perceptron as classification engine. Our novel approach, which significantly extends a state-of-the-art-method previously presented, is able to distinguish between steganographic embedding methods with an accuracy of 85.7%, which outperforms a state-of-the-art-method by +5.9%. This creates the opportunity for a more accurate attribution, which can possibly identify the context of attackers (for example: software fingerprinting). Additionally, we are able to distinguish between steganographic embedding methods and embedded message types with an accuracy of 75.2%, which significantly improves the ability to conclude what type of message was embedded (improvement of +9.3% compared to state-of-the-art). Message type classification following a successful detection of steganographic channels may help in the attribution of different malicious payloads of stealthy malware in the future. This can be potentially achieved by differentiation between different malware code types as payload (e.g. script/shellcode vs. binary code vs. command & control instructions), deployed by different attacker groups. While steganographic communication of malware is considered to be used for illegitimate data aggregation within limited boundaries of ICS subnets, future stegomalware attacks may also make use of gateway communication, traversing borders between isolated ICS subnetworks and Information Technology (IT) network segments of the informational infrastructure of enterprises. Thus, the combination of additional forensic traces discovered on the system under attack (such as TCP/IP network traces) and steganalytic properties such as the payload type and length may allow to attribute the origin of the attack in the future for example for data in- and exfiltration via the gateway more precisely.

In future work, we would like to analyze more message types (e.g., source-code-like structures) and significantly more steganographic embedding methods. Additionally, our novel feature space has the potential to be extended for a more accurate classification. We will expand our experiments with network data from more complex ICS systems and with longer network data recordings to create a significantly larger number of samples for training and testing. Additionally, more potential classification models based on traditional and modern machine learning techniques should be trained and analyzed to potentially improve the classification performance.

ACKNOWLEDGEMENTS

The research in this work has been performed in context of the project ATTRIBUT (https://omen.cs.uni-magdeburg.de/itiamsl/deutsch/projekte/attribut.html). This comprises in particular the conceptional design of the analysis testbed for steganographic network data and embedding method EM_3, as well as software realization in Python of all embedding methods and feature extraction. The Project ATTRIBUT is supported by funding of the Agentur für Innovation in der Cybersicherheit GmbH (Cyberagentur). The Agentur für Innovation in der Cybersicherheit GmbH did not interfere in the research process and its results. It was further supported by the SSE-concept to generate synthetic steganographic network data and embedding method EM_5 generously contributed by the project SYNTHESIS, funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV, project no. 1501666B) in the framework of the German reactor safety research program.

REFERENCES

[1] T. Neubert, B. Peuker, E. Schueler, H. Ullrich, L. Buxhoidt, and C. Vielhauer, "An analysis framework for steganographic network data in industrial control systems," *in Proceedings of SECURWARE2024 in Nice, France from November 3, 2024 to November 7, 2024; ISBN: 978-1-68558-206-7; ISSN: 2162-2116; online: https://www.thinkmind.org/library/SECURWARE/SECURWARE_2024/securware_2024_2_130_30058.html*, 2024.

[2] MITRE-ATT&CK, "Data obfuscation: Steganography," *https://attack.mitre.org/versions/v14/techniques/T1001/002/*, 2020.

[3] D. Kushner, "The real story of stuxnet," *https://spectrum.ieee.org/the-real-story-of-stuxnet, last access: 19/09/2024*, 2013.

[4] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," *SANS Institute, https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf*, 2016.

[5] I. Dragos, "Assessment of reported malware infection at nuclear facility," *https://www.dragos.com/blog/industry-news/assessment-of-reported-malware-infection-at-nuclear-facility/*, 2019.

[6] M. Hildebrandt, K. Lamshoeft, J. Dittmann, T. Neubert, and C. Vielhauer, "Information hiding in industrial control systems: An opc ua based supply chain attack and its detection," *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security, https://doi.org/10.1145/3369412.3395068*, 2020.

[7] T. Neubert, B. Peuker, L. Buxhoidt, E. Schueler, and C. Vielhauer, "Synthetic embedding of hidden information in industrial control system network protocols for evaluation of steganographic malware," *Tech. Report, arXiv, https://doi.org/10.48550/arXiv.2406.19338*, 2024.

[8] T. Neubert, C. Kraetzer, and C. Vielhauer, "Artificial steganographic network data generation concept and evaluation of de- tection approaches to secure industrial control systems against stegano- graphic attacks," *In The 16th International Conference on Availability, Relia- bility and Security (ARES 2021), August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3465481.3470073*, 2021.

[9] K. Lamshoeft, T. Neubert, J. Hielscher, C. Vielhauer, and J. Dittmann, "Knock, knock, log: Threat analysis, detection & mitigation of covert channels in syslog using port scans as cover," *Digital Investigation 2022 (DFRWS EU 2022), https://doi.org/10.1016/j.fsidi.2022.301335*, 2022.

[10] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, T. Neubert, and S. Zillien, "A generic taxonomy for steganography methods," *ACM Comput. Surv. vol.57, no.9, https://doi.org/10.1145/3729165*, 2025.

[11] W. Mazurczyk, S. Wendzel, and K. Cabaj, "Towards deriving insights into data hiding methods using pattern-based approach," *ARES 2018, 13th International Conference on Availability, Reliability and Security; Hamburg, Germany, August 27 - August 30, ISBN: 978-1-4503-6448-5*, 2018.

[12] OPC-Foundation, "Unified architecture," *https://opcfoundation.org/about/opc-technologies/OPCUA/*, 2008.

[13] ACROMAG-Incorporated, "Introduction to modbus tcp/ip," *https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf, last access: 19/09/24*, 2005.

[14] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, and T. Neubert, "A revised taxonomy of steganography embedding patterns," *In the Proceedings of 16th International Conference on Availability, Reliability and Security (ARES 2021), Article No.: 67, Pages 1 - 12, August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 12 pages. https://doi.org/10. 1145/3465481.3470069*, 2021.

[15] K. Lamshoeft, C. Kraetzer, J. Dittmann, T. Neubert, and C. Vielhauer, "Information hiding in cyber physical systems: Challenges for embedding, retrieval and detection using sensor data of the swat dataset," *In Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security (IHMMSec '21), pp. 113 - 124, June 22–25, 2021, Virtual Event, Belgium. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3437880.3460413*, 2021.

[16] Wireshark-Foundation, "About wireshark," *https://www.wireshark.org/about.html*, 2024.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl. 10.1145/1656274.1656278*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[18] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: Data mining toolbox in Python," *Journal of Machine Learning Research, http://jmlr.org/papers/v14/demsar13a.html*, vol. 14, pp. 2349–2353, 2013.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous systems," *https://www.tensorflow.org/*, 2015.

[20] F. Chollet *et al.*, "Keras," *https://keras.io*, 2015.

[21] L. Caviglione, "Trends and challenges in network covert channels countermeasures," *Applied Sciences*, vol. 11, 02 2021.

[22] T. Neubert, A. J. C. Morcillo, and C. Vielhauer, "Improving performance of machine learning based detection of network steganography in industrial control systems," *In the Proceedings of 17th International Conference on Availability, Reliability and Security (ARES 2022), Article No.: 51, pp. 1 - 8, August 23– 26, 2022, Vienna, Austria. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3538969.3544427*, 2022.

[23] Hak5, "Packet squirrel mark ii," *https://shop.hak5.org/products/packet-squirrel-mark-ii*, 2025.

[24] K. Pearson, "X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 50, no. 302, https://doi.org/10.1080/14786440009463897*, 2009.

[25] Fischertechnik, "Instruction material for the learning factory industry 4.0 24v," *https://www.fischertechnik.de/en/industry-and-universities/technical-documents/simulate/training-factory-industry-4,-d-,0-24v*, 2025.