International Journal on Advances in Security

2010 vol. 3 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

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

Reference should mention:

> *International Journal on Advances in Security, issn 1942-2636*
> *vol. 3, no. 1 & 2, year 2010, http://www.iariajournals.org/security/*

Reference to an article in the journal is as follows:

> *<Author list>, "<Article title>"*
> *International Journal on Advances in Security, issn 1942-2636*
> *vol. 3, no. 1 & 2, year 2010, <start page>:<end page> , http://www.iariajournals.org/security/*

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Security
Volume 3, Number 1 & 2, 2010

- Aljosa Pasic, ATOS Origin, Spain
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Michiaki Tatsubori, IBM Research - Tokyo Research Laboratory, Japan
- Ian Troxel, SEAKR Engineering, Inc., USA
- Hans P. Zima, Jet Propulsion Laboratory/California Institute of Technology - Pasadena, USA // University of Vienna, Austria

Security in Internet

- Evangelos Kranakis, Carleton University, Canada
- Clement Leung, Victoria University - Melbourne, Australia
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Yong Man Ro, Information and Communication University - Daejon, South Korea

International Journal on Advances in Security
Volume 3, Numbers 1 & 2, 2010

CONTENTS

A survey of software tools for the creation of networked testbeds **1 - 12**
Christos Siaterlis, Inst. for the Protection and Security of the Citizen Joint Research Centre, Italy
Marcelo Masera, Inst. for the Protection and Security of the Citizen Joint Research Centre, Italy

Performance Analysis of Time Synchronization Protocols in Wireless Sensor Networks with Regular Topologies **13 - 26**
José A. Sánchez Fernández, Universidad Politécnica de Madrid, Spain
José F. Martínez Ortega, Universidad Politécnica de Madrid, Spain
Ana B. García Hernando, Universidad Politécnica de Madrid, Spain
Lourdes López Santidrián, Universidad Politécnica de Madrid, Spain

Network Forensics Models for Converged Architectures **27 - 33**
Juan C. Pelaez, U.S. Army Research Laboratory, USA
Eduardo B. Fernandez, Florida Atlantic University, USA

Self-Healing and Secure Adaptive Messaging Middleware for Business-Critical Systems **34 - 51**
Habtamu Abie, Norwegian Computing Center, Norway
Reijo M. Savola, VTT Technical Research Centre of Finland, Finland
John Bigham, Queen Mary University of London, UK
Ilesh Dattani, Q-Sphere Ltd, UK
Domenico Rotondi, TXT e-solutions SpA, Italy
Giorgio Da Bormida, CNIT, Italy

Information Geometrical Approximation of Quantum Channel Security **52 - 67**
Laszlo Gyongyosi, Budapest University of Technology, Hungary
Sandor Imre, Budapest University of Technology, Hungary

Security Patterns for Untraceable Secret Handshakes with optional Revocation **68 - 79**
Annett Laube, Bern University of Applied Science, Switzerland
Alessandro Sorniotti, Institut Eurecom Sophia-Antipolis, France
Paul El Khoury, LIRIS University of Claude Bernard Lyon 1, France
Laurent Gomez, SAP Research Sophia-Antipolis, France
Angel Cuevas, University Carlos III of Madrid, Spain

A survey of software tools for the creation of networked testbeds

Christos Siaterlis
Inst. for the Protection and Security of the Citizen
Joint Research Centre
Via E. Fermi 2749, 21027 Ispra (VA) Italy
e-mail: christos.siaterlis@jrc.it

Marcelo Masera
Inst. for the Protection and Security of the Citizen
Joint Research Centre
Via E. Fermi 2749, 21027 Ispra (VA) Italy
e-mail: marcelo.masera@jrc.it

***Abstract*—The development of testbeds for networking research has been driven by the need for experimentation with complex systems, like the Internet, that simplistic simulators fail to reproduce. Recently, networked testbeds seem to head towards more advanced, flexible and automated experimental platforms mainly as the results of many projects and research initiatives in the field of Future Internet architectures. Although numerous publications can be found, most of them refer to prototypes and work in progress rather than to publicly available software that is ready to be widely used for the creation of such testbeds. The first contribution is the development of a framework that can be used to capture the main features of the available software. The second contribution is a literature review of state-of-the-art tools and their comparison under common criteria. This systematic analysis allows other researchers to make informed decisions about the usability of already available tools and decrease the initial cost of developing a new testbed, leading to an even wider use of such platforms. This paper provides the reader with a useful reference list of readily available software to choose from while designing or upgrading a research infrastructure, laboratory or experimentation facility.**

***Keywords*—distributed test bed; emulation ; network research;**

I. Introduction

The development of advanced testbeds for networking research and in particular research for Future Internet protocols and architectures is a recent trend that becomes evident after consideration of the amount of related projects and research initiatives [1]. Most notable are: the 'Global Environment for Network Innovation' or GENI [2] as the most important initiative with a multi million budget in the US, the ICT FIRE (Future Internet Research & Experimentation) initiatives [3] and the FEDERICA project [4] in Europe. A researcher that will try to familiarize himself with the topic of experimental platforms for networking research will face tons of acronyms and a huge list of relevant projects. The terms testbed and "experimental platforms" do not have clear definitions and are frequently used interchangeably although they might have different interpretations especially in size and sophistication. The lack of precise definitions has often introduced confusion in the related literature. In our point of view the fidelity or "level of realism" of an experiment often determines which description is most suitable for the specific setup: simulation, emulation or real testbed [5]. As this labeling might lead to confusions we will try to avoid it and characterize an *experimental platform as a combination of hardware and software based on an architectural design that enables the researcher to conduct experiments using components that provide different levels of realism or abstraction* (e.g., real or virtual hosts, simulators, traffic generators, mathematical models).

Recently, experimental platforms seem to gain wider use in other subfields of networking research as well e.g., in security research[6]. The driving force is the need not only for theoretical but also empirical security research that is based on more solid and compelling evidence and will produce useful results that can be promptly used to strengthen our Critical Information Infrastructures. The field of Internet security in particular, is often handled in a non-systematic way [7]. Furthermore:

- new developments in the field of security are often presented as "hacks", without detailed analysis of prerequisites and consequences;
- metrics about the security of a systems, system-of-systems or networks do not exist as a shared basis among researchers and practitioners [8];
- security-relevant data that can be used for research are scarce (mostly because of their sensitive nature);
- experimental platforms are often oversimplified and of limited scale and cannot accurately simulate real complex Internet environments;
- experiments are designed ad-hoc, without a methodology and a clearly stated approach for setting up testing campaigns, measuring significant variables and examining their outcome. The consequence is that the results are hardly reproducible by other researchers.

This strong need for networking research that follows rigorous scientific methods and produces provable results that are closely bound to reality by proving and disproving hypotheses drives us to the development of new experimental platforms. This need is further analyzed by Neville and Li in [9]. For the time being, one of the barriers one has to overcome while designing, extending or developing a new testbed is the required effort to review the available approaches and software. Our paper aims to lower this initial barrier.

The excess of available information and the fact that most sources refer to prototypes and work in progress rather than published software that is ready for wider use, are two factors that magnify the need for a structured and practical review of available software. A survey of emulation related products has been published by Rimondini et al. [10], a second one about tools that can be used to develop, test or utilize routing protocols [11] was carried out by Oliver Bonaventure and finally the work of Volvnkin et al. [12] deals with general recommendations and architectural issues regarding the development of a testbed for information security experiments. To the authors's knowledge until today a study that compares under common criteria the software that is available for the development of distributed experimental platforms for networking research does not exist. Our work focuses on publicly available software that can be used for networking research and intentionally excludes platforms :

- that share computational resources (e.g., GRIDS);
- that focus only in simulation (like Simgrid [13]) ;
- that are specific to wireless or sensor networks (like signetLab [14]);
- that run on a single computer (like Marionnet [15], IMUNES [16] and Netkit [17]) and aren't a distributed testbed;
- that use custom hardware (like the Open Network Laboratory [18]) rather than using Commercial, off-the-shelf (COTS) components.

The paper is structured as follows. We begin in Section II with a presentation of an experiment's life cycle on an experimental platform. We continue in Section III with the proposed framework that can capture the features of a software suite for the creation of such platforms. In Section IV we provide the reader with a useful reference list of readily available software that are compared under common criteria. This list can be extremely useful while designing or upgrading a research infrastructure, laboratory or experimentation facility. In Section V, some significant ideas and approaches that are part of related projects will be shortly described. Finally in Section VI we summarize the main conclusions of our study.

II. Experiment lifecycle

In this paper an experimental platform is understood as a combination of hardware, software, architectural and operational policies that form a facility for conducting experiments with computer networks. In order to analyze further the features of experimental platforms it is useful to model the life of an experiment on top of an experimental platform. We define six phases of an experiment's lifecycle (Figure 1) - similar to the work of Guiller et al. [19] and Miyachi et al. [20].

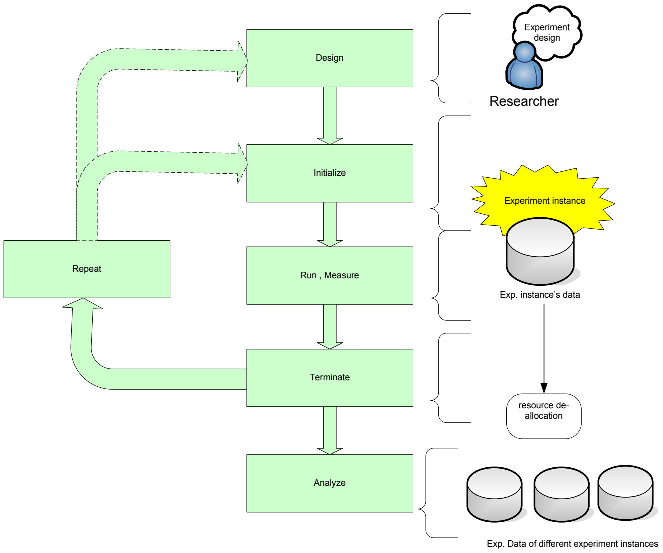


Figure 1. An experiment's lifecycle

1) **Design phase**. In this phase the researcher designs an experiment according to a scenario. The experiment has to be defined: a) structurally by using different building blocks such as hosts, network devices, links etc; b) functionally by defining the main and background processes that recreate the experiment environment as well as the variables of interest that have to be captured; c) procedurally by specifying the phases and steps of the experiment, and the conditions and criteria for launching and terminating each one of them. An important element for the procedural definition of an experiment is the notion of an experiment timeline (relative time).
2) **Initialization phase**. Before the experiment runs, all components that were defined in the design phase have to be initialized. The completion of this process involves the instantiation and configuration of all building blocks.
3) **Execution and Measurement phase**. During this phase the experiment is running and the platform is triggering events and actions according to a predefined schedule (experiment design). Each experiment that runs on the platforms should be identifiable with a unique ID that defines a specific instance/run of an experiment's design. A single experiment design can be instantiated and run many times in order to perform statistical validation of the outcome. During the experiment execution, measurement processes capture all data that will be valuable to the researcher during the analysis phase. The collected data should be stored and labeled with the specific experiment instance's unique ID into a repository.
4) **Termination phase**. When an experiment's in-

stance is meeting some termination criteria (e.g., amount of time that has passed) the instance is considered finished and a termination process is initiated. All components that are part of the experiment have to be brought into a clean state without retaining history (for example by shutting them down).

5) **Repetition phase**. An experiment might be repeated several times with or without changes in its design. Repeating an experiment without changes serves statistical validation of the outcome whereas the repetition of an experiment by changing one or more controlled variables (not only in a strict term e.g., a detection threshold but also in a wider sense e.g., the network topology or a host's configuration) can be used to conduct sensitivity analysis. Automation of this phase is very important so that the researcher can conduct repeating experiments efficiently without manual interaction.
6) **Analysis phase**. After all data has been collected the researcher will be able to analyze it without interaction with the experiment platform.

III. Features of software for experimental platforms

Ideally, an experimental platform for networking research would support the execution of complex, large scale and even disruptive experiments using rigorous scientific methods. The desired characteristics - features of such platforms are many and can be realized with different means; for example with the use of software, hardware or even organizational measures. It is obvious that every single feature can influence the overall usefulness and simultaneously implementing all of them is definitely nontrivial. One of the main reasons is that these features are not independent and design choices regarding one of them can influence the available implementation options of a different feature. By extending previous work of the DETER project [21] and Masera et al. [22], we first identify a set of the most important basic features and then present other more sophisticated characteristics that build upon the basic features and are called compound features. The features are labeled as Fx where $x \in N$ and discussed one by one in following Sections. We provide also a map of the dependencies between basic and compound features (Figure 2) that could serve as a "scorecard" for evaluating different approaches for the creation of a testbed. This framework can be used also to provide an overview of desired features and demonstrate the complexity of the development process of a new experimental platform.

A. Basic features

F1. Control of the experiment's environment is one of the most important attributes of the scientific approach and enables the researcher to analyze how a hypothesis is influenced from dependent and independent variables. This does not imply that we cannot use random or stochastic processes in an experiment as long as they are of a controlled nature (e.g., statistically modeled).

F2. An experiment clock is a necessary feature for every experimental platform as it can provide a solid reference point to characterize the occurrence of events (event scheduling) and the measurement of various variables. The synchronization of the internal clock of different devices with the master experiment clock is a non trivial and important task.

F3. Separation of control, measurement and experiment planes. Measurements should not interfere with the experiment because they might alter the experiment's outcome. Preferably control and measurement planes should be differentiated as well to maximize measurement accuracy.

F4. Storage facilities are needed to store the description of an experiment and the data that was collected during its execution (measurements). Storage facilities should provide secure access to the data and support backup and restore. An important aspect of choosing storage facilities are the supported data structures. Relational databases are not the only solution and specialized data structures like Round Robin Databases could be more efficient for specific measurements like time-series [23].

F5. The use of standard Application Programming Interfaces (APIs) by the software that supports an experimental platform is not only required as a good design and programming practice but because it is crucial for allowing researchers to extend its functionality in custom ways. The ability to automate tasks on an experimental platform is also dependent on the existence of APIs that are exposed to the user.

F6. Heterogeneity of technologies that can be included in an experiment (e.g., components that range from custom hardware to simulators) is an essential attribute of a platform in order to permit experiments with future technologies. The platform should not restrict the researcher to specific hardware and software vendors and should allow mixed configurations in order to resemble real world scenarios. This could be achieved in one extent with the use of virtualization technologies and standards.

F7. Clean reconfiguration means that the initial state of the experiment is build from the scratch without past experiments influencing it. This functionality is imperative for reliable repetitive experiments. No hidden state should be kept by the components of the experiment.

F8. Virtualization is important for scalability and can provide some independence from physical resources helping thus to automation, rapid reconfiguration and topology flexibility. Additionally it can lower operational and capital expenses. In this context virtualization is the ability to reuse a single physical resource. The extent of virtualization can differ significantly between two approaches. For example host virtualization can be implemented in a way where a single host running a specific Operating System (OS) can act as being multiple hosts with the same OS

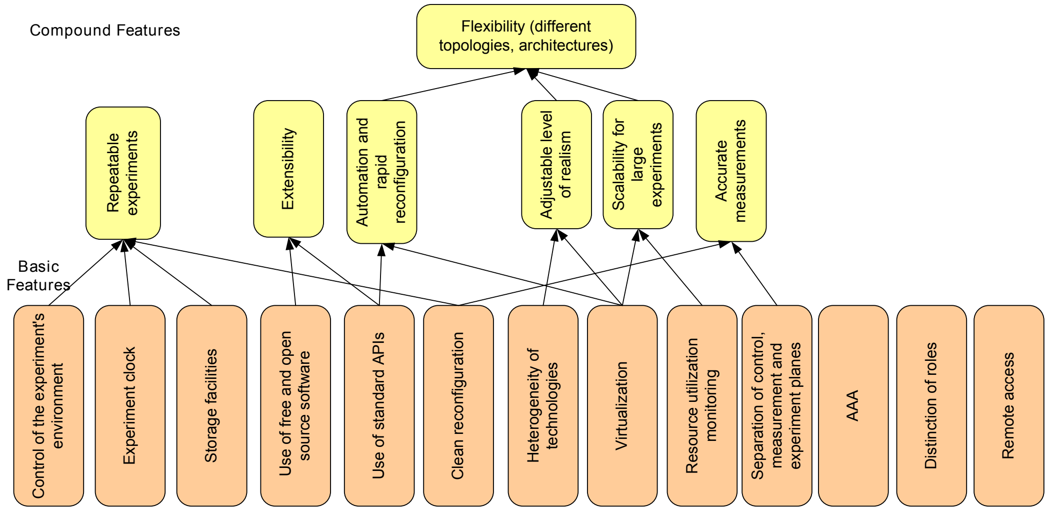

Compound Features
Flexibility (different topologies, architectures)
Repeatable experiments
Extensibility
Automation and rapid reconfiguration
Adjustable level of realism
Scalability for large experiments
Accurate measurements
Basic Features
Control of the experiment's environment
Experiment clock
Storage facilities
Use of free and open source software
Use of standard APIs
Clean reconfiguration
Heterogeneity of technologies
Virtualization
Resource utilization monitoring
Separation of control, measurement and experiment planes
AAA
Distinction of roles
Remote access

Figure 2. Dependencies of compound features from basic features.

(like in FreeBSD jails or Solaris zones) or can act as being multiple hosts with different OS (like VMware). Recently, besides host virtualization the concept of router virtualization emerged as a new approach; although software routers date back many years. Typical examples are logical routers by Cisco and Juniper and the Openflow switch [24].

F9. Resource utilization monitoring deals with the need of knowing the status of the available resources (in real time and historically). Such information is crucial for capacity planning and accounting purposes. The level of detail of the utilization information can differ significantly ranging from simple CPU utilization to per process details.

F10. Use of free and open source software is important because it can foster deeper understanding of underlying mechanisms and principles. Open source software can be reviewed and altered by other researchers that want to collaborate. Finally, academic researchers often consider that the free distribution of software helps to create a community and to reach critical mass, needed to support and extend a software suite.

F11. AAA - Authentication, Authorization and Accounting is essential if multiple users and especially from different organizations have access to an experimental platform. The importance of such security related functionality, e.g., isolating experiment resources and prohibiting users to view each other's data, is due to the fact that the experiments might involve the handling of sensitive data like network traffic captures. AAA functions are also related to the storage facilities as access to the stored data should be controlled.

F12. Distinction of roles. Access to an experimental platform could be restricted to private users but in some cases it might be beneficial to open up to a wider community. These issues can be defined in "Usage and Operation policies" that might differentiate the users and assign different roles and rights. For example requests to use an experimental platform that come from external users might have to get reviewed and prioritized before granted access.

F13. Remote access can form the basis of deeper and wider collaboration with researchers throughout the world. A first step is to support remote access to experimental data. An extension would involve the possibility to remotely control and monitor an experiment.

B. *Compound features*

On top of the basic features more advanced features and functionality can be built:

F14. Repeatable experiments require a controlled environment but to achieve them the researcher has to define clearly and in detail the experiment's initial and final state as well as all events in between these two states. These states and events form an experiment scenario. To reproduce a previously stored experiment scenario the researcher should be able to setup the experimental platform in the initial state and trigger all necessary events in the right order and time of occurrence.

F15. Extensibility can be viewed as the ability to adapt to future needs and requirements. To allow future extensions an experimental platform should have a modular design with clearly defined interfaces. In addition following standards and best practices (for measurements and setup) and the use of open source software improves extensibility.

F16. Automation and rapid reconfiguration aim to ease the researcher in his work and enable a more efficient use of the available resources. If 'rapid reconfiguration' is implemented in the form of a scripted experiment setup the benefits are multiple: experiments can be conducted without human interaction and thus minimum dependence from human errors, working hours and the limited speed of human actions. This can eventually lead to massively repetitive experiments and the possibility of statistical validation of the results.

F17. Adjustable level of realism means that we use only the required level of detail that is sufficient to test the experiment hypothesis. For example one experiment might need to reproduce a network at the very low level using real routers and computers with specific configurations (reproducing even the lower layers of the OSI model i.e., Layer 1 and 2). Of course reproducing all details is not easy (eg. reproducing a link with specific delay and loss). Another experiment might just need to reproduce reality with less detail and thus the use of router and traffic simulators might be sufficient (focusing for example at the application layer). The concept of adjustable level of realism is to have the option to use real hardware when it's really needed and simulators or other abstractions when not. Even if someone uses real hardware and software it is very hard to reproduce realistic network traffic. This is one of the greatest challenges for realistic experiments, rather than solely functional, and address it one might use special traffic generators or try to replay real traffic from the Internet.

F18. Scalability for large experiments implies that the platform can be extended to support a large number of experimental nodes (real or virtual) potentially distributed over several physical locations. An experiment might have a considerable size in order to serve as a meaningful abstraction of the complex structure and interaction phenomena of today's Internet. To achieve scalability an experimental platform should be designed to leverage on parallelism whenever it's possible (e.g., multi-process or multi-threaded software).

F19. Accurate measurements are the key for every scientific experiment. The separation of the measurement processes from the experiment might not be always achievable but the measurement process should have a minimum impact to the experiment result. Furthermore the researcher has to consider that the accuracy of measurements depends on many factors like the choice of sampling rate and sampling strategy. Capturing raw data from all possible and diverse sources (both nodes and links) is a good approach because the researcher will be able to extract measurements from the collected data at a later stage.

F20. Probably the most sophisticated feature is "Flexibility" i.e., the ability to reproduce different topologies and architectures of many different layers and with different levels of realism. In the sake of rapid reconfiguration and automation we might have to abandon the physical layer and focus above the data link layer up to the application layer.

IV. Review of existing software

Most experimental platforms aim to provide many of the above-mentioned features but each one has its one strengths and weaknesses. In the ideal case the software that is used to create an experimentation platform should help the researcher in all steps of an experiment's lifecycle: design, initialization, execution and measurement, termination, repetition and analysis, while minimizing the cost in terms of required human effort, financial cost and time commitment. During the review, we first uncovered the basic features of the reviewed software then inferred the higher level compound features and finally summarized them in a set of tables (Tables I-III). During this process we have taken also into account certain architectural and operational requirements that the specific approach/software implies. Furthermore two clarifications have to be made. First, it should be clear that the tools that we will present might evolve and get extended. This review doesn't serve as a definite guide about the possible uses of the software but rather as a first level overview that can quickly familiarize the reader with the different approaches. The second point is that practical details are often hidden from high level presentations and research papers making thus the process of understanding the practical use of some software very hard. Although a literature review will always miss some details, we have made an extensive effort to dig out the most important details from many different sources (e.g., mailing lists, configuration manuals).

In order to present the review of the available software tools in a systematic way we group them in two broad categories that were proposed by the NSF Report on Network Research Testbeds [1] [25]:

- **Overlay testbeds** are build on top of existing infrastructures and have been extensively used in the past to test and deploy new types of protocols, services etc in a large scale environment that cannot be recreated in a dedicated facility. By relying on an existing and widely used underlying infrastructure they can host experiments of great scale but on the other hand they are constrained by the limitations (e.g., bandwidth, delay) of the underlying infrastructure that connects the overlay nodes. In this sense these testbeds are not isolated from the production systems and networks. It is interesting to note that the early Internet was essentially an overlay on the telephone network.
- **Cluster testbeds** are typically experimental facilities that contain a large number of dedicated physical resources that can be used as components for network emulation e.g., links, routers and generic servers that are used as hosts, routers or even WAN emulators. The software used on cluster testbeds often assumes that all the resources are under a single administrative domain.

A. *Software for overlay testbeds*

The most famous software in this category is **Planetlab** [26], a term that is some times used to refer to the software suite (Table I) and other times to the infrastructure. Planetlab as an infrastructure consists of a set of PCs

[1] according to their naming convention we are covering multi-user, experimental facilities (MXF) that support research rather than proof of concept testbeds.

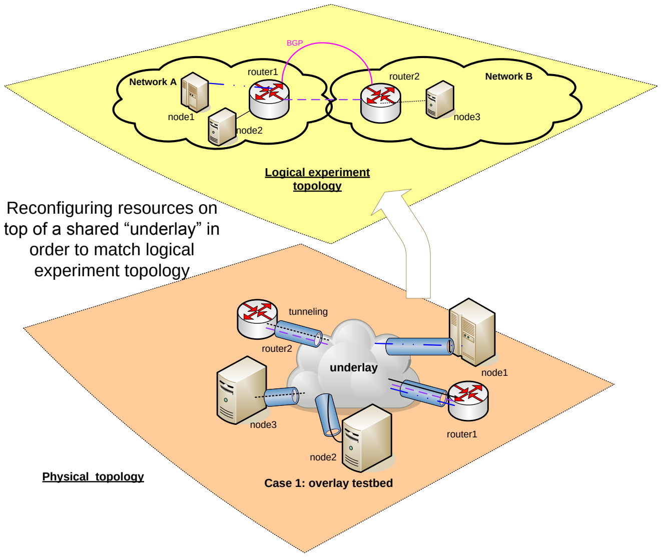


Figure 3. A rough illustration of an experiment instantiation over an overlay testbed.

(called nodes) connected to the Internet and forming an overlay network. The nodes are based on Linux and are divided into slices using virtualization techniques. A slice is a network of virtual machines, with a subset of a node's resources bound to each virtual machine. Users can deploy an experiment over specific slices of nodes and links of the overlay network. This approach is ideal for experiments that need dispersed points of presence (Internet-wide) e.g., realistic testing of new protocols. The Planetlab software allows the creation of "private Planetlabs" through a version called MyPLC. An example is Everlab [27] which is essentially a private Planetlab on high-end clusters spread over Europe (Evergrow). Planetlab's success has triggered the creation of various projects such as ONELAB which extends Planetlab to support wireless research and VINI [28] which extends the Planetlab software to support simultaneous experiments with arbitrary network topologies on a shared physical infrastructure. VINI is suitable for networking research because it exposes lower level interfaces (interfaces to virtual network devices e.g., TUN/TAP interfaces instead of sockets) to slices. The virtual network topologies are build using a clever combination of various open source software like Quagga and User Mode Linux (UML) and in a simplified manner they could be described as multiple UML instances connected by virtual point-to-point Ethernet links. VINI was deployed initially as a prototype on top of Planetlab (as an infrastructure) but the intension of its creators was to deploy it on a separate infrastructure.

In Figure 3 we present how an experiment would be implemented over an overlay testbed.

A similar concept for the creation of network-layer overlays over the Internet is used in **X-Bone** (Table I). X-Bone is a software tool that officially runs on FreeBSD and Linux and can be used to discover, configure, and monitor network resources to create overlays over a multicast enabled IP network. The X-Bone overlays isolate experiments from each other because they are constructed in such a way that packets sent to the virtual addresses of an overlay will go through the virtual links, while packets addressed to the base addresses will go through the base network (usually the Internet) instead of the virtual network. Of course this mechanism doesn't ensure proper isolation in cases of misconfiguration or error. Nodes communicate by sending and receiving packets to and from virtual addresses of the overlay. X-Bone provides a primitive mechanism to support application deployment on top of overlays and the applications have to be modified to use the overlay network.

B. Software for cluster testbeds

The need to reduce the required time to create the experiment environment e.g., vulnerable hosts and attackers, and allow the researchers to focus on developing novel technologies rather than rebuilding their test infrastructure is well known. An interesting approach dealing with this problem but without obvious continuation over the years was ViSe (Virtual Security Testbed) [32]. It proposed the use of VMware to create and store virtual machine images of different hosts that could represent attackers, detectors and victims in a cyber-attack scenario. The idea of building a virtual machine image repository is interesting and could help the collaboration between researchers but ViSe addresses only this issue and the flexible configuration of the network topology during an experiment is outside of its scope.

On the other hand another important approach that seems actively supported is the **Grid'5000** project [33]. The main purpose of this platform is to serve as an experimental testbed for research in Grid Computing and in all the layers between the network protocols up to the applications. Grid'5000 allows its users to deploy their own operating system on the resources they reserve for a limited number of hours. Furthermore its assumed that the resources are not a single cluster located in a single geographical position but distributed on different sites (allover France). The sites are connected through VLANS implemented by MPLS but are isolated from the Internet. Therefore the network interconnection of the resources is specific to the project. Nevertheless the project has released several tools that automate the tasks of reserving resources, deploying, configuring, monitoring and repeating experiments (Table II) like

- The HIPerNET tool is exploring the possibility for virtual network creation.
- Katapult automates some tasks for experiments e.g., deploying the nodes, re-deploying the nodes if too many of them failed, copying the user's SSH key to the node etc.

Table I
FEATURES OF SOFTWARE FOR OVERLAY TESTBEDS.

Feature	PlanetLab	X-Bone
F1.Control of the experiment's environment	In Planetlab there is no control of the underlying network as its designed to subject network services to real-world conditions and not to provide an isolated controlled environment. Using a private Planetlab installation on a dedicated infrastructure is nevertheless possible.	X-Bone was designed to create overlay networks and can be used either over the Internet or a dedicated controlled environment.
F2.Experiment clock and event scheduling mechanisms	Nodes are NTP synchronized but problems with clocks have been reported. No event triggering service.	Global clock is not provided but nodes can be NTP synchronized. No event triggering service.
F3.Separation of control, measurement and experiment planes	No separation between measurement and experiment planes. Isolation between experiments is provided in virtual hosts (on the OS level) and there are extensions like VIOLIN [29].	Control of the overlay is implemented with TCP connections over the underlying IP network. X-Bone keeps IP traffic on different overlays separated but additionally the researcher can enable IPSec and restrict each node to participate in one overlay at a time.
F4.Storage facilities	No centralized storage facility for user data but experiments are registered in central database.	No centralized storage facility. Custom overlays can be described in custom files and created through scripts.
F5.Use of programming interfaces	An XML-RPC programming interface exists (Planetlab Central API) through which users can access and update information in the database about users, nodes, sites, slices etc.	An X-Bone API protocol that serves as a programming interface exists.
F6.Heterogeneity of technologies (e.g., ability to use different OS)	Planetlab software runs only on Linux hosts. Custom hardware can get connected just like any other host in the Internet.	X-Bone is build by definition on top of hosts running FreeBSD or Linux. Unofficially Cisco routers are supported in overlay networks.
F7.Clean reconfiguration	Each sliver is independently created/configured. Secure ping of death can be used to reboot inaccessible nodes.	No support. It is left to the researcher.
F8.Virtualization of nodes and links	Virtual hosts can be used but with Linux OS. Virtual links exist e.g., connections between hosts in a slice but these are implemented in the OS (socket level).	Virtual hosts with different OS are not supported. Applications can access multiple overlays by using different IP address spaces.
F9.Resource utilization monitoring	Supported by software like CoMon , Plush etc.	No support. It is left to the researcher.
F10.Use of free and open source software	Yes, code is available via SVN.	Yes.
F11.Authentication and Authorization	Based on SSH keys.	Based on X.509 certificates.
F14.Repeatable experiments	No control of the underlying network can lead to repeatability problems.	No control of the underlying overlay network can lead to repeatability problems.
F15.Extensibility	Open source software and programming interfaces can provide extensibility.	Open source software and programming interfaces can support extensibility.
F16.Automation and rapid reconfiguration	Partially provided by other projects like Plush [30] where users describe their experiments in an XML document and Plush prepares the required resources.	The X-Bone API protocol could be used to automate the processes of requesting the deployment of an IP overlay and monitoring it.
F17.Adjustable level of realism	Researchers can use simulators in Linux OS and use any resource of the underlying infrastructure (e.g., Internet).	Researcher can use simulators but connecting with real hardware in custom ways is out of the scope of X-Bone's normal usage.
F18.Scalability for large experiments	Yes. Available nodes in 2009 are approximately 1000.	Yes, but underlying IP network has to support multicast.
F19.Accurate measurements (nodes and links)	Measurements during experiments are left to the researchers. For a discussion over the accuracy of the measurements depending on the measurement methods we refer to [31].	Measurements during experiments are left to the researchers.
F12.Distinction of roles	Yes. There exist roles for plain users and the people with the responsibility of overseeing their site's participation in Planetlab.	Roles could be defined in terms of Unix user accounts.
F13.Remote access	The testbed be used remotely.	Yes, overlays are remotely configurable.

- OAR is a reservation tool that allows the researcher to submit or reserve nodes either in an interactive or a batch mode.
- Kadeploy is an automatic deployment system that supports OS installation and configuration of nodes. Currently it deploys successfully Linux, *BSD, Windows, Solaris on x86 and 64 bits computers.
- Network eXperiment Engine is a tool that allows to simply script experiments involving hundreds of nodes. The scenarios are described through XML files where the topology, the configuration of and the interactions between the experimental nodes are documented.

Some of these tools are specific to Grid'5000 and the extent of the required modifications to run them on a different testbed is hard to estimate.

ModelNet [34] is a software that can emulate wide-area network conditions within a local area network. The ModelNet architecture comprises of 'edge nodes' and 'core routers' which are both hardware-wise normal servers. Users run the applications under test on the edge nodes using any OS and IP network stack and run unmodified application binaries. The edge nodes can even support multiple Virtual Machines as demonstrated in DieCast [35]. The core routers are FreeBSD servers that emulate the behavior of a WAN under the offered traffic load. The core routers upon receiving packets from the edge nodes route the traffic through an emulated network of pipes with specific characteristics such as queue length, bandwidth, latency and loss-rate. The emulation runs in real time, so packets traverse the emulated WAN with the same rates, delays and losses as the real network. ModelNet doesn't address issues like resource allocation and resource monitoring neither offers a measurement infrastructure for the nodes but nevertheless is a powerful tool to emulate WAN dynamics.

A more comprehensive approach has been followed by the StarBED [36] project in Japan. It consists of a facility with hundreds of generic PCs interconnected with Layer 2 switches that are shared between multiple concurrent experiments. In order to automate the resource allocation and the experiment execution processes, a software under the name **SpringOS** was designed and developed (Table III) . Each experiment is described in a file with the use of a custom script language. The system evaluates this description and through the interaction of several daemons automatically handles the following tasks:

- allocation of the required nodes according to the experiment description plus spares;
- initialization of experimental nodes using disk images;
- interconnection of experimental nodes by setting up VLANs in switches and IP addresses in nodes;
- synchronization of experimental nodes and message exchange;
- execution of an experiment scenario e.g., launching different programs;
- after the termination the resources are released;

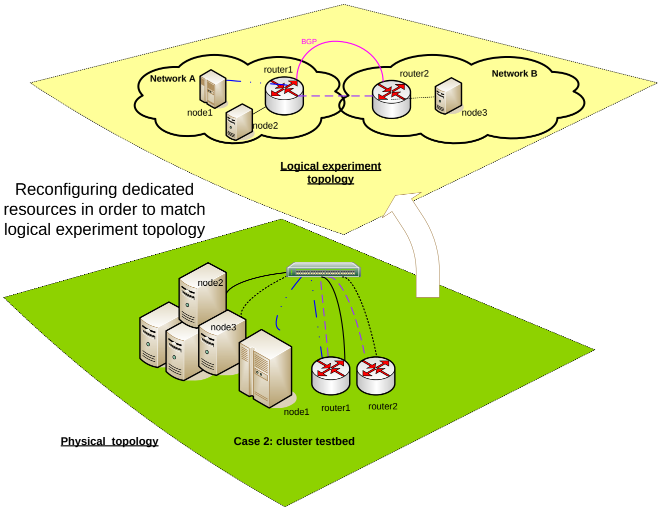


Figure 4. A rough illustration of an experiment instantiation over a cluster testbed.

SpringOS is bundled as a set of individual tools and does not cover all components that are required by the predefined architecture e.g., the dhcp and tftp daemons. This implies additional workload for the development of a complete and operational experimental platform. Furthermore the lack of a GUI for testbed manipulation might be seen as a shortcoming. Future plans of the StarBED community is to evolve into a testbed for ubiquitous networks and to provide an emulation environment.

One of the most advanced software suite for cluster testbeds is **Emulab** [37]. The name Emulab refers both to a facility at University of Utah and to a software. Nowadays the software is actively supported by multiple universities and there are many private installations throughout the world.

In Figure 4 we present how an experiment would be implemented over a cluster testbed.

From a technical point of view Emulab is quite sophisticated and feature rich and specifically:

- uses an extension of the NS [38] configuration language to describe an experiment;
- deals with the reservation and allocation of resources on the testbed;
- automates the installation of many operating systems on generic hosts i.e., experimental nodes;
- automates the deployment of custom disk images and also the creation of updated snapshots of disk images;
- supports the use of virtual hosts (FreeBSD jails) and virtual links (multiplexed links on FreeBSD);
- recreates a network topology by connecting nodes with a programmable switch using multiple VLANs;
- provides an event system that can launch arbitrary commands on hosts and modify link characteristics;
- allows the use of the simulation tool NS inside the

emulated network;

- supports packet capturing for monitoring purposes.

Finally, a software very similar to Emulab that can be used for emulating different network topologies between real or virtual hosts with the use of VLANs, is the Network Emulation Testbed (NET)[39] but the released software dates back to 2005 when the original publication was made.

V. Related work

The need for new experimental infrastructures for networking and distributed systems research as well as studies related to the design of the Internet of the Future has been the basis for several projects. Although some of them are frequently referenced we have not included them in our review because they do not provide software for the creation of testbeds (yet). Nevertheless for the sake of completeness we should mention the following:

- the 'Global Environment for Network Innovation' or GENI [2] as the most important initiative with a multi million budget in the US. GENI aims to become a wide-area shared platform that will be based on ideas from Planetlab, Emulab and Tycoon and act as a) a facility for controlled and repeatable experiments under safe conditions; b) a facility for precise, non-invasive observations of the behavior of existing networks and distributed systems under current network conditions; c)a field experimental station where new systems can be tested under actual network conditions.
- The ICT FIRE (Future Internet Research & Experimentation) initiative [3] that funds several projects towards Future Internet Research. The facility development efforts have been mainly towards overlay testbeds.
- The FEDERICA project [4](Federated E-infrastructure Devoted to European Researchers Innovating in Computing networking Architectures) aims to create a European wide experimental platform using the existing network infrastructures of National Research and Education Networks (NREN all over Europe and their interconnection network GÉANT). The proposed approach is to cut slices of the underlying resources (virtual hosts and routers) and allocate them for each experiment.
- DAS-3 (The Distributed ASCI Supercomputer 3) [40] is a project in Netherlands which provides a common computational infrastructure for researchers rather than a generic experimentation platform. It consists of a five-cluster wide-area distributed system, but its unique characteristic is that the clusters are connected by a optical network backbone which can be reconfigured on the fly (using optical routers with configurable wavelengths forming light paths). This hybrid optical network is called StarPlane and allows network users to partition the network resources and to create multiple overlay networks, each with a different logical topology.

VI. Conclusion

Networking research and particularly the assessment of the security of the current and future Internet should be based on solid and compelling evidence. In addition to a systematic analysis of incidents occurring to actual Internet systems, there is a strong need for testbeds for empirical security research. Current experimental platforms can be categorized in two types: cluster testbeds that use dedicated resources in isolation from production systems (or connected under strict control and monitoring safeguards) and overlay testbeds that use resources from an existing infrastructure and build an experimental overlay on top of them. The main advantages of cluster testbeds are the control and repeatability of experiments but their downside is that they offer artificial network conditions whereas overlay testbeds can offer real network conditions but less repeatability. Although the literature is rich of related projects and platforms, the publicly available software for the creation of a new testbed is limited. We provide an overview by presenting available tools according to a common set of basic and compound features. Emulab and Planetlab provide the most sophisticated software for each testbed type as well as documentation to support the development of private testbeds. In comparison with other tools they require the least customization and effort for creating a new private testbed. A promising approach that tries to combine the best from both approaches was recently proposed under the name "Flexlab" by Ricci et al. [41]. However further work in this direction as well as towards federated testbeds [42] is still needed.

References

[1] C. Siaterlis and M. Masera, "A review of available software for the creation of testbeds for internet security research," in *Advances in System Simulation, 2009. SIMUL '09. First International Conference on*, Sept. 2009, pp. 79–87.

[2] NSF, "Global environment for network innovations (GENI)," http://www.geni.net/ [Accessed January 14, 2010].

[3] "The european FIRE initiative future internet research and experimentation," overview of projects http://cordis.europa.eu/fp7/ict/fire/fire-fp7_en.html [Accessed January 14, 2010].

[4] "FEDERICA project," http://www.fp7-federica.eu/ [Accessed January 14, 2010].

[5] E. Göktürk, "A stance on emulation and testbeds, and a survey of network emulators and testbeds," in *21st EUROPEAN Conference on Modelling and Simulation ECMS*, 2007.

[6] DARPA, "National Cyber Range (NCR) Program," http://www.darpa.mil/sto/ia/ncr.html [Accessed January 14, 2010].

[7] R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, K. Levitt, B. Lindell, P. Liu, D. Miller, R. Mundy, C. Neuman, R. Ostrenga, V. Paxson, P. Porras, C. Rosenberg, J. D. Tygar, S. Sastry, D. Sterne, and S. F. Wu, "Cyber defense technology networking and evaluation," *Commun. ACM*, vol. 47, no. 3, pp. 58–61, 2004.

[8] INFOSEC Research Council (IRC), "Hard problems list," 2005.

[9] S. W. Neville and K. F. Li, "The rational for developing larger-scale 1000+ machine emulation-based research test beds," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 1092–1099, 2009.

[10] M. Rimondini, “Emulation of computer networks with netkit,” in *Technical Report RT-DIA-113-2007, Roma Tre University*, Jan 2007.

[11] O. Bonaventure, “Software tools for networking,” in *IEEE Network Magazine, Volume 18, Issue 6*, Nov.-Dec. 2004, pp. 4 – 5.

[12] V. S. A. Volvnkin, “Large-scale reconfigurable virtual testbed for information security experiments,” in *Proceedings of TridentCom*, May 2007, pp. 1–9.

[13] H. Casanova, A. Legrand, and M. Quinson, “Simgrid: A generic framework for large-scale distributed experiments,” in *Proceedings of UKSIM 2008*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 126–131.

[14] R. Crepaldi, S. Friso, A. F. Harris, III, A. Zanella, and M. Zorzi, “The design, deployment, and analysis of signetlab: a sensor network testbed and interactive management tool,” in *Proceedings of WiNTECH 2006*. New York, NY, USA: ACM, 2006, pp. 93–94.

[15] J.-V. Loddo and L. Saiu, “Status report: marionnet or "how to implement a virtual network laboratory in six months and be happy",” in *ML ’07*. New York, NY, USA: ACM, 2007, pp. 59–70.

[16] M. M. M. Zec, “Operating system support for integrated network emulation in imunes,” in *Proceedings of the 1st Workshop on Operating System and Architectural Support for the on demand IT InfraStructure*, Oct. 2004.

[17] M. Pizzonia and M. Rimondini, “Netkit: easy emulation of complex networks on inexpensive hardware,” in *Proceedings of TridentCom*, 2008, pp. 1–10.

[18] J. DeHart, F. Kuhns, J. Parwatikar, J. Turner, C. Wiseman, and K. Wong, “The open network laboratory,” in *Proceedings of 37th SIGCSE*. New York, NY, USA: ACM, 2006, pp. 107–111.

[19] P. V.-B. P. R. Guillier, “Nxe: Network experiment engine: software to automate networking experiments in real testbeds.”

[20] K.-i. C. Toshiyuki Miyachi, Shinsuke Miwa and Y. Shinoda, “On the nature of network experiments —issues to automate network experiments—,” in *TESTCOM/FATES2008 Supplementary Proceedings*.

[21] T. Benzel, R. Braden, D. Kim, C. Neuman, A. D. Joseph, and K. Sklower, “Experience with DETER: A testbed for security research,” in *TRIDENTCOM*, 2006.

[22] M. Masera and I. N. Fovino, “Methodology for experimental ict industrial and critical infrastructure security tests,” in *International Conference on Availability, Reliability and Security (ARES) conference*, 2009.

[23] T. Oetiker, “About RRDtool,” http://oss.oetiker.ch/rrdtool/ [Accessed January 14, 2010].

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[25] “Report of NSF workshop on network research testbeds,” Oct 2002, http://www-net.cs.umass.edu/testbed_workshop/ [Accessed January 14, 2010].

[26] “Planetlab,” http://www.planet-lab.org/biblio [Accessed January 14, 2010].

[27] E. Jaffe, D. Bickson, and S. Kirkpatrick, “Everlab: A production platform for research in network experimentation and computation,” in *LISA*, 2007, pp. 203–213.

[28] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, “In vini veritas: realistic and controlled network experimentation,” in *SIGCOMM*. New York, NY, USA: ACM, 2006, pp. 3–14.

[29] X. Jiang and D. Xu, “Violin: Virtual internetworking on overlay infrastructure,” in *ISPA*, 2004, pp. 937–946.

[30] J. R. Albrecht, C. Tuttle, A. C. Snoeren, and A. Vahdat, “Planetlab application management using plush,” *Operating Systems Review*, vol. 40, no. 1, pp. 33–40, 2006.

[31] N. Spring, L. Peterson, A. Bavier, and V. Pai, “Using planetlab for network research: myths, realities, and best practices,” *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 17–24, 2006.

[32] M. Richmond, “ViSe: A virtual security testbed,” Jun 2005, master’s Project Report. Department of Computer Science, University of California, Santa Barbara.

[33] “The grid’5000 project,” https://www.grid5000.fr [Accessed January 14, 2010].

[34] K. Yocum, K. Walsh, A. Vahdat, P. Mahadevan, D. Kostic, J. Chase, and D. Becker, “Scalability and accuracy in a large-scale network emulator,” *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 28–28, 2002.

[35] D. Gupta, K. V. Vishwanath, and A. Vahdat, “Diecast: Testing distributed systems with an accurate scale model,” in *NSDI*, 2008, pp. 407–422.

[36] T. Miyachi, K.-i. Chinen, and Y. Shinoda, “Starbed and springos: large-scale general purpose network testbed and supporting software,” in *valuetools ’06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*. New York, NY, USA: ACM, 2006, p. 30.

[37] “Emulab ,” http://www.emulab.net/ [Accessed January 14, 2010].

[38] ISI, “Network simulator ns-2,” http://www.isi.edu/nsnam/ns/ [Accessed January 14, 2010].

[39] S. Maier, D. Herrscher, and K. Rothermel, “Experiences with node virtualization for scalable network emulation,” *Computer Communications*, vol. 30, no. 5, pp. 943–956, 2007.

[40] “DAS-3 the distributed asci supercomputer 3 project,” http://www.cs.vu.nl/das3/ [Accessed January 14, 2010].

[41] R. Ricci, J. Duerig, P. Sanaga, D. Gebhardt, M. Hibler, K. Atkinson, J. Zhang, S. K. Kasera, and J. Lepreau, “The flexlab approach to realistic evaluation of networked systems,” in *NSDI*, 2007.

[42] E. F. Gouveia and S. Wahle, “Public deliverable d2.2 approach to technical infrastructure of the panlab project,” 2008.

[43] F. C. Benjamin Quétier and Vincent Neri, “Selecting a virtualization system for grid/p2p large scale emulation,” in *Proceedings of the EXPGRID*, 2006.

[44] P. V.-B. Primet, “INRIA, proposals for virtualization of the network in g5k.”

[45] T. Miyachi, K.-i. Chinen, and Y. Shinoda, “Automatic configuration and execution of internet experiments on an actual node-based testbed,” in *TRIDENTCOM ’05: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 274–282.

Table II
Features of software for cluster testbeds.

Feature	Grid'5000	ModelNet
F1.Control of the experiment's environment	Grid'5000 provides a controlled environment with hosts running on reserved resources. MPLS tunnels provide isolation and QoS guarantees. Different experiments can although interfere with each other.	ModelNet provides a controlled network environment which is implemented by the "core routers" that emulate a network model.
F2.Experiment clock and event scheduling mechanisms	Global clock not provided but applications like NXE can provide event scheduling functionality.	Global clock not provided. No event triggering service.
F3.Separation of control, measurement and experiment planes	Different VLANs are supported for control and experimentation.	Not defined. The experiment uses a different address space than the one used for controlling the hosts.
F4.Storage facilities	Storage is provided through NFS mounts. Experiments are registered centrally to a database for resource allocation purposes.	No storage facilities provided. The emulated network is described in XML files.
F5.Use of programming interfaces	Although OAR does not officially expose an API, a tool under development called CoRDAGe can provide a XML-RPC interface.	The emulated network topology is described in XML files. A remote API for controlling the "core routers" is not provided.
F6.Heterogeneity of technologies (e.g., ability to use different OS)	Hosts can run different OS, traffic generators, simulators etc. Custom hardware can get connected to the LAN.	The network emulator part of ModelNet ("core routers") has to run FreeBSD whereas the edge nodes can run different OS, traffic generators, simulators etc. Real hardware can get connected to the LAN. The use of virtualization inside nodes is supported on Linux, Solaris and FreeBSD.
F7.Clean reconfiguration	Hosts can be reinstalled with custom OS and rebooted.	No support. It is left to the researcher.
F8.Virtualization of nodes and links	As different OS images are supported common host virtualization software can be used. Recent work of Quétier et al. used Vserver [43] and relevant work in progress is HIPerNET [44].	Virtual hosts with different OS can act as edge nodes. Applications running on them can access the emulated network by using specific source and destination IP addresses.
F9.Resource utilization monitoring	Yes, the reservation of resources and the utilization of network links is monitored. Specialized tools like Kaspied provide more details.	No support. It is left to the researcher.
F10.Use of free and open source software	Yes but split into several tools without a central point for downloads.	Open source software available for download.
F11.Authentication and Authorization	Based on LDAP accounts.	No strict model. For automated deployment to edge nodes public key cryptography can be used for authentication (through use of daemons like authd).
F14.Repeatable experiments	Repeatable experiments are possible especially if a researcher uses an automated tool for experiment deployment.	Repeatable experiments are possible especially if the researcher uses an automated tool for experiment deployment.
F15.Extensibility	Extensibility is theoretically possible although there are concerns about programming interfaces.	Yes, the software can be extended as source code is available.
F16.Automation and rapid reconfiguration	Several tools are available for automated deployments Kadeploy, Katapult etc.	Under certain assumptions, tools for automated execution of an application on multiple virtual nodes with a single command are available.
F17.Adjustable level of realism	Yes, based on the ability to use different OS, traffic generators, simulators etc. on real or virtual hosts.	WAN traversal is always emulated by the "core routers" and does not include real routers (hardware or software). All traffic is routed along the shortest path, without emulation of routing protocols.
F18.Scalability for large experiments	Yes. Currently thousands of nodes are supported.	Yes, the emulated topologies can scale up to 1000 of nodes.
F19.Accurate measurements (nodes and links)	Measurements during experiments are left to the researchers.	Measurements during experiments are left to the researcher.
F12.Distinction of roles	A user has full privileges (root) on the reserved resources.	A user needs full privileges (root) on the testbed.
F13.Remote access	Yes, researchers can control the testbed remotely.	Yes, researchers can control the test bed remotely.

Table III
Features of software for cluster testbeds.

Feature	SpringOS	Emulab
F1.Control of the experiment's environment	SpringOS provides a controlled environment as it runs on top of a dedicated cluster with dedicated links.	Emulab provides a controlled environment as it runs on top of a dedicated cluster with dedicated links.
F2.Experiment clock and event scheduling mechanisms	The researcher has to configure NTP synchronization of nodes manually but event triggering services exist using a centralized or distributed model.	Global clock not provided but hosts are NTP synchronized. A powerful event launching system is provided with ability to schedule events statically and dynamically.
F3.Separation of control, measurement and experiment planes	Separation of control and experimental networks using VLANs. (nodes are required to have at least two NICs.	Separation of control and experimental networks using VLANs.
F4.Storage facilities	Is left to the researcher. An ftp server is needed for disk images. The descriptions of experiments are not stored in a central location but in plain files. The resources of the testbed are also stored in a file accessed by the resource management daemon.	User data stored centrally through a network file system (NFS). The descriptions of experiments are stored in a database.
F5.Use of programming interfaces	Work towards the creation of a standard API was referenced in [45] but the results remain unclear.	A XML-RPC interface is provided that supports creation, modification and termination of experiments as well as rebooting and configuring nodes and change of link characteristics.
F6.Heterogeneity of technologies (e.g., ability to use different OS)	Experimental nodes can run FreeBSD or Linux but not Windows. Therefore standard software such as traffic generators can be used in these nodes. The manager node must run Linux.	Hosts can run different OS, traffic generators, simulators etc. Real hardware can get connected to the LAN.
F7.Clean reconfiguration	Yes, nodes can be rebooted and reinstalled (through the snmpmine and ni tools).	Yes, nodes can be rebooted and reinstalled.
F8.Virtualization of nodes and links	Virtual nodes can be used but their complete control through SpringOS has to be developed further.	The use of virtualization inside nodes is supported out of the box on FreeBSD (although running virtual machines with different OS inside a host seems possible). Hosts can access also virtual links that are implemented on top of physical links.
F9.Resource utilization monitoring	Although the existence of a traffic and node state monitor is referenced the required software is not offered and deployment is left to the researcher (e.g., through snmp). The allocation of nodes to different experiments is accessible through the resource management daemon.	There is basic support for monitoring CPU usage, network output and ssh/serial port.
F10.Use of free and open source software	Yes, actively supported and maintained but English documentation is not complete.	Yes, actively supported and maintained.
F11.Authentication and Authorization	Based on username/password pairs.	Based on SSH keys and X.509 certificates.
F14.Repeatable experiments	Repeatable experiments are possible.	Repeatable experiments are possible.
F15.Extensibility	The availability of the source code provides some extensibility that is limited by the lack of documentation and programming interfaces.	The availability of the source code, detailed documentation and programming interfaces (XML-RPC API back-end) provides excellent extensibility.
F16.Automation and rapid reconfiguration	Several features for automated configuration, command execution are available.	Several features for automated configuration, command execution are available.
F17.Adjustable level of realism	A realistic environment for experimentation can be recreated, although it is limited by the lack of Windows OS support.	Based on the heterogeneity of the components that can be used obtaining different levels of realism seems possible.
F18.Scalability for large experiments	Reasonably scalable but the use of virtualization would permit scaling beyond the limits that are imposed by the availability of physical resources.	Yes, especially if virtual hosts and links are used.
F19.Accurate measurements (nodes and links)	No support. Is left to the researcher.	Emulab supports traffic monitoring by running tcpdump on intermediate nodes. Further measurements are left to the researchers.
F12.Distinction of roles	A user is associated to a project. The distinction between different roles and user's permissions remains unclear.	Several roles are defined (project leader, group leader, local root and user) and implemented using standard Unix users and groups.
F13.Remote access	Yes, researchers can control the testbed remotely.	Yes, researchers can control the testbed remotely.

Performance Analysis of Time Synchronization Protocols in Wireless Sensor Networks with Regular Topologies

José A. Sánchez Fernández, José F. Martínez Ortega, Ana B. García Hernando, Lourdes López Santidrián
Dpto. de Ingeniería y Arquitecturas Telemáticas. E.U.I.T. de Telecomunicación
Universidad Politécnica de Madrid
Campus Sur UPM. Cta. de Valencia, km.7. 28031 Madrid, Spain
e-mail: {jsanchez, jfmartin, abgarcia, llopez}@diatel.upm.es

Abstract— **Time synchronization in wireless sensor networks is an essential issue in their operation. The synchronization is deeply influenced by network size and complexity. System dynamics and algebraic graph theory provide the suitable mathematical framework to describe the dynamical and topological features of complex networks. These features are very useful to understand overall aspects of network time evolution, in particular the ability to achieve steady synchronization states. In this paper, we apply the mathematical tools provided by theory to assess the ease of synchronization of a wireless sensor network with regular topology, initially designed to support surveillance applications. From these theoretical results, the research work described will focus on the performance analysis of different time synchronization protocols that allow network nodes to share a common global time, either with the diffusion of a master reference timestamp or with the consecutive exchange of local timestamps among neighbor nodes, in order to achieve a global dynamical consensus.**

Keywords— wireless sensor network; synchonization protocol; spectral graph theory; consensus dynamics; surveillance application.

I. Introduction

For numerous wireless sensor network (WSN) applications, e.g., localization, security or surveillance, where event detection and reporting is a usual task, time synchronization is a major issue. Different sources of unreliability, and also the complexity and size of the WSN affect the performance of different synchronization methods. Certain knowledge of the mathematical models proposed for the analysis of complex networks is needed to understand and solve the underlying problems that could arise. Also, these models provide a good assistance in the design of reliable synchronization protocols.

These mathematical models are directly inspired by those historically developed for complex biological, chemical or physical systems, which can be described in terms of interactions among mutually coupled oscillators. A dynamical analysis of many of these systems reveals important analogies regarding WSN performance, since the time evolution of a single sensor, acting as a node of the whole network, can be modeled as a simple oscillator interacting with the physical environment and the rest of the network.

System Dynamics provide the description of time evolution of the state variables of nodes, reporting useful information about the stability of the network states, while Algebraic Graph Theory is a valuable help to visualize the topology of the network. The study of values of some graph invariants, e.g., the eigenvalues of matrices that describe the connections among network nodes, is very useful to understand the conditions of synchronizability of the whole network. The main purpose of this paper is to apply some of these mathematical tools to a specific WSN with regular topology, initially designed to support surveillance applications; these results will establish the synchronization ability of the network topology chosen. As a extension of [1], alternative synchronization methods will be tested and compared.

The outline of the paper is given as follows. Section II summarizes main theoretical research done about synchronization of complex networks composed of coupled oscillators. Section III describes the approximation to physical time made by hardware oscillators in WSN nodes. Section IV introduces the mathematical model that describes the evolution of a network composed of mutually coupled oscillators, and some basic concepts of Algebraic Graph Theory, which are needed in the analysis of time synchronization. Section V presents the generic results that establish the conditions to reach the synchronization in a complex network. Section VI shows the surveillance application domain, the underlying WSN and the requirements of its synchronization. Section VII discusses the results provided by the application of the mathematical models to state the conditions of a better synchronization for the WSN. Section VIII shows the performance of different synchronization methods on the scenario described above. Finally, Section IX summarizes the conclusions and provides ideas for possible future works.

II. Literature survey

Historically, different but related mathematical models have been proposed to study the synchronization of complex systems, which are described through networks composed of coupled oscillators. Some of the main references in this research field are discussed next. Mirollo, Watts and Strogatz studied the spontaneous synchronizability of biological systems composed of globally coupled identical oscillators [2][3], showing the conditions that lead to the stability of the synchronization state. They analyzed the influence of the traditionally called “small-world effect” on network synchronization, i.e., the addition of some random links to regular lattices, acting as a kind of shortcut connection, enhance the synchronization capabilities of the network.

Chemical systems were studied by Kuramoto [4], who proposed a mathematical model for arbitrary nonlinear phase-coupled oscillators; he showed that the ability to reach a synchronization state relies on the coupling strength among oscillators. Barahona and Pecora studied the synchronization of complex networks composed of identical oscillators through a linear model [5][6], and they showed the strong dependence of the synchronization state to some topological parameters. Barbarossa, Celano and Scutari proposed an extension of the Kuramoto model applied to WSN [7][8][9], which is designed as a system composed of phase-coupled oscillators with nonlinear coupling. They studied the application of distributed synchronization algorithms, and also they showed the impact of synchronization in network overall energy consumption.

In these models, each network node, e.g., a network device including some sensors that collect data from its physical environment, is characterized by a set of state variables. Time evolution of these variables, and coupling interactions among nodes, are described by a system of differential equations. The whole network is modeled with a graph, which includes the information about network topology, basically the existence of communication links between each pair of nodes.

Since the graph is constructed by a set of matrices (adjacency, incidence and laplacian matrices), it is possible to apply Algebraic Graph Theory, a well established field of Discrete Mathematics [10][11][12][13]. Results obtained establish the requirements of synchronizability of the whole network. These requirements rely on a set of inequalities, connecting different parameters, strongly dependent on the overall topology of the network [6][14].

These models are the formal basis in the development of synchronization protocols for WSN. Several proposals have been done, and excellent surveys about this topic have been previously published [15][19]. These surveys present pertinent remarks about challenges and constraints in synchronization protocol design. Also, they include exhaustive classifications of protocols, based on clock models used (constant frequency, bounded frequency deviation, drift-constraint) [16][17][18][19], synchronization classes (external vs. internal, probabilistic vs. deterministic, permanent vs. by request, complete vs. partial) [15][16][19], synchronization techniques (unilateral, bilateral, cyclic, by broadcast correction) [16][17][19], and models of interaction among network nodes (unicast vs. multicast, symmetric vs. asymmetric, explicit vs. implicit) [15][16][19]. Critics of features and performance of the synchronization protocols proposed are also included. It is not aim of this brief summary to discuss these details; the reader interested in these topics is encouraged to address to the above references.

The research work described in this paper focuses on synchronization methods that allow network nodes to share a common global time, either with the diffusion of a master reference timestamp or with the consecutive exchange of local timestamps among neighbor nodes, in order to achieve a global dynamical consensus. These approaches are adopted in some proposals, e.g., [8][20][21]. The synchronization algorithms will be applied to a static WSN with a selected regular topology. Next two sections will focus on the description of formal expression of the time measured by a WSN, and also on the mathematical models proposed to study the time synchronization.

III. Physical Time and Physical clocks

First of all, it will be appropriate to describe the way that the measurement of physical time is made by WSN nodes. This estimation must be coherent with the mathematical model introduced later, in Section IV.

In a WSN, its nodes are usually equipped with a computer clock assisted by a hardware oscillator. Every node i (i= 1,…, N) implements a local approximation $x_i(t)$ to an external source of physical time t, e.g., based on Universal Time Coordinated (UTC) [16][19]. The approximation can be expressed as a function $x_i(t)$: $\mathbb{R} \rightarrow \mathbb{R}$, which is

$$x_i(t) = x_i(0) + K\int_0^t \omega(\tau)d\tau. \tag{1}$$

This function depends on the oscillator frequency, $\boldsymbol{\omega}(t)$, and of a scale constant, K. Initially, $\boldsymbol{\omega}(t)$ is supposed to be equivalent in all nodes. $x_i(t)$ is recorded in a register, which is updated through the oscillator interrupt cycles. The clock accuracy, and therefore its energy consumption, are proportional to the frequency $\boldsymbol{\omega}$.

Ideally, with an appropriate selection of K and $\boldsymbol{\omega}(t)$, $x_i(t)$ = t. However, there are unavoidable sources of error, which are due, mainly, to a limited accuracy of hardware clocks, and also to instabilities of the physical environment. Variations of physical magnitudes, e.g., temperature, or even internal changes in power supply can affect the performance of the clock. The clock absolute *offset* is defined as the difference $x_i(t) – t \neq 0$; the relative offset between two clocks is the difference $x_i(t) – x_j(t)$. The clock *skew* is the difference of its local frequency with respect to the external reference frequency, $\boldsymbol{\omega}(t) – \boldsymbol{\omega}_{ref}$; and the clock *drift* is the variation of this frequency, i.e., its derivative with respect to time, $\boldsymbol{\omega}'(t)$. Figure 1 shows the effects of clock skew and drift in real clocks.

These sources of inaccuracy induce a deviation of physical time recorded at the local clock of every WSN node with respect to the external time reference, and also with other local clocks. Usually, clock accuracy is reported in technical specifications of hardware clocks, given by the manufacturer [15][17]. So, although clock deviation cannot be completely removed, it can be properly bounded. If the required synchronization accuracy is small with respect to the cumulative effect due to frequency fluctuations, it is possible to assume that $\boldsymbol{\omega}(t)$ is approximately constant. If frequency fluctuations are not negligible but they are known, the bounds are given by the next inequalities [16][19]:

$$1-\rho \le \frac{dx_i(t)}{dt} \le 1+\rho, \qquad (2)$$

where $\boldsymbol{\rho}$ is the maximum skew rate specified by the manufacturer. A clock never can stop or run backwards, so $\boldsymbol{\rho} > -1$. The effect of drift could be ignored if frequency fluctuations are small compared with the required synchronization accuracy. The presence or lack of these sources of error and related bounds establishes the clock model to deal with. Assuming the existence of clock skew and drift, $x_i(t)$ can be expanded by its Taylor series [17], in order to simplify the clock model, as

$$x_i(t) = \alpha_i + \beta_i t + \gamma_i t^2 + ... \qquad (3)$$

So, the coefficient α_i can be associated with the initial offset of clock $x_i(t)$; the linear parameter, β_i, with the skew (1 ± $\boldsymbol{\rho}$); and the quadratic parameter, γ_i, with a deviation from the linear behavior. These parameters can be estimated by statistical calculations. From (2) and (3), assuming a linear behavior in clocks, $x_i(t)$ can be bounded as

$$x_i(0) + (1-\rho)t \le x_i(t) \le x_i(0) + (1+\rho)t. \qquad (4)$$

Finally, two clocks, $x_i(t)$ and $x_j(t)$, can be related to their relative offset, α_{ij}, and skew, β_{ij} [15], as

$$x_i(t) = \alpha_{ij} + \beta_{ij} x_j(t). \qquad (5)$$

After consecutive requests and receptions of values of $x_j(t)$, node i can estimate the relative offset and skew with node j. Equation (5) can be extended to a group of connected nodes,

$$x_i(t) = \sum_{j\in N_i} (\alpha_{ij} + \beta_{ij} x_j(t)), \qquad (6)$$

where N_i denotes the neighborhood of node i, i.e., the set of nodes connected to it. Since the difference between time records of any pair of nodes, $x_i(t) - x_j(t)$, trends to increase as t grows, the synchronization will try to cancel this difference, or, at least, to minimize it. The general formulation of the synchronization problem consists in a direct adjustment of the value $x_i(t)$ by the application of a synchronization algorithm. To proceed, each node must receive one or various packets containing external time references, and the algorithm should infer some additional estimations. Due to inaccuracies in the operation of WSN, some values need to be calculated ad hoc, e.g., the packet round trip time between different nodes, by timing exchanges of specific messages [15]. Other values, as the expected average delay in internal processing tasks, could be statistically or randomly estimated. Finally, the synchronization algorithm should not suppose a substantial overload to WSN operation. Next Section presents the dynamical model proposed to describe the behavior of complex networks. It will be shown that this formulation is consistent with the formal description of time measure given above.

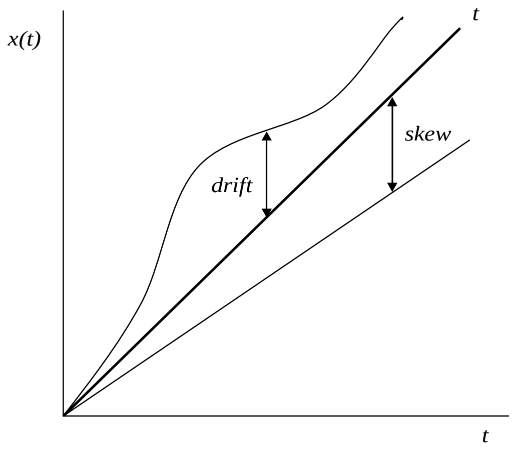


Figure 1. And ideal clock and two real clocks, with an initial offset equal to zero, and $x(0) = 0$. Effect of skew and drift shows, respectively, linear and nonlinear deviations from ideal reference time.

IV. WSN Dynamical Model

Among the formal models proposed to describe complex network dynamics, the classical development of Barahona and Pecora has been selected [5][6], due to its generality and simplicity. This model considers a network composed of N nodes, each one including some sensors of the physical environment. The i-th node ($i = 1,\ldots,N$) takes different measures of M specific events, expressed by its state variable vector, $\boldsymbol{x}_i(t) = (x_{i1}(t),\ldots,x_{iM}(t))$.

The sensors act as mutually coupled oscillators, since they periodically communicate with other sensors falling into their coverage radius, sending and receiving some of the collected data or synchronization timestamps. Therefore, the nodes can adapt their state variables evolution according to data received from others, e.g., the time values of their clocks, to reach the synchronization.

The dynamical system present at each node evolves according to the following system of first order differential equations, as proposed in [6]:

$$\frac{d\boldsymbol{x}_i}{dt} = \boldsymbol{F}(x_i(t)) - \sigma \sum_{j=1}^{N} L_{ij} \boldsymbol{H}(x_j(t)), i = 1,..., N. \qquad (7)$$

$\boldsymbol{F}(x_i(t))$ is a vector function of dimension M, that expresses the dynamics at each node, i.e., the evolution of its M state variables $\boldsymbol{x}_i(t)$, and $\boldsymbol{\sigma}$ is an overall coupling strength, supposed to be identical in all network nodes. The M-vector output function $\boldsymbol{H}(x_j(t))$ of the state variables of each oscillator represents the coupling among oscillators. Finally, L_{ij} are the components of a certain N x N connection matrix $\mathbf{L}$, which specifies the existing connections among nodes. $\mathbf{L}$ is symmetric, since it is assumed that the network nodes have nondirectional links.

Results obtained in [6] show the generic conditions that vector functions $\boldsymbol{F}(x_j(t))$, $\boldsymbol{H}(x_j(t))$, and matrix $\mathbf{L}$ must satisfy to reach an stable synchronization state in the network of coupled oscillators whose dynamics is modeled by (7). It has been shown that this stability depends, mainly, on selected eigenvalues belonging to the spectrum of the connection matrix **L** [6], which is invariant under permutations of network node labels. Therefore, it is essential to establish the elements L_{ij} of **L**.

To proceed, system of equations expressed in (7) can be conveniently simplified to facilitate the next analysis. It is possible to reduce the number of state variables in each node from *M* to 1, since the local time to be measured in each node from its internal hardware oscillator, $x_i(t)$, will be the only state variable to consider further. In that case, (7) gets a more familiar fashion:

$$\frac{dx_i}{dt} = K\omega(t) - \sigma\sum_{j=1}^{N} L_{ij} H(x_j(t)), \tag{8}$$

where $\boldsymbol{\omega}(t)$ represent the oscillation frequency, supposed to be identical in all nodes. It is interesting to note that, if coupling among nodes is nonexistent, the sum contained in the right member of (8) is equal to zero,

$$\sum_{j=1}^{N} L_{ij} H(x_j(t)) = 0, \tag{9}$$

and (8) is transformed in

$$\frac{dx_i}{dt} = K\omega(t), \tag{10}$$

which is formally equivalent to (1). Thus, the simplest case of dynamics at every node (nonexistent coupling) leads to the initial estimation of time made by network nodes. It could be considered as an argument of validity of the dynamical model presented above.

Once the network is synchronized, (8) must be identical for all the nodes. This is assured if the sum of its right member is constant:

$$\sum_{j=1}^{N} L_{ij} H(x_j(t)) = K. \tag{11}$$

In the simplest case, this sum is supposed null [6]. The connection matrix **L**, then, is restricted to have zero sum rows:

$$\sum_{j=1}^{N} L_{ij} = 0. \tag{12}$$

With this constraint, it is possible to obtain the elements L_{ij} of **L**. To that end, it will be useful to introduce before some basic concepts of Algebraic Graph Theory [10][11]. A network can be modeled by a graph $U{=}U(V{,}E)$ composed of *N* nodes or vertices *V*, labeled from 1 to *N*, and a set of connections or edges *E* among them. The number of edges can vary from 0 (no nodes are connected) to $N(N\text{–}1)/2$ (every node is connected with all the others). A graph is represented by its adjacency matrix **A**, a symmetric $N \times N$ matrix where A_{ij} = 1 if the nodes *i* and *j* are connected, and A_{ij} = 0 otherwise. Also, the components of its main diagonal are defined as A_{ii} = 0. The study of invariants of the adjacency matrix (i.e., its spectrum properties) has been widely studied [10][11].

The degree d_i of the node *i* is the sum of the number of edges connecting it to other nodes, which can be obtained from the sum of the values belonging to the *i*-th row of the adjacency matrix, i.e.,

$$\sum_{j=1}^{N} A_{ij} = d_i, \tag{13}$$

which suggests to form a new matrix **D**, called valency or degree matrix, whose main diagonal elements D_{ii} are equivalent to the *i*-th sums in (13). It is possible to build a new matrix, the laplacian matrix, as **L** = **D** – **A**, which clearly fulfills the requirement derived from (12). The elements L_{ij} of **L**, at last, can be expressed as

$$L_{ij} = \delta_{ij} D_{ij} - A_{ij}. \tag{14}$$

Spectrum properties of **L** have been studied, as well [12, 13]; it would be useful to obtain its eigenvalues through analytical expressions, but, in general, this is not possible. The characteristic equation of **L**, $det(\mathbf{L} - \gamma\mathbf{1}) = 0$, gives a polynomial that is difficult to solve for large values of *N* (also, it is due to a strong dependence of eigenvalues with little variations of the polynomial coefficients).

However, cyclic graphs, associated with networks with ring topology, generate adjacency and laplacian matrices with circulant structure [6][11][12][22], whose eigenvalues are well known. In a circulant matrix, each row is a cyclic permutation of the first row. A ring network of *N* nodes, where each node is connected exclusively with its two neighbors, generates a cyclic graph called *cycle* ($\mathbf{C}_N$). In such graphs, adjacency and laplacian matrices adopt the next structure:

$$\mathbf{A}(\mathbf{C}_N) = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \quad \mathbf{L}(\mathbf{C}_N) = \begin{pmatrix} 2 & -1 & 0 & \ldots & 0 & -1 \\ -1 & 2 & -1 & \ldots & 0 & 0 \\ 0 & -1 & 2 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 2 & -1 \\ -1 & 0 & 0 & \ldots & -1 & 2 \end{pmatrix}$$

And eigenvalues of $\mathbf{L}(\mathbf{C}_N)$ are given by [6][22]

$$\gamma_i = 2\left(1 - \cos\frac{2\pi(i-1)}{N}\right). \tag{15}$$

In general, when a node is connected with its 2*k* neighbors, the resultant graph, known as *k*-cycle ($k\mathbf{C}_N$), gives the next Laplacian eigenvalues [6][22]:

$$\gamma_i = 2\left(k - \sum_{j=1}^{k} \cos\frac{2\pi(i-1)j}{N}\right). \tag{16}$$

Figure 2, adapted from [6], shows the eigenvalues of *k*-cycles, for $k = \{1,..,4\}$ and a network of $N = 100$ nodes. For *N* even, the graphics are symmetric from the ordinate line at $N/2 + 1$. For each *k*, there are *k* local maximum values, and maximum degeneracy of eigenvalues is 2*k*. The analytical expression for Laplacian eigenvalues has been shown for this regular topology. Section VI will show the importance of such a topology from the application domain point of view. Also, Section V will explore the relationship among eigenvalues of **L**, network topology and time synchronization.

Meanwhile, it will be suitable to study the synchronization convergence properties of (8). In the simplest coupling among nodes, the variation of $x_i(t)$ with time is a linear combination of its own value and the values $x_j(t)$ of the nodes connected to it. So, $H(x_j(t)) = x_j(t)$, and (8) is transformed in

$$\frac{dx_i}{dt} = -\sigma \sum_{j=1}^{N} L_{ij} x_j(t). \tag{17}$$

Introducing the definition of the elements L_{ij}, given in (14),

$$\frac{dx_i}{dt} = -\sigma \sum_{j=1}^{N} (\delta_{ij} D_{ij} - A_{ij}) = -\sigma \sum_{j \in N_i} (x_i(t) - x_j(t)), \tag{18}$$

where N_i denotes the neighborhood of node *i*. Equations (17) and (18) can be expressed in a more compact form,

$$\frac{d\boldsymbol{x}}{dt} = -\sigma \mathbf{L} \boldsymbol{x}(t). \tag{19}$$

This set of differential equations is known as *consensus dynamics* [9][23][24][25]. The state variable of each node evolves in time as a linear combination of its own state and the states of coupled nodes. If the state vector $\boldsymbol{x}(t)$ is initialized with the local measurement of time in nodes, $\boldsymbol{x}(0) = \boldsymbol{x}_0$, and the network is connected, $\boldsymbol{x}$(t) converges to the average consensus vector

$$\boldsymbol{x}(t) \to \frac{1}{N} \boldsymbol{1^T}\, \boldsymbol{x}_0 \boldsymbol{1}. \tag{20}$$

where $\boldsymbol{1}^T = (1, 1, \ldots, 1)$. That is,

$$x_{\boldsymbol{i}}(t) \to \frac{1}{N} \sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} x_{0j}, \tag{21}$$

for $t \to \infty$.

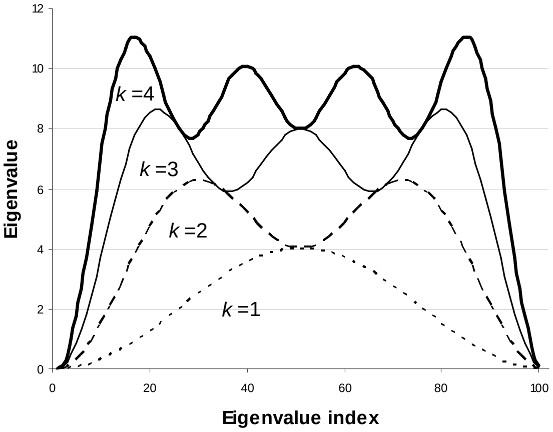


Figure 2. Eigenvalues of *k*-cycles for *k*={1,2,3,4}in a network of *N*=100 nodes (adapted from [6]). It is easy to visualize the simmetry of graphics from the ordinate line at $i = N/2 + 1$, the equivalence between the number of local maxima and *k*, the maximum degeneracy of eigenvalues, equal to 2*k*, and the growth of maximum eigenvalues with *k*.

In other words, all local clocks converge to their initial average value, with independence of the coupling strength, σ. To visualize this convergence, (18) can be approximated with the next iterative expression:

$$x_{i,n+1} = x_{in} - \sigma \Delta t \sum_{j \in N_i} (x_{in} - x_{jn}). \tag{22}$$

In this approximation, *n* represents the *n*-th iteration step, and Δt is the interval of time elapsed between consecutive iterations. Figure 3 shows the convergence of (22), for a ring network example of $N = 5$ nodes, coupling strength $\sigma = 1$, interval time $\Delta t = 0{,}1$ sec., and exclusive coupling among nearest neighbor nodes. It is clearly shown that the local clocks converge to the constant average time value given in (21).

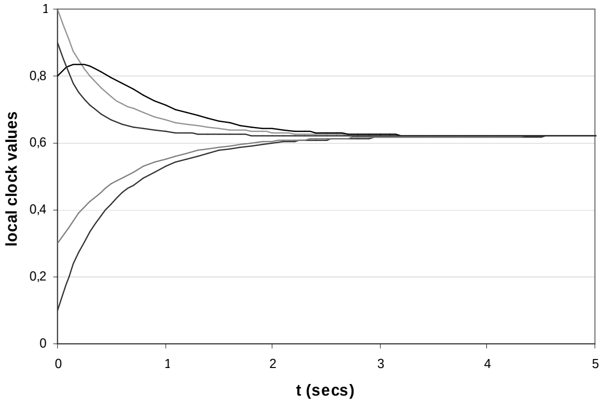


Figure 3. Consensus dynamics. Synchronization convergence example, in a ring network with $N = 5$ nodes, coupling strenght $\sigma = 1$, and interval time between succesive iterations $\Delta t = 0{,}1$ sec. At $t = 5$ secs., relative offset among clocks is $0{,}5 \times 10^{-3}$ sec.

To consider the increment of local clocks $x_j(t)$ with time, the frequency $\boldsymbol{\omega}$ must be taken into account, as follows,

$$\frac{dx_i}{dt} = K\omega - \sigma \sum_{j \in N_i} (x_i(t) - x_j(t)). \tag{23}$$

Synchronicity is acquired in a time varying form [23], and all the $x_i(t)$ converge to

$$x_i(t) \to \frac{1}{N} \sum_{j=1}^{N} K\omega t = K\omega t, \tag{24}$$

for $t \to \infty$**.** Adding the effect of frequency to the convergence steps expressed in (23), Figure 4 shows the convergence of this time-varying consensus dynamics, for the same type of network as above, with $K\boldsymbol{\omega}$ chosen to be 1, which gives a valid measure of time t, i.e., $x_i(t) \to t$.

Consensus dynamics, in both forms (constant or time varying), assures the stability of the global synchronization state. The time needed to get a full synchronicity is infinite, but in a real network the limited accuracy of local clocks influences the elapsed time to get a sufficient synchronization. This feature is present in different proposals of synchronization methods [8][9][21].

V. Generic Conditions of Synchronizability for a WSN

This Section will use some results presented in [6][13][14], very useful from the synchronization point of view, to extract generic conditions of synchronizability for a WSN.

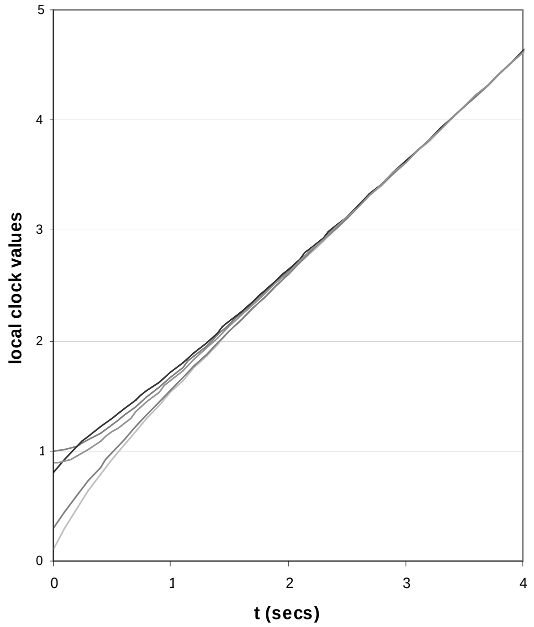


Figure 4. Time varying consensus dynamics. Synchronization convergence example. Ring network with N = 5 nodes, $\sigma = 1$, $\Delta t = 0{,}1$ sec, and $K\boldsymbol{\omega} = 1$. At $t = 4$ sec, relative offset among clocks is 0,2 x 10^{-2}sec. Convergence rate is independent of $\boldsymbol{\omega}$.

Laplacian matrix **L** is a positive, semi-definite matrix, with a maximum of N different nonnegative eigenvalues, γ_i, $1 \le i \le N$. They can be arranged in the sequence $\gamma_1 \le \gamma_2 \le \ldots \le \gamma_M$, where $M \le N$. By construction of **L**, its smaller eigenvalue, γ_1, is always zero, and it is associated with the eigenvector whose components are all equal to 1. The next larger eigenvalue, γ_2, is called algebraic connectivity, and is different from 0 if, and only if, the graph is connected. If γ_2 is close to 0, the graph can be easily split in subgraphs by the deletion of a few edges.

The algebraic connectivity, thus, plays a special role on the synchronizability of the network. Large values of γ_2 are associated with networks with a good connectivity among nodes, which improves the synchronization. On the other hand, if $\gamma_2 \to 0$, the network cannot reach synchronization easily. It has been shown that the synchronization time in the consensus dynamics context, given by (19), is proportional to γ_2^{-1} [25].

As it is noticed in [14], eigenvalue γ_2 by itself does not provide information about network topology. Nevertheless, some bounds for γ_2 have been found, which include the desired topological information. These bounds are expressed in the following set of inequalities [6][14]:

$$\gamma_2 \le \frac{N}{N-1}\delta \le \frac{N}{N-1}\Delta \le \gamma_M \le 2\Delta, \tag{25}$$

$$\frac{4}{ND} \le \gamma_2 \le \frac{Nd_i}{N-1}, \forall i, \tag{26}$$

where d_i is the degree of node i, δ is the minimum degree of the graph (the minimum value of the node degrees) and Δ is the maximum degree of the graph (the maximum value of the node degrees). The distance between two nodes, i and j, is defined as the minimum number of edges to traverse from node i to node j. The graph diameter, D, is the maximum value of distances. Networks with large values of N and D will give a small lower bound for γ_2. However, if these values are small, γ_2 will be greater and the network will synchronize better.

Also, the quotient γ_M/γ_2 should be as small as possible. The lowest value is 1, only possible if every node is connected with the rest of nodes. From (25) and (26), it can be shown [6][14] that

$$\frac{\Delta}{\delta} \le \frac{\gamma_M}{\gamma_2}. \tag{27}$$

If the quotient γ_M/γ_2 is large (great difference between maximum and minimum degrees of the graph), the synchronization will be difficult. Maximum bounds for this quotient have been obtained [6][14]:

$$\frac{\gamma_M}{\gamma_2} \le \frac{ND\Delta}{2} \tag{28}$$

Thus, if the values of maximum degree Δ, diameter D and number of nodes N can be reduced, the synchronizability of the associated network will be enhanced. Finally, the average distance among nodes, *l*, can also give information about network synchronization properties [6][14]:

$$\frac{1}{\gamma_2} \le \frac{(N-1)l}{2} - \frac{N-2}{4} \tag{29}$$

Networks with small number of nodes and small average distance generate a great value of γ_2, so they can be easily synchronized. Although there are some more bounds for γ_2 and γ_M, involving new topological parameters, e.g., isoperimetric number, clustering coefficient, and edge connectivity [14], the above inequalities show the essential approach to WSN synchronizability.

As conclusion, the combination of Network Dynamics and Algebraic Graph Theory provides useful tools to model a network and to obtain the conditions of its synchronization, which rely on overall network topological parameters. Next Section describes the specific surveillance application scenario of interest for this research, together with a justification on the importance that time synchronization plays in such a deployment.

VI. Application Domain

One of the most important application types that can benefit from a WSN are those related to physical safety and watching of buildings or areas, i.e., surveillance applications. There is a wide range of related services, some of them applied to detect people and vehicle crossings on a virtual perimeter, and to warn about if need be. A network with regular ring topology is suitable to properly cover a virtual closed perimeter.

As part of the work inside the project µSWN (Solving Major Problems in Microsensorial Wireless Networks), financed by European Union VI Framework Program [26], a surveillance application of a closed virtual perimeter has been designed. The application is built through two different application agents, running on two different mote (WSN nodes) profiles, namely perimeter motes and bracelet motes. They will be designated perimeter agents and bracelet agents, respectively.

Perimeter motes are physically deployed covering a virtual perimeter. Some of these motes are equipped with one or two presence sensors (passive infrared sensors, PIR). Every hop between two neighbor perimeter motes is covered at least by one of these sensors. If someone crosses the perimeter, at least one of the PIR sensors detects it and triggers the activation of the corresponding perimeter agent or agents. Let us denote the area covered by the PIR sensors of node *i* as S_i. Let us also denote the position of a person crossing the perimeter as $\mathbf{r}_m$.

Whenever a crossing is signaled to a perimeter agent, i.e., $\exists\, i \;/\; \mathbf{r}_m \in S_i$, this agent tries to find out if the crossing has been caused by someone known or by an intruder. Known people (e.g., staff or authorized clients) wear a bracelet including a mote with a bracelet agent running, while of course intruders do not. Once the presence is known to the perimeter agent, it broadcasts to its neighbors (motes located inside its radio coverage) a specific message, which may include an identification of the PIR actually detecting the crossing or an identification of the node (if need be for the Surveillance algorithm). If a bracelet agent receives the message, it will answer to the enquirer providing its identifier. This identifier will be further forwarded by the perimeter agent to the sink, together with additional data, which may be significant for the algorithm running in the server. If no answer is received by the perimeter agent in a reasonable time, it is assumed that an intruder has crossed the perimeter. Therefore, this event is notified to the sink.

The fixed perimeter nodes and the mobile bracelets are equipped with omnidirectional antennas. Let R_i and R_m be the radio coverage areas of node n_i and of the mobile node, respectively, and let d_{mi} be the distance between n_i and the mobile node. If an unauthorized person crosses the perimeter, an intruder alert will be generated by all nodes n_i where $\mathbf{r}_m \in S_i$. If an authorized person (i.e., a person carrying a bracelet) crosses the perimeter, a bracelet-crossing indication will be generated by all nodes n_i where the following statement is true:

$$(\mathbf{r}_m \in S_i) AND (d_{mi} \le R_i) AND (d_{mi} \le R_m). \tag{30}$$

In that case, also, an intruder alert will be generated by all nodes n_j where the following statement is true:

$$(\mathbf{r}_m \in S_j) AND \lfloor (d_{mj} > R_j) OR (d_{mj} > R_m) \rfloor \tag{31}$$

Figure 5 shows a graphical representation of the influence of these parameters on the described surveillance algorithm. There are some issues that arise in the scenario described above, that lead to the conclusion that time synchronization of perimeter nodes could be highly beneficial. These issues are summarized as follows:

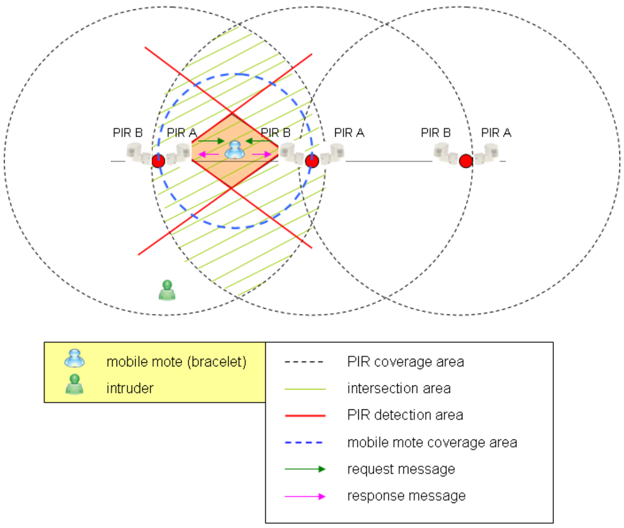


Figure 5. Influence of the different coverage areas of WSN perimeter nodes in the surveillance application algorithm.

- First of all, packets simultaneously generated in different nodes could invest different periods of time to reach the sink. In fact, since power consumption in a WSN is of the utmost importance, many communication protocols implement a low duty-cycle. The nodes are operative and ready to forward information during a small percentage of time in this cycle. Thus, if the packet must traverse several hops, the overall delay may be significant (e.g., tens, hundreds or even thousands of milliseconds).
- On the other hand, it is possible, depending on the combination among the coverage radio of the motes (R_i), the coverage radio of the mobile bracelets (R_m) and the PIR-detection areas (S_i), that two different nodes detect the same physical event (i.e., a particular crossing) and report it to the sink. Also, as shown in (30) and (31), the same crossing event could be notified by different nodes.
- All the above may lead to the reception of more than one warning to the sink (and to the server system beyond it) with a significant time offset but that actually correspond to the same crossing event. This is especially true if all warning messages do not follow the same path to the sink.

If the warning messages were timestamped by the originating motes (e.g., with the local time associated with the crossing event, which was locally reported to the perimeter agent) it would be much easier to detect and eliminate duplicated information, reducing the number of “false positives”. Obviously, if different nodes timestamp their messages and if these timestamp values are compared later in the sink or in the server, it is necessary that those nodes have synchronized clocks with a certain tolerance, in order to decide if two event reports correspond to the same actual real-world event.

It is important to note that, although the detection range of the PIR sensors is not often configurable (and it makes difficult to avoid duplicated detections in some deployment scenarios), the radio coverage of a node usually is. Thus, the logical topology of the WSN may, under certain circumstances, be modified adjusting the transmission power of the nodes, which will reduce or increase the number of neighbors for each mote.

For instance, the increment of transmission power at perimeter nodes, and consequently R_i, leads to a growth in the number of neighbors of each node, reducing the associated graph diameter, *D*, of the surveillance WSN. For $k = 2$ (2-cycle, with four neighbors per node), $D = N/4$ if N is even, and $D = (N - 1)/4$ if N is odd, while for $k = 4$ this diameter value halves, with subsequent effects in WSN synchronization, as shown in (28). The drawbacks of raising R_i are basically twofold: firstly, an increased transmission power implies higher energy consumption, a usually scarce resource in any WSN; secondly, large R_i may cause some inaccuracies when reporting crossings.

For instance, if there is an intruder crossing that is detected by node n_i, and R_i is too large, the corresponding question message may be received by a surrounding bracelet, which is not actually crossing the perimeter, but where the next statement is true:

$$(d_{mi} \leq R_i) AND (d_{mi} \leq R_m). \qquad (32)$$

In that case, a bracelet-crossing event will be erroneously generated. In order to prevent this kind of collateral effects, if R_i is increased, the values of S_i (e.g., by a more precise physical orientation of the PIR sensors) and R_m should be carefully chosen. Thus, it will be possible to minimize the effect of false alarm incidences.

Next Section of this paper applies the results derived from the mathematical model described in previous sections to find adequate values for the WSN logical topology that ease and simplify the time synchronization process in this particular surveillance application scenario.

VII. Conditions of Synchronizability of the Surveillance WSN

In Section V, the mathematical results that link the eigenvalues of **L** and some overall network topological parameters have been shown, mainly in (27) and (28). Therefore, it is possible to apply those results to the surveillance WSN described in Section VI, in order to facilitate its synchronization. Conclusions are summarized as follows [1]:

- The number of nodes, *N*, must be reduced to a value as small as possible, but obviously the connectivity among node *i* and its two neighbors (nodes $i - 1$, $i + 1$) must be always assured to keep the surveillance perimeter properly closed.
- The WSN diameter, *D*, must be also reduced. This could be done increasing the coverage radio of every node *i*, e.g., to get connectivity with nodes $i - 2$ and $i + 2$ as well, which reduces *D* by 1/2. Further reductions of *D* can be obtained increasing the coverage radio of nodes even more, but it could be prohibitive in terms of energy consumption, as explained in Section VI.
- The maximum degree of nodes, $\mathbf{\Delta}$, should not be increased excessively. It leads to the conclusion that the reduction of network diameter *D* and the corresponding increase of the maximum degree $\mathbf{\Delta}$ must be balanced.
- The difference between minimum degree ($\boldsymbol{\delta}$) and maximum degree ($\mathbf{\Delta}$) must be as small as possible. It could be achieved if the physical distance among adjacent nodes is similar (and also if the coverage radio is equal for all nodes), to accomplish $\boldsymbol{\delta} = \mathbf{\Delta}$.

These generic conditions must be verified in specific cases. Since the Laplacian eigenvalues of the ring network selected to support surveillance applications are given by (15), it is possible to obtain an approximation to its maximum eigenvalue, γ_M. Although the values of i are integer, it will be assumed that $i \in \mathbb{R}$. Thus, the usual conditions of the maximum of a function can be imposed:

the first derivative of (15) with respect to *i* must be zero, and its second derivative, negative:

$$\frac{d\gamma_i}{di} = \frac{4\pi}{N}\sin\frac{2\pi(i-1)}{N} = 0, \tag{33}$$

$$\frac{d^2\gamma_i}{di^2} = \frac{8\pi^2}{N^2}\cos\frac{2\pi(i-1)}{N} < 0. \tag{34}$$

Both equations are satisfied by *i* =1 + *N*/2. If this selected value of *i* is included in (15), eigenvalue γ_M is

$$\gamma_M \equiv \gamma_{\frac{N}{2}+1} = 2(1-\cos\pi) = 4. \tag{35}$$

And also, for *i* = 2, eigenvalue γ_2 is

$$\gamma_2 = 2(1-\cos\frac{2\pi}{N}). \tag{36}$$

The maximum eigenvalue $\gamma_{N/2+1}$, remains constant (independent of *N*), and algebraic connectivity, γ_2, trends to zero as *N* grows; the first derivative of (36) with respect to *N* is continuously decreasing, and trends to 0 as *N* grows, as well:

$$\frac{d\gamma_2}{dN} = -\frac{4\pi}{N^2}\sin\frac{2\pi}{N}. \tag{37}$$

The quotient between $\gamma_{N/2+1}$ and γ_2 must be bounded, as it was shown in Section V; the application of (27) and (28) shows, respectively,

$$1 < \frac{\gamma_{\frac{N}{2}+1}}{\gamma_2} < \frac{N(N-1)}{2}. \tag{38}$$

The upper bound for this quotient is not convenient for time synchronization for large *N*, since it grows with *N* at quadratic rate.

The above analysis can be extended to *k*-cycles, which gives similar results. Figure 6 shows, from (16), the dependence of *k*-cycles maximum eigenvalues, γ_M, with respect to the network number of nodes, *N*, for *k* = {1, 2, 3, 4}. These maximum eigenvalues grow with *k*, and they reach a steady value as *N* grows. In fact, γ_M only gets the maximum value given in (35) for *N* even. As a consequence of the previous assumption made, i.e., $i \in \mathbb{R}$, Figure 6 shows that values of γ_M are not strictly constant, although they trend, for large *N*, to the maximum value given in (35).

Figure 7 shows, from (16), the dependence of *k*-cycles algebraic connectivity, γ_2, also called first nonzero eigenvalue (FNZE), with respect to *N*, for *k* = {1, 2, 3, 4}. In all cases, FNZE trends to zero as *N* grows, which is not suitable for WSN synchronization. Finally, Figure 8 shows, from (16) and (38), the spectral bounds for the quotient $\gamma_{N/2+1}/\gamma_2$ in a ring network, with *k* = 1, as functions of *N*. The increasing upper bound for large *N* is not favorable for time synchronization, either.

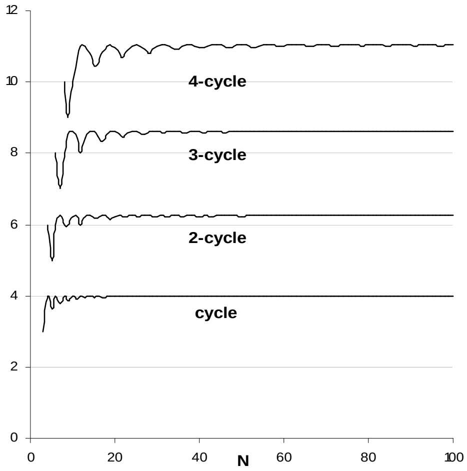


Figure 6. Maximum eigenvalues for *k*-cycles, γ_M , with *k*={1,2,3,4}, as functions of the number of network nodes, *N*. These eigenvalues grow with *k*, and they stabilize for N large.

Although the convergence of the synchronization is initially guaranteed for WSN with ring topology, as it has been shown in Section IV, it can be concluded that this kind of network topology is not a good candidate for time synchronization, except for a reduced number of nodes. The increase of network degree, by the extension of coverage radio in network nodes, does not suppose a valuable help, and also it leads to a significant increment of power consumption. Next Section explores the validity of these theoretical results with the application of various synchronization protocols.

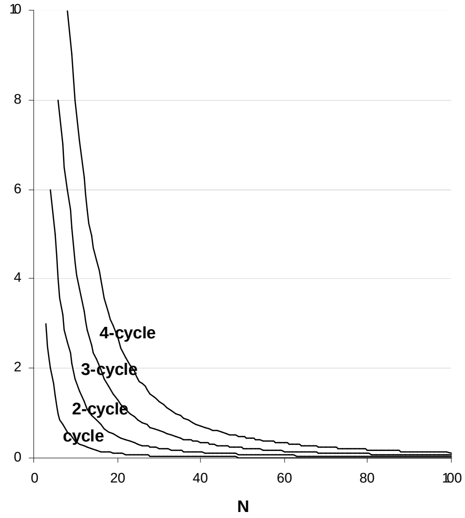


Figure 7. Algebraic connectivity, γ_2, or first nonzero eigenvalue (FNZE), for *k*-cycles, with *k*={1,2,3,4}, as a function of the number of network nodes, *N*. In all cases, FNZE trends to zero as *N* grows.

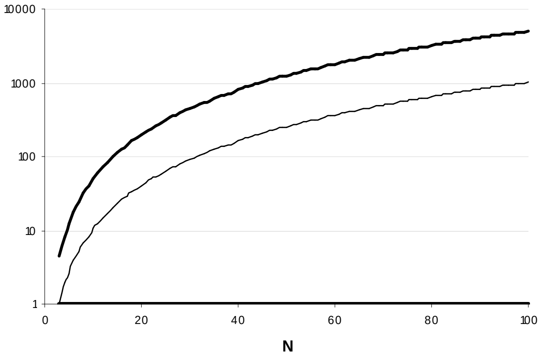


Figure 8. Spectral bounds for $\gamma_{N/2+1}/\gamma_2$ (bold lines) for a WSN with ring topology (cycle, with k = 1), as functions of N, number of network nodes. The curve $\gamma_{N/2+1}/\gamma_2$ remains bounded between 1 and $N(N–1)/2$. The increasing upper bound, for large N, is not adequate for time synchronization.

VIII. Performance of Synchronization protocols in the Surveillance WSN

This Section explores the application and performance of various synchronization algorithms in the surveillance WSN with ring topology. The simulations will be applied to a WSN with a maximum of N = 100 nodes. The initial local clock times are randomly chosen, and the initial maximum offset between each pair of nodes is set up to one second. Figure 9 shows a typical distribution of the initial local clocks offset, with respect to the local clock of sink node, which will be chosen as the master clock in some synchronization algorithms. The unavoidable delay caused by transmission of messages among neighbor nodes and internal processing tasks is supposed constant. All the local clocks have the same angular frequency, and the effects of clock drift and skew are supposed to be negligible with respect to clock frequency during the synchronization stage. This last assumption is reasonable since the clock accuracy is limited. This simplified scenario, which can be further extended, can prove the suitability of different synchronization techniques applied to ideal situations, without the appearance of nondeterministic effects.

Some different synchronization algorithms have been tested. They can be divided in two types: the algorithms that rely on a distributed diffusion and adjustment of local clock values among adjacent nodes, following the time varying consensus model described in Section IV, and the algorithms based on a master clock time reference forwarded to all WSN nodes. In such cases, the synchronization is acquired in a progressive way, after a certain number of synchronization rounds. It is assumed, also, that the elapsed time between consecutive synchronization rounds is constant. The main goal of this Section is to identify the synchronization methods that minimize the number of synchronization steps.

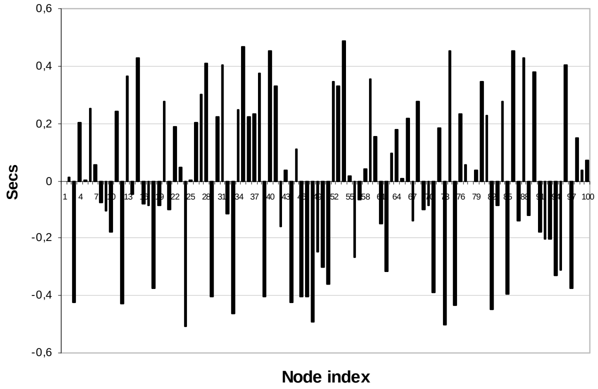


Figure 9. Initial local clocks offset (in seconds) for a network with N = 100 nodes. Initial offsets are relative to the master clock time of sink node. The initial local clock values are randomly chosen, with a maximum dispersion of 1 second among them.

A. Synchronization by distributed diffusion

Following the principles of the consensus dynamics model, shown in Section IV, the first synchronization method to test will be based on the exchange of local clock values among adjacent nodes. An initial broadcast message from the sink node can start the synchronization process. Immediately after the reception of this message, each node sends a message to its neighbors, containing its local clock time value.

Figure 10 shows this exchange process. The node i receives the clock values x_{i+1} and $x_{i–1}$. Since this reception is not simultaneous, let $\boldsymbol{\varepsilon}$ be this temporal difference. The packet transmission delay between nodes is $\boldsymbol{\mu}$. Assuming that x_{i+1} and $x_{i–1}$ has been received at the same period of time T (elapsed time between two consecutive synchronization rounds), it could be considered that the contribution of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\mu}$, are negligible (i.e., $T >> \boldsymbol{\varepsilon}$ and $T >> \boldsymbol{\mu}$) .

Once the nodes have received and recorded these values, each node compares its local clock time, x_i, and the timestamps received by its neighbors, x_{i+1} and $x_{i–1}$. Subsequently, the node adjusts its local clock with the average value from the local time x_i and the received values, x_{i+1} and $x_{i–1}$. Let us explore the synchronization convergence properties of this synchronization algorithm. Each synchronization round can be expressed as

$$x_{i,n+1} = \frac{1}{3}x_{in} + \frac{1}{3}\sum_{j=1}^{N} A_{ij}x_{jn}. \qquad (39)$$

From the definition of laplacian matrix components given in (14), $A_{ij} = \delta_{ij}D_{ij} – L_{ij}$,

$$x_{i,n+1} = \frac{1}{3}x_{in} + \frac{1}{3}\sum_{j=1}^{N} (\delta_{ij}D_{ij} - L_{ij})x_{jn}. \qquad (40)$$

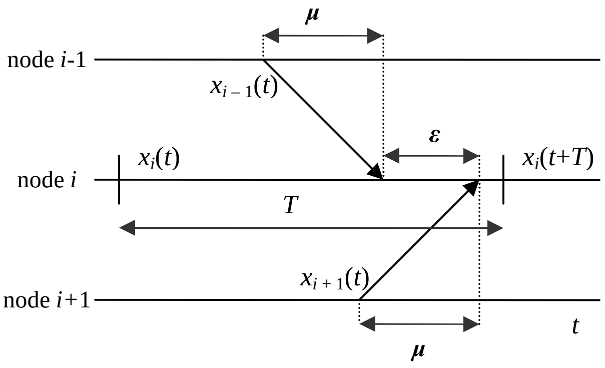


Figure 10. Exchange of local clock values among adjacent nodes. The reception of x_{i-1}, x_{i+1} by node i is supposed to take place into the same period of time, T, interval of elapsed time between two consecutive synchronization rounds .

Rearranging the second member terms of (40),

$$x_{i,n+1} = x_{in} - \frac{1}{3}\sum_{j=1}^{N} L_{ij} x_{jn}, \tag{41}$$

$$\Delta x_{in} = -\frac{1}{3}\sum_{j=1}^{N} L_{ij} x_{jn}. \tag{42}$$

And dividing for $\Delta t \equiv T$, the elapsed period of time between two consecutive synchronization rounds,

$$\frac{\Delta x_{in}}{\Delta t} = -\frac{1}{3\Delta t}\sum_{j=1}^{N} L_{ij} x_{jn} \approx -\frac{1}{3T}\sum_{j=1}^{N} L_{ij} x_{jn}. \tag{43}$$

Equation (43) can be expressed in vectorial form, as

$$\frac{\Delta \boldsymbol{x_n}}{\Delta t} = -\frac{1}{3T}\mathbf{L}\boldsymbol{x}_n. \tag{44}$$

So, it is proved that the synchronization method chosen, from the average values of local clocks of neighbors, follows the consensus dynamics model expressed in (19). This result can be easily extended for a regular k-cycle; the average time value of local clock $x_i(t)$ is calculated from its $2k$ neighbors. In that case, (44) can be generalized as

$$\frac{d\boldsymbol{x}}{dt} = -\frac{1}{(k+1)T}\mathbf{L}\boldsymbol{x}(t). \tag{45}$$

The coupling strength is equal to $[(k + 1)mT]^{-1}$. Adding the increment of local clocks with time, Figure 11 shows the convergence of this algorithm, for ring networks with different number of nodes, with a random initial distribution of local clocks offset similar to the distribution shown in Figure 9. In general, the time to get the synchronization trends to grow with N, but it depends, also, on the initial relative offset between local clocks. The individual evolution of local clock values follows a similar pattern to those shown in Figure 4 (time varying consensus dynamics).

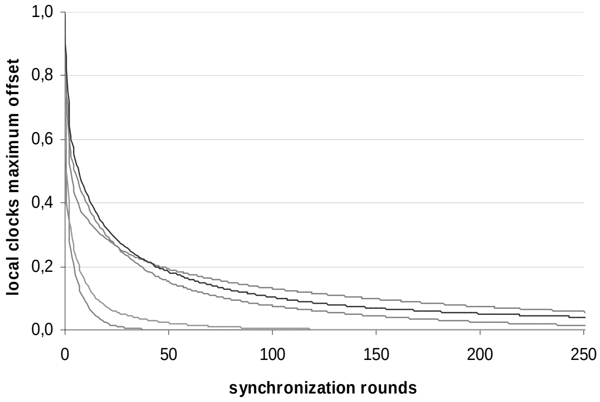


Figure 11. Evolution of maximum offset between local clocks, as a function of the synchronization rounds. Synchronization is driven by the time average value from adjacent nodes, x_{i-1}, x_i, and x_{i+1}, for ring networks with different N values, from N = 10 to N = 100. As N grows, synchronization trends to slow down.

The above synchronization algorithm does not consider a fundamental constraint in time synchronization: the local clocks should not run backwards [15]. To avoid this undesirable effect, if the time average value from $x_{i-1,}$ x_i, and x_{i+1} is lower than the local time x_i, it will not be considered. Then, the synchronization process is given by

$$x_{i,n+1} \leftarrow \max\left(\frac{x_{in} + x_{i-1,n} + x_{i+1,n}}{3}, x_{in}\right). \tag{46}$$

Figure 12 shows the performance of this synchronization algorithm variant. Its performance is similar to the previous algorithm, but the reduction rate of local clocks relative offset is worse.

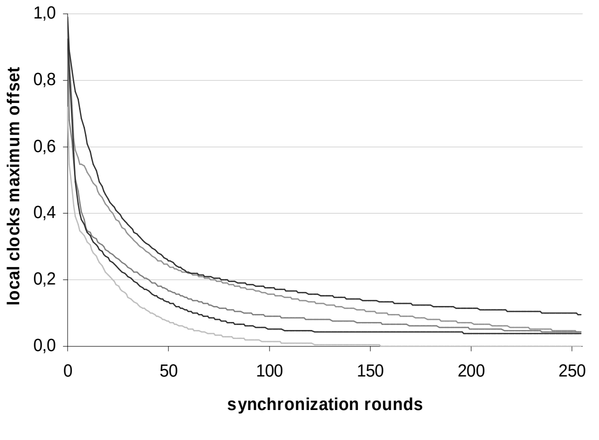


Figure 12. Evolution of maximum offset between local clocks, as a function of the synchronization rounds. Synchronization is driven by the time average value from adjacent nodes, x_{i-1}, x_i, and x_{i+1}, if the average value is greater than x_i, for ring networks with different N values, from N = 10 to N = 100. Although the graph is similar to the previous one, the performance of the algorithm is worse.

If the synchronization process was driven by the maximum time value of the neighbor local clocks, it seems plausible that the synchronization would be faster, since in all the synchronization rounds, the maximum value among $x_{i-1,}$ x_i, and x_{i+1} is expanded towards two nodes:

$$x_{i,n+1} \leftarrow \max\{x_{in}, x_{i-1,n}, x_{i+1,n}\}. \qquad (47)$$

Figure 13 shows the performance of this synchronization algorithm, for ring networks with different number of nodes, from N = 10 to 100, and initial local clocks offset randomly chosen. The synchronization process will be always finished after N/2 synchronization rounds, for N even, and after (N – 1)/2 rounds, for N odd. Figure 14 shows the same performance, for a ring network with N = 100 nodes and different initial local clock times. These results prove that this synchronization method is better than the previous diffusion algorithms, expressed in (41) and (46). The most advanced clock guides the whole synchronization process, which could be inappropriate for some applications. However, for the surveillance application purposes, it is preferable a fast synchronization.

B. Synchronization by a master clock reference

Another approach to the network synchronization can be driven by the periodic diffusion of a message containing a master clock reference, t_M, which is chosen as the local clock time of the sink node, n_1, in a similar way that the proposals included in [20][27][28]. It is assumed that the period of local clocks is larger than the message round trip time between adjacent nodes. The reference timestamp is sent sequentially from node 1 to node N:

$$n_1 \rightarrow n_2 \rightarrow \ldots \rightarrow n_N \rightarrow n_1.$$

Each node keeps a record of t_M, and also it relays the master time reference, t_M. It records the local timestamp, t_{Ii}, associated with the reception of t_M, as well. Once the message path is finished, n_1 sends a new message containing its local timestamp, t_E, associated with the reception of the initial message containing t_M. With these two values of time it is possible to estimate the global network round trip time between adjacent nodes, i.e., $(t_E – t_M)/N$.

Since each node knows its index i in the relay sequence, it will be able to adjust its local clock time, x_i, with the next expression:

$$x_i \leftarrow x_i - t_{Ii} + t_M + \frac{t_E - t_M}{N}(i-1). \qquad (48)$$

Here, $x_i – t_{Ii}$ reflects the difference between the current local time at node i and the reception time of t_M. The term t_M + $(t_E – t_M)(i – 1)N^{–1}$ contains a correction due to the reception delay of t_M at node i, which is proportional to the position of node in the relay sequence, i – 1. To explore the convergence of this synchronization algorithm, x_i and t_{Ii} will be expressed from the initial time value, x_{0i},

$$x_i \leftarrow x_{0i} + \frac{t_E - t_M}{N}(i-1+N) - x_{0i} - \frac{t_E - t_M}{N}(i-1) +$$

$$+ t_M + \frac{t_E - t_M}{N}(i-1), \qquad (49)$$

$$x_i \leftarrow t_M + \frac{t_E - t_M}{N}(N+i-1). \qquad (50)$$

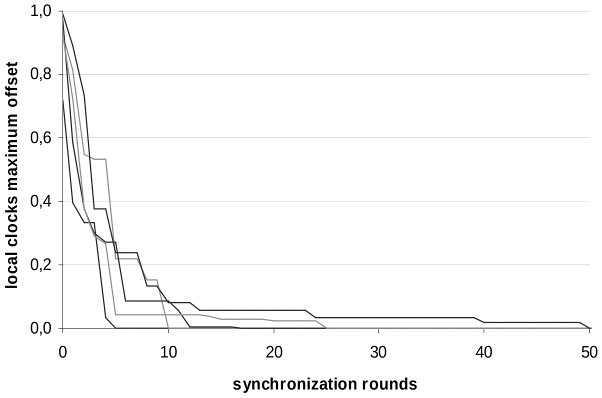


Figure 13. Maximum offset between local clocks. Synchronization driven by the time maximum value from adjacent nodes, $x_{i–1}$, x_i, and x_{i+1}, for ring networks with different N values, from N = 10 to N = 100. The synchronization is always reached after (N-1)/2 or N/2 synchronization rounds, for N odd or even, respectively.

Thus, the adjusted time x_i depends on t_M, t_E, and the position of node i in the network. Each node adjusts its local clock in a different instant of time, according to its position in the synchronization sequence.

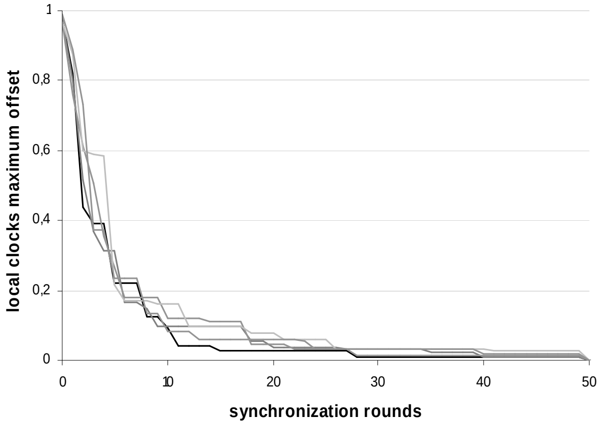


Figure 14. Performance of the synchronization protocol with the exchange of local timestamps from neighbor nodes and the selection of the maximum time value, for a WSN with ring topology with N = 100 nodes and different initial local clock times, which are randomly chosen with a maximum offset equal to 1 second. In all cases, synchonization time is reached in N /2 synchronization rounds.

Adding to (50) the correction of this difference, (t_E – t_M)(N – i)N^{-1}, all the x_i values are simultaneously equivalent:

$$x_i \leftarrow t_M + \frac{t_E - t_M}{N}(2N-1). \qquad (51)$$

The above process apparently reaches the synchronization in a fast way. However, as it was explained before, it is important to avoid adjustments of local clocks that could cause that local times could run backwards. So, (50) should be applied as

$$x_i \leftarrow \max\{x_i, t_M + \frac{t_E - t_M}{N}(N+i-1)\}. \qquad (52)$$

Figure 15 shows the local clocks offset with respect to the master clock after the application of (52). The resulting offset is caused by the above constraint: although maximum initial offset is reduced to the 50%, the synchronization has no effect on local clocks with an initial time advanced with respect to the master clock. Due to this constraint, more additional synchronization rounds cannot reduce this offset in a substantial way.

A further method could consist of the selection of the maximum value of local clocks. It can be initiated by the sink node; it transmits in sequence its local time value, and each node compare the received timestamp with its local time value, and it relays the maximum of these values. When the sequence is finished, the sink node receives the maximum value of local clocks, which will be chosen as the master clock reference. Then, the first synchronization method, expressed in (50), is applied. In that case, the synchronization time will be reached faster, as well, because all the nodes adjust their local time with the maximum value of local clocks.

Figure 16 allows to make a comparison of performances of some of the previous algorithms, in a WSN with ring topology with N = 100 nodes, after 100 synchronization rounds. The diffusion of a global time has a bad performance, due to the constraints discussed above. After these synchronization rounds, the reduction of the initial maximum offset is next to 50%.

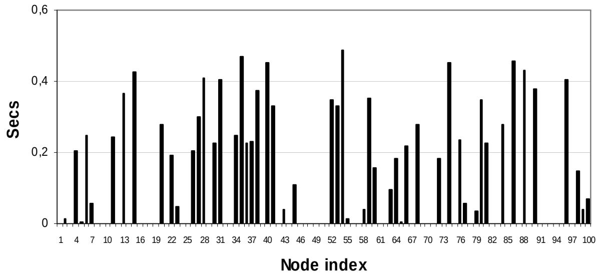


Figure 15. Synchronization driven by the transmission of a master clock reference. Local clock offset (in seconds) with respect to master clock time after one synchronization round. The reduction of the initial offset is approximately of 50%. The synchronization has no effect in local clocks advanced with respect to the master clock.

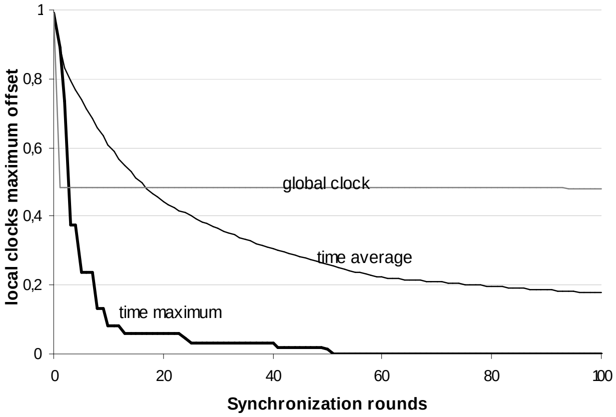


Figure 16. Performance of some synchronization algorithms described in Section VIII, in a WSN with ring topology with N= 100 nodes. The difussion of a global clock time across the network is the worst option, while the selection of the maximum time value from local clock time of neighbor nodes seems to be a more effective technique.

The selection of the average time value from neighbor nodes softly decreases the maximum offset between local clocks as the number of synchronization rounds grows; in that case, the reduction of the offset after the same number of synchronization rounds is about 80%. The best choice seems to be the selection of the maximum time value from neighbor nodes. The rough shape of the graphic line is due to the appearance of groups of local maximum times. These local values can remain unchanged during some synchronization rounds, until a greater time value gets its neighborhood.

IX. Conclusion and Future Work

This paper has applied some well established results from System Dynamics and Algebraic Graph Theory to state the ease of the synchronization of a WSN with regular ring topology, which has been initially designed to support surveillance applications. It is assumed that local clocks of network nodes exhibit a linear behavior during the whole synchronization process, since the effects of clock drift and skew have been supposed negligible, due to the limited accuracy of local clocks. Furhermore, a constraint to prevent that clocks could run backwards had to be applied.

The synchronizability of the WSN is enhanced by means of a reduction of the number of network nodes, and also through the decrease of another overall topological parameters, as the network diameter and the maximum network degree. Apparently, as it is established by the theory, the selected topology by needs of the application domain is not a good initial candidate for time synchronization. However, the application of various synchronization algorithms, initially based on a distributed diffusion of local clock values and subsequently on the diffusion of a master clock reference, shows significant performance differences. The exchange of local timestamps among adjacent nodes, and the subsequent selection of the maximum of these values to adjust local clocks, seems to be an effective technique, as well as the diffusion of a master clock reference, which is chosen as the maximum of local clock values.

As part of the future work, the above synchronization algorithms should be enriched with the addition of drift, skew and other nondeterministic effects. It is expected that the results obtained may guide the design of surveillance WSN and time synchronization protocols, and it is also foreseen to extract a set of more general rules to be extended to WSN with more complex topologies, and maybe designed for different purposes.

ACKNOWLEDGMENT

The research work described above has been partially funded by the European Commission, VI Framework Program, IST STREP Program, Action line IST-2005-2.5.3 (Embedded Systems), µSWN Project, code IST-034642. Also, the authors would like to thank Marta Zuazúa and Pedro Castillejo their initial contribution to the numerical simulations made.

REFERENCES

[1] José A. Sánchez Fernández, Ana B. García Hernando, José F. Martínez Ortega, and Lourdes López Santidrián, "Synchonization in a wireless sensor network designed for surveillance applications," Proc. Fifth International Conference in Wireless and Mobile Communications (ICWMC 2009), Cannes, France, Aug. 2009, IEEE Computer Society, pp. 369-372. doi: 10.1109/ICWMC.2009.68.

[2] Renato E. Mirollo and Stephen H. Strogatz, "Synchronization of pulse-coupled biological oscillators," SIAM J. Appl. Math., vol. 50, no. 6, Dec. 1990, pp.1645-1662.

[3] Duncan J. Watts and Steven H. Strogatz, "Collective dynamics of 'small-world' networks," Nature, vol. 393, June 1998, pp. 440-442. MacMillan Publishers Ltd., 1998.

[4] Yoshiki Kuramoto, Chemical Oscillations, Waves and Turbulences. Springer, 1984.

[5] Mauricio Barahona and L. M. Pecora, "Synchronization in small-world sytems," Phys. Rev. Letters, vol. 89, no. 5, July 2002, pp. 054101-1-054101-4.

[6] Louis M. Pecora, "Synchronization of oscillators in complex networks," Pramana – J. of Physics, vol. 70, no. 6, June 2008, pp. 1175-1198. Indian Academy of Sciences, 2008.

[7] Sergio Barbarossa and Francesco Celano, "Self-organizing sensor networks designed as a population of mutually coupled oscillators," Proc. IEEE Sixth Workshop on Signal Processing Advances in Wireless Communication (SPAWC 2005), New York, USA, June 2005, pp. 475-479.

[8] Sergio Barbarossa and Gesualdo Scutari, "Decentralized maximun-likelihood estimation for sensor networks composed of nonlinearly coupled dymanical systems," IEEE Transactions on Signal Processing, vol. 55, no. 7, July 2007, pp. 3456-3470.

[9] Sergio Barbarossa, Gesualdo Scutari, and Anantram Swami, "Achieving consensus in self-organizing wireless sensor networks: the impact of network topology in energy consumption," Proc. IEEE 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007), Honolulu, Hawaii, USA, April 2007, pp. II.841-II.844..

[10] Norman L. Biggs, Algebraic Graph Theory. Cambridge University Press, 1974.

[11] Allen J. Schwenk and Robin J. Wilson, "On the Eigenvalues of a Graph," in Selected Topics in Graph Theory, Lowell W. Beineke and Robin J. Wilson, Eds. Academic Press, 1978, pp. 307-336.

[12] Miroslav Fiedler, "Algebraic connectivity of graphs," Czechoslovak Mathematical Journal, vol. 23, no. 2, 1973, pp. 298-305. Institute of Mathematics, Acad. of Sc., Czech Republic.

[13] Bohan Mohar, "The Laplacian Spectrum of Graphs," in Graph Theory, Combinatorics, and Applications, vol. 2, Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, Eds. Wiley, 1991, pp. 871-898.

[14] Franesc Comellas and Silvia Gago, "Synchronizability of complex networks," Journal of Physics A: Mathematical and Theoretical, vol. 40, 2007, pp. 4483-4492. IOP Publishing, 2007.

[15] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," Ad-Hoc Networks, 3(3). May 2005, pp. 281-323.

[16] Ya R. Faizulkhakov, "Time synchronization methods for wireless sensor networks: a survey," Programming and Computer Software, vol. 33, no. 4, 2007, pp. 214-226. Pleiades Publishing, Ltd. ISSN. 0361-7688.

[17] Brian M. Sadler and Ananthram Swami, "Synchronization in sensor networks: an overview," milcom, pp. 1-6. MILCOM 2006. ISBN: 1-4244-0617-X.

[18] Fikret Sivrikaya and Büllent Yener, "Time synchronization in sensor networks: a survey," IEEE Network, July/Aug. 2004, pp. 45-50.

[19] Kay Römer, Philipp Blum, and Lenart Meier, "Time Synchronization and Calibration in Wireless Sensor Networks," in Handbook of Sensor Networks: Algorithms and Architectures, I. Stojmenovic, Ed. Wiley and Sons, Oct. 2005, pp. 199-237.

[20] Qun Li and Daniela Rus, "Global clock synchronization in sensor networks," IEEE Transactions on Computers, vol. 55, no. 2, Feb. 2006, pp. 214-226, doi:10.1109/TC.2006.25.

[21] Osvaldo Simeone and Umberto Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," EURASIP Journal on Wireless Communications and Networking, vol. 2007, pp. 1-13. Hindawi Publishing Corp. Article ID 57054, doi: 10.1155/2007/57004.

[22] Shyi-Long Lee, Yeung-Long Luo, Bruce E. Sagan, and Yeong-Nan Yeh, "Eigenvectors and eigenvalues of some special graphs. IV. Multilevel circulants," Int. J. of Quantum Chemistry, Vol. 41, 1992, pp. 105-116.

[23] Demetri P. Spanos, Reza Olfati-Saber, and Richard M. Murray, "Dynamic consensus on mobile networks," Proc. 16th International Federation of Automatic Control World Congress (IFAC 2006), Prague, Czech Republic, July 2005. Elsevier, 2006. ISBN: 978-0-08-0451084.

[24] Reza Olfati-Saber and Richard M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, vol. 49, no. 9, Sept. 2004, pp. 1520-1532.

[25] Juan A. Almendral and Albert Díaz-Guilera, "Dynamical and spectral properties of complex networks," New Journal of Physics, vol. 9, (2007) 187, doi: 10.1088/1367-2630/9/6/187.

[26] µSWN Project website: http://www.uswn.eu. Date of last access: July, 20th 2010.

[27] Jeremy Elson, Lewis Girod, and Deborah Estrin, "Fine-grained network time synchronization using reference broadcasts," Proc. Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, Massachussets. Dec. 2002, Vol 36, pp. 147-163.

[28] Michael Mock, Reiner Frings, Edgar Nett, and Spiro Trikaliotis, "Continuous clock synchronization in wireless real-time applications," Proc. 19th IEEE Symposium on Reliable Distributed Systems (SRDS-00), Nuremberg, Germany, Oct. 2000, pp. 125–133..

Network Forensics Models for Converged Architectures

Juan C. Pelaez
U.S. Army Research Laboratory
APG, MD 21005, USA
juan.c.pelaez@arl.army.mil,

Eduardo B. Fernandez
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33433, USA
ed@cse.fau.edu

Abstract— We discuss a systematic approach to network forensic collection and analysis of data in converged networks. Since attacks cannot be completely avoided, it is necessary to have appropriate forensics systems. Upon integration into a network forensic infrastructure, we expect this forensic model will enable a faster response and more structured investigations of Voice over IP (VoIP)-based network attacks.

Keywords—forensic patterns, network architecture, software architecture, Voice over IP.

I. INTRODUCTION

The generic solutions for problems that occur in similar ways in different contexts or environments can be expressed as patterns. A pattern is an encapsulated solution to a problem in a given context and can be used to guide the design or evaluation of systems [Gam94]. Analysis, design, and architectural patterns are well established and have proved their value in helping to produce good quality software. Recently, security patterns have joined this group and they are becoming accepted by industry [Sch06].

In a recent paper we introduced the concept of attack (i.e., misuse) patterns [Fer07a]. Attack patterns are a systematic description of the steps and objectives of an attack. This type of pattern describes, from the point of view of the attacker, how an attack is perpetrated and analyzes the ways of stopping the attack, including how to trace the steps of the attacker and what information (evidence) can be obtained at each phase. The pattern also attempts to correlate events with specific parts of the system.

In this paper we propose another type of pattern, the Forensic pattern [Pel09a]. It represents a systematic approach to network forensic collection and analysis of data. We introduce it in terms of VoIP networks. In conducting network forensics investigations in a VoIP environment, the collection of voice packets in real time and the use of automatic mechanisms are fundamental. We expect that forensic patterns will enable a faster response and more structured investigations of network attacks. Attacks on some VoIP applications such as VoIP in Tactical Internet require real-time evaluation and analysis, in contrast to the traditional method used in law enforcement, in which the victim's device is taken off-line after an attack has occurred. These patterns would also be useful for training apprentice forensics technicians about common investigative techniques and tools.

Figure 1 shows the relationships between our forensic patterns and existing security patterns. The patterns presented here are indicated with a double line and those under development with a dash line. The first set of Network Evidence forensic patterns provides abstract methods for collection and analysis of evidence; on the other hand, Tactical Evidence patterns are intended for military use (i.e., Tactical Internet). These forensic patterns will also be applicable to law enforcement and to some degree the relevant industry. The collection of all these patterns can be used to build a VoIP network forensic model.

The only other work we know about the use of patterns in forensics is [Dla09], although other works use UML models to describe forensic aspects, e.g., [Bog07].

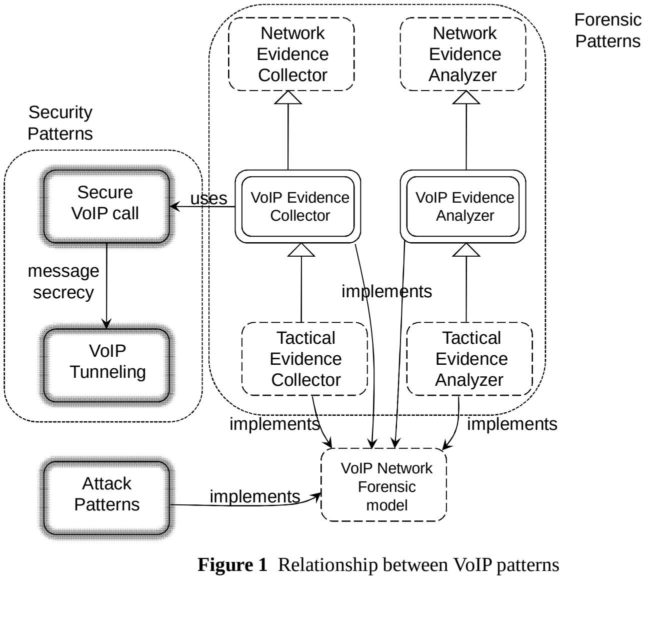
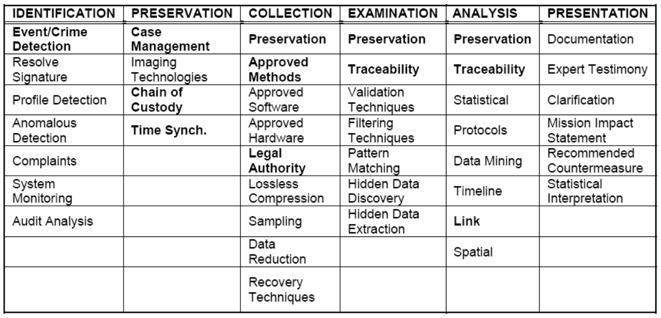


Figure 1 Relationship between VoIP patterns

The rest of the paper is structured as follows. In Section 2 we discuss a Reference Forensic model. In Section 3 we introduce the VoIP Evidence Collector pattern which collects attack packets on the basis of adaptively setting filtering rules of real-time collection. In Section 4 we show the VoIP Evidence Analyzer pattern which analyzes the collected forensics data, and presents a way to investigate and trace back attackers. Section 5 compares our approach to others, while Section 6 presents some conclusions.

II. Reference Forensic Model

Several models are used for investigation in forensic science. We chose the framework from The Digital Forensics Research Workshop (DFRWS) because it is a comprehensive approach and is more oriented to this paper's goals. The DFRWS model shows the sequential steps for digital forensic analysis [DFRWS01]. These steps are shown in Table 1.



IDENTIFICATION	PRESERVATION	COLLECTION	EXAMINATION	ANALYSIS	PRESENTATION
Event/Crime Detection	**Case Management**	**Preservation**	**Preservation**	**Preservation**	Documentation
Resolve Signature	Imaging Technologies	**Approved Methods**	**Traceability**	**Traceability**	Expert Testimony
Profile Detection	**Chain of Custody**	Approved Software	Validation Techniques	Statistical	Clarification
Anomalous Detection	**Time Synch.**	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement
Complaints		**Legal Authority**	Pattern Matching	Data Mining	Recommended Countermeasure
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation
Audit Analysis		Sampling	Hidden Data Extraction	**Link**	
		Data Reduction		Spatial	
		Recovery Techniques			

Table 1 - DFRWS Digital Investigative Framework [DFRWS01]

The initial phase or the identification of potential digital evidence (i.e., where might the evidence be found) is covered by Intrusion Detection Systems (IDS) and in some sense by attack patterns, which identify which units of the system have been used in the attack. The Preservation phase involves acquiring, seizing, and securing the digital evidence; making forensic images of the evidence; and establishing the chain of custody. We will concentrate on the middle phases of the forensic process (i.e., the collection, examination and analysis of the evidence) where the presented patterns will provide network investigators an structured method to collect more and better evidence and to reduce the analysis time in VoIP networks.

The presentation phase involves the legal aspects of the forensic investigation – presenting the findings in court and corporate investigative units by applying laws and policies to the expert testimony and securing the admissibility of the evidence and analysis. This phase is outside of the scope of this research, but it must be considered in order to create a comprehensive model.

III. VoIP Evidence Collector

The VoIP Evidence Collector pattern defines a structure and process to collect attack packets on the basis of adaptively setting filtering rules for real-time collection. The collected forensic data is sent to a network forensics analyzer for further analysis. This data is used to discover and reconstruct attacking behaviors.

Context

We are considering a VoIP environment, in which the monitored network should not be aware of the collection process. We assume that evidence is being preserved securely. We also assume a high-speed network with an authentication mechanism and secure transport channel between forensic components.

Problem

How to efficiently collect digital attack evidence in real-time from a variety of VoIP components and networks?

The solution to this problem is affected by the following *forces:*

- General security mechanisms, such as firewalls and Intrusion Detection Systems (IDS), cannot detect or prevent all attacks. They are unable to stop/detect unknown attacks, internal attacks, and attacks that come in the body of the messages (at a higher level). We need to analyze how an attack happened so we can try to stop it in the future, but we first need to collect the attack information.
- A real-time application, like VoIP, requires an automated collection of forensic data in order to provide data reduction and correlation. Current techniques dealing with evidence collection in converged networks are based on post-mortem (dead forensic) analysis. A potential source of valuable evidence (instant evidence) may be lost when using these types of forensics approaches.
- Even though there are a number of best practices in forensic science, there are no universal processes used to collect or analyze digital information. We need some systematic structure.
- The amount of effort required to collect information from different data sources is considerable. In a VoIP environment we need automated methods to filter huge volumes of collected data and extract and identify data of particular interest.
- The large amount of redundancy in raw alerts makes it difficult to analyze the underlying attacks efficiently [Wan05]
- A forensic investigator needs forensic methods with shorter response times because the large volume of irrelevant information and increasingly complex attack strategies make manual analysis impossible in a timely manner [Wan05].

Solution

Collect details about the attacker's activities against VoIP components (e.g., gatekeeper) and the voice packets on the VoIP network and send them to a forensic server. A forensic server is a mechanism that combines, analyzes, and stores the collected evidence data in its database for real-time response.

A common way of collecting data is to use sensors with examination capabilities for evidence collection. In VoIP forensic investigations, these devices will be deployed in the converged environment, thus reducing human intervention. These hardware devices are attached in front of the target servers (e.g., gatekeeper) or sensitive VoIP components, in order to capture all voice packets entering or leaving the system. These sensors are also used by the Intrusion Detection System (IDS) to monitor the VoIP network. Examiners can also use packet sniffers and Network Forensic Analysis Tools (NFAT) to capture and decode VoIP network traffic.

When the IDS detects any attempt to illegally use the gatekeeper or a known attack against VoIP components, it gives alarms to the forensic server, which in turn makes the evidence collector start collecting forensic data.

The evidence collector then collects and combines the forensic information from several information sources in the network under investigation. It will also filter out certain types of evidence to reduce redundancy.

Structure

Figure 2 shows the UML class diagram of the evidence collector (based on [Ren05]). The **Evidence Collector** is attached to hosts or network components (e.g., **gatekeeper**) at the node where we need to collect evidence in a **VoIP network**. Forensic data is collected using **embedded sensors** attached to key VoIP components or **NFAT** tools. VoIP components that are monitored can provide forensics information once an attack occurs. The Evidence Collector should be designed to extract forensic data and securely transport it (i.e., hash and encrypt) to the **forensic server** using a VoIP secure channel [Fer07b]. The forensic server combines the logs collected from the target servers and the VoIP network and stores them in its database to allow queries via command user interfaces. The network forensics server also controls the Evidence Collectors.

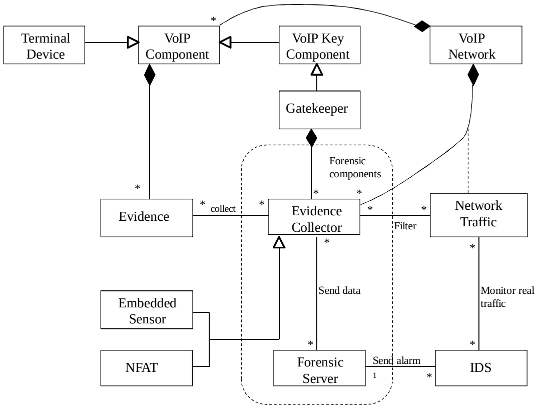


Figure 2 Evidence Collector Class Diagram

The **evidence** data collected from VoIP key components includes the IDS log files, system log files, and other forensic files. Other sensitive files may include the system configuration files and temp files. When attached to a terminal device, the Evidence Collector captures the **network traffic** to record the whole procedure of the intrusion and can be used to reconstruct the intrusion behavior [Ren05]. The evidence collector is also able to filter out certain types of evidence to reduce redundancy.

Implementation

After collecting the desired forensic data, the evidence collectors will send two types of data to the network forensics server, depending on the function performed. If the sensor is attached to a key VoIP component, it will collect logging system and audit data; otherwise (i.e., attached to a terminal device) it will act as packet sniffers do (with the Network Interface Card (NIC) set to promiscuous mode) or NFAT tools extracting raw network traffic data (e.g., entire frames, including the payloads, are captured with tcpdump). These data are used to discover and reconstruct attacking behaviors.

As mentioned before, after each attack against the VoIP network, the forensic data collected from key components and attacking sources may include logging data. The following data may also be useful to discriminate calls and call types:

- Terminal device information
  - Numbers called
  - Source and destination IP addresses
  - IP geographical localization
  - Incoming calls
  - Start/end times and duration
  - Voice mail access numbers
  - Call forwarding numbers
  - Incoming/outgoing messages
  - Access codes for voice mail systems
  - Contact lists

- VoIP data
  - Protocol type
  - Configuration data
  - Raw packets
  - Inter-arrival times
  - Variance of inter-arrival times
  - Payload size
  - Port numbers
  - Codecs

In order to maintain efficiency when capturing network traffic, we select the data to save, such as source and destination addresses and ports, and protocol type. The evidence collector can then extract all or selective voice packets (i.e., incoming or outgoing) over the VoIP network by applying a filter. The database on the forensics server will store the data sent by evidence collectors in order to perform the corresponding forensics analysis. We can use network segmentation techniques [Fer07b] to monitor the voice VLAN traffic independently from data VLAN traffic although the two share the same converged network.

Dynamics

The sequence diagram of Figure 3 shows the sequence of steps necessary to perform evidence collection in VoIP. In this scenario, as soon as an attack is detected against the gatekeeper by the IDS, the evidence collector starts capturing all activities of the possible attackers. The evidence collector will then send the collected data to the forensic server using a secure VoIP channel. Additionally, the collected forensic data is filtered and stored in the system database.

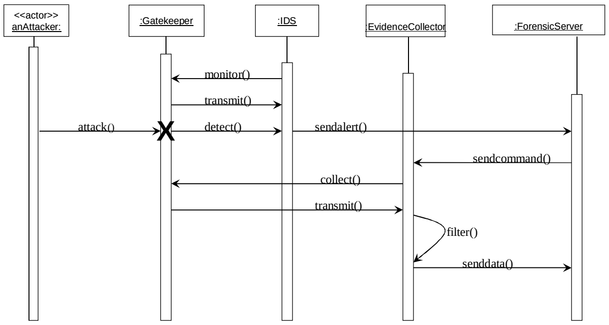


Figure 3 Sequence diagram for evidence collection in VoIP

Consequences

The *advantages* of this pattern include:

- The use of automated forensic tools as prescribed by this pattern allows effective real-time collection of forensic information which will reduce the investigation time in VoIP incidents.
- Significant logging information can be collected using this approach.
- The approach should be helpful to network investigators in identifying and understanding the mechanisms needed to collect real-time evidence in converged systems, because it provides a systematic way to collect the required information.
- The VoIP Evidence Collector pattern will also enable the rapid development and documentation of methods for preventing future attacks against VoIP networks.
- It is possible to investigate alleged voice calls using the evidence collector since voice travels in packets over the data network.
- For efficiency, the evidence collector can be set up for capturing selectively network packet streams over particular servers such as call, database, and web servers. The network forensics server can control the filter rules on the collector.
- On the other hand, based on the source/destination information, the evidence collector can filter the packets of a particular phone conversation.
- When encryption is present, the evidence collector can capture the headers and contents of packets separately.

The *disadvantages* of this approach are the limited scalability and relative inefficiency of the traffic's monitoring and recording. In large-volume traffic environments, there is a tradeoff between the monitored traffic and the available disk space [Ren05].

Known uses

The Solera Networks DS series [Sol09] is a commercial product line of network forensics appliances that capture, filter and store data in near real-time.

Related patterns

The VoIP Evidence Collector pattern has direct relationships to the VoIP Evidence Analyzer pattern, which will be presented next, and to the Secure VoIP Call pattern. Attack patterns could be used to select where to collect evidence.

IV. VoIP Evidence Analyzer

VoIP Evidence Analyzer pattern defines a structure and process to analyze the collected forensic data packets. It also presents a method of investigating an alleged IP attack scene and tracing back attackers.

Context

We are considering a VoIP environment in which the monitored network should not be aware of the collection process. We assume the existence of a mechanism to collect real-time evidence in converged systems and the preservation of such evidence in a secure way. We also assume a high-speed network with an authentication mechanism and secure transport channel among forensic components. We also assume that evidence has been collected by a VoIP Evidence Collector.

Problem

How to analyze evidence identified and extracted by the VoIP Evidence Collector in order to discover the attack source and other characteristics of the attack?

The solution is affected by the following *forces:*

- Two of the most costly, time-consuming and human-intensive tasks are the analysis and reconstruction of attacks in a compromised system.

- In order to correlate and interpret attacks against real-time converged networks, examiners need a structure for forensic analysis.
- An automated technique is fundamental to locate the attackers and reconstruct their criminal actions.
- We need shorter response times, a large volume of irrelevant information and increasingly complex attack strategies make manual analysis impossible in a timely manner [Wan05].
- Because the amount of data generated by VoIP networks is huge, storing network data for forensic analysis may be complicated.
- Encrypted packets are difficult to analyze.
- The forensic analysis process must guarantee data preservation and integrity.
- Attacks in converged networks are becoming more frequent and more complex to counter.
- A method is required for reusing network forensic knowledge and documenting forensic investigations.
- Forensic incidents in VoIP are often faced by examiners who do not have experience executing investigations or using similar forensic tools.

Solution

Combine (i.e., pre-process and store) all forensic logs and network traffic captured by the Evidence Collector into a forensic data repository (database and files) and analyze them using techniques such as log correlation and normalization [For04]. Logs are processed and converted into a simple format and then compared with the set of predefined attack patterns to identify possible security violations [Ren05]. The raw traffic data must also be converted into a readable format and stored in a separate database.

The evidence analyzer then performs automated inference based on the evidence database and presents results to the forensic investigator. The analysis process involves using automated methods to sift through large amounts of acquired data and extract and identify data of particular interest [Gra05].

Structure

Figure 4 shows a class diagram describing how an IP telephony and a forensic system integrate. This model shows the three primary forensic components: the **evidence collector**, the **forensic server** and the **network investigator**. The Evidence Collector is attached to a host that may be attacked in a VoIP network (e.g., **Gatekeeper**).

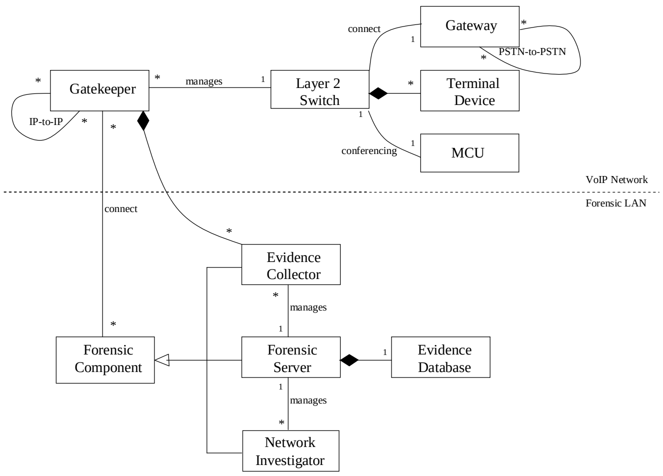


Figure 4 Class diagram for a VoIP network forensics system

The main function of the forensic server is combining the logging information collected from the VoIP network and its key components, and storing it in the evidence database. On the other hand, the network investigator acquires information about attackers and their sources by using techniques such as IP traceback and packet marking and by mapping topology to geographic locations so as to conduct further investigations.

Implementation

After the IDS gives the alert, the network forensics server sends a command to the network investigator (the response is in real-time). The network investigator receives information from the forensic server about sensitive spots on the VoIP network. Then the network investigator surveys the network in order to obtain useful information, such as the attacker location and phone numbers. The network investigator will also scan the network for mapping topology to find, for example, a false proxy server, or traceback the location of the attacker [Ren05]. Finally, the network investigator sends the scan and survey result to the forensic server using a VoIP secure channel [Fer07b]. This result will include such information as the topology of the network, the IP address, the MAC address, the possible geographic location of the IP.

The network forensics server can also analyze the attack behavior by replaying the attacking procedures. Network forensics tools can reorganize the packets into individual transport-layer connections between machines [Ren05]. With the appropriate tools, investigators can capture the packets and decode their voice packet payloads in order to analyze VoIP calls.

The forensics server provides correlations of forensics data in order to discover the attack behavior. This process will provide network investigators a better way to monitor voice traffic data and correlate events from VoIP security mechanisms (e.g., IDS).

To construct the given same events, it is necessary to correlate the different format logs to a single-layer data format by time, IP, and User ID. This task is known as normalization [For04]. Correlation in forensics is based on the knowledge of previous attacks gained by historical methods, geographical location, strength of signal, and the behavior of the attacker. Likewise, Attack Patterns [Fer07a] will provide prior knowledge of known exploits. VoIP Correlation Rules correlate events taken from multiple VoIP source devices, including Call Managers, IP PBXs, and voice gateways [Hic07]. These correlation rules will detect, for example, theft of service attempts as well as DoS attacks against VoIP servers.

Dynamics

The sequence diagram of Figure 5 shows the sequence of steps necessary to perform evidence analysis in VoIP. In the initial phase, the forensic evidence sent by the evidence collector is preprocessed and stored in the forensic server database. After scanning and surveying the network, the network investigator sends the results to the forensic server for further analysis and replay of the attacking procedures.

Consequences

The advantages of this pattern include:

- Investigators will be able to perform network forensic investigations (in real-time) in converged networks in a structured way.
- Designers will be able to correct weak points in a VoIP network perimeter in order to prevent future similar attacks.
- Automated evidence analysis will produce an immediate impact on the forensic investigator’s ability to reduce response times [Wan05].
- The information that is collected can be used to anticipate adversarial actions, understand the current state of affairs, and help in determining appropriate courses of action [Gio02].
- The Evidence Analyzer can provide information about logs and for tracing back attackers.
- All the data from the monitored host, NFAT, and the network investigator will be stored as the evidence and analyzed for the final presentation.
- Encrypted data can be examined using traffic analysis. By examining the flow of packets over time, it is possible to determine such matters as when a user is using the VoIP device, whom the user communicates with, and the call history.

Possible disadvantages include:

- Disk storage space time overhead requirements may be a concern in some environments.
- Attack patterns need to be continually updated, and this will normally require human expertise.

Known uses

QRadar is a commercial product designed by Q1 labs to offer security monitoring for Voice over IP (VoIP) networks. This module combines network behavior analysis and security event correlation for monitoring across the network protocol, application, and security services layers of a VoIP network [Hic07].

Related patterns

The VoIP Evidence Analyzer pattern has direct relationships to the VoIP Evidence Collector pattern that was previously introduced and the Secure VoIP Call pattern. As indicated, attack patterns could help in forensic evidence analysis.

V RELATED WORK

The patterns have been inspired by ideas of Ren and Jin [Ren05], who developed a model based on distributed adaptive network forensics and active real time network investigation. Likewise, Tang [Tan05] developed a network forensics framework based on distributed techniques, which provides an integrated platform for automatic forensic evidence collection and data storage, supporting the integration of known attribution methods, and an attack attribution graph generation mechanism to illustrate hacking procedures. Finally, Wang and Daniels [Wan05] propose an evidence graph model to facilitate the presentation and manipulation of intrusion evidence. For automated evidence analysis, they developed a hierarchical reasoning framework that included local reasoning and global reasoning.

Kahvedzic and Kechadi prented a framework using ontologies for modeling , analyzing, and reusing forensic knowledge [Kah09]. However, their objective is to systematize and clarify the vocabulary used in describing forensic investigations.

Dlamini , Olivier, and Sibiya applied design patterns to add flexibility and reusability to traffic isolation so that forensic analysis can be performed more conveniently [Dla09}. We emphasize the collection and analysis of forensic information in a systematic way.

VI CONCLUSION AND FUTURE WORK

We have introduced the concept of forensic patterns as they relate to VoIP investigations. We illustrated these ideas using UML object oriented models. Likewise, some issues involved in VoIP forensic investigations were studied. Since attacks cannot be completely avoided, it is necessary to have appropriate forensics systems. By using these forensic patterns, investigators will have a structured method to collect, search and analyze network forensic data.

The proposed VoIP Evidence Collector pattern could use NFATs in combination with hardware sensors for real-time collection. Likewise, the VoIP Evidence Analyzer pattern analyzes the collected forensic data packets, and presents a process of investigating attacks against the VoIP network.

The usefulness of VoIP forensic patterns will depend on the creation and implementation of a VoIP pattern system [Pel09b]. These are the first steps toward a methodology for modeling network forensics. Future work will include the development of more general forensic patterns (i.e., not just for VoIP), as well as the corresponding wireless forensic patterns for a Tactical Internet environment (i.e., the integration of tactical digital radios and commercial Internet technology). In addition, we will develop a UML network forensic model based on this pattern system as a reference architecture for forensics. Other possibilities include combining our patterns with the design patterns of [Dla09] to improve their implementation.

References

[Bog07] A.C. Bogen D. A. Dampier, and J.C. Carver, "Support for computer forensics examination planning with domain modeling: A report of one experiment trial", *Procs. of the 40th Annual Hawaii Int. Conf. on System Sciences (HICSS* 2007).

[Dla09] I. Dlamini, M. Olivier, and S. Sibiya, "Pattern-based approach for logical traffic isolation forensic modeling", *Procs. of the Third Int. Workshop on Secure System Mehologies Using Patterns (SPattern 2009),* IEEE, Sept. 2009, 145-149.

[DFRWS01] Digital Forensics Research Workshop. "A Road Map for Digital Forensics Research 2001." *Digital Forensics Research Workshop 6 November* (2001): http://www.dfrws.org /2001/dfrws-rm-final.pdf *(last accessed 8 June 2010).*

[Fer07a] E. B. Fernandez, J. C. Pelaez, and M. M. Larrondo-Petrie. "Attack patterns: A new forensic and design tool." *Proceedings of the Third Annual IFIP WG 11.9 International*. Conference on Digital Forensics, Orlando, FL, Jan. 29-31, 2007.

[Fer07b] E.B. Fernandez, J.C. Pelaez, and M.M. Larrondo-Petrie, "Security patterns for voice over IP networks", *Proceedings. of the 2nd IEEE Int. Multiconference on Computing in the Global Information Technology* (ICCGI 2007), March 4-9, Guadeloupe, French Caribbean.

[For04] D. Valentino Forte, The Art of Log Correlation - Tools and Techniques for Correlating Events and Log Files, IR Italy Project, 2004.

[Gam94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Boston, Mass., 1994.

[Gra05] T. Grance and S. Chevalier. "Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response." *Recommendations of the National Institute of Standards and Technology*. August, 2005.

[Hic07] A. Hickey. "VoIP security monitoring gets proactive." SearchVoIP.com, 25 Jan 2007 *(last accessed 8 June 2010). http://searchunifiedcommunications.techtarget.com/news/article/0,289142,sid186_gci1240544,00.html*

[Kah09] D. Kahvedzic and T. Kechadi, "DIALOG: A framework for modeling, analysis and reuse of digital forensic knowledge*", Procs. of the 2009 Digital Forensics Research Workshop (DFRWS'09) , (last accessed 8 June 2010)* http://www.dfrws.org/2009/proceedings/p23-kahvedzic.pdf

[Pel09a] J.C. Pelaez and E.B. Fernandez. "VoIP Network Forensic Patterns." *Proceedings of the Fourth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2009).* Cannes, France, August 23-29, 2009.

[Pel09b] J.C. Pelaez, E.B. Fernandez, and M.M. Larrondo-Petrie, "Misuse patterns in VoIP", accepted for Wiley's Security and Communication Networks Journal.

[Ren05] W. Ren, H. Jin. "Distributed Agent-based Real Time Network Intrusion Forensics System Architecture Design." *Proceedings of the 19th International Conference on Advanced Information Networking and Applications* (AINA'05). March, 2005.

[Sch06] M. Schumacher, E.B. Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating Security and Systems Engineering, Wiley publishing, New York, 2006.

[Sol09] Solera Networks. "DS Series Network Forensics Appliances."http://www.soleranetworks.com/products/forensics-appliances.php *(last accessed 8 June 2010).*

[Tan05] Y. Tang and T. E. Daniels, "A Simple Framework for Distributed Forensics," icdcsw, vol. 2, pp.163-169, Second International Workshop on Security in Distributed Computing Systems (SDCS) (ICDCSW'05), 2005.

[Wan05] W. Wang and T. Daniels. "Building Evidence Graphs for Network Forensics Analysis." *Proceedings of the 21st Annual Computer Security Applications Conference* (ACSAC 2005). September 2005.

Self-Healing and Secure Adaptive Messaging Middleware for Business-Critical Systems

Habtamu Abie
Norwegian Computing Center,
Oslo, Norway
e-mail: Habtamu.Abie@nr.no

Reijo M. Savola
VTT Technical Research Centre of Finland,
Oulu, Finland
e-mail: Reijo.Savola@vtt.fi

John Bigham
Queen Mary, University of London, London, UK
e-mail: john.bigham@elec.qmul.ac.uk

Ilesh Dattani
Q-Sphere Ltd, London, UK
e-mail: ilesh@q-sphere.com

Domenico Rotondi
TXT e-solutions SpA, Valenzano (BA), Italy
e-mail: Domenico.Rotondi@TXTGroup.Com

Giorgio Da Bormida
CNIT, Florence, Italy
e-mail: dabormida@gmail.com

Abstract **— Current business-critical systems have stringent requirements for the significant and measurable increase in the end-to-end intelligence, security, scalability, self-adaptation and resilience. Existing state-of-the-art messaging systems achieve arbitrary resilience by a brute-force approach. Self-healing is either rudimentary or non-existent. In this study we present a self-healing and secure adaptive messaging middleware that provides solutions to overcome limitations in robustness, resilience, self-adaptability, scalability, and assurance against security threats and erroneous input during run-time in the face of changing threats. This developed system supports a messaging infrastructure which enables adaptive functions and assurance against security vulnerabilities and erroneous input vulnerabilities to improve the reliability, robustness and dependability of business-critical infrastructures. It provides autonomous adjustments of the run-time configuration of the system in order to preserve and maintain optimal and uninterrupted operation, improvement of the strength of security and degree of trust in the system, and improvement of the assessability and verifiability of the trustworthiness of the system. The methodology used in this research is partly analytical and partly experimental. We develop the new core functionalities theoretically and validate them practically by prototyping.**

Keywords - Self-Adaptation, Messaging Middleware, Self-healing, Resilience, Self-protection, Adaptive Security, Security Metrics

I. Introduction

The environment surrounding modern business-critical systems is in a continuous state of change throughout the lifetime of an application. With the increase in the dependence of businesses on messaging middleware systems, the need for dependable, trustable, robust and secure adaptive messaging systems becomes ever more acute.

The primary contribution of this work is the analysis and synthesis of a self-healing and secure adaptive messaging middleware for business-critical systems introduced in our earlier work [1]. This paper analyzes (i) the autonomous adjustments of the run-time configuration of the system the purpose of which is the preservation and maintenance of optimal and uninterrupted operation, (ii) the improvement of the strength of security and degree of trust in the system, (iii) the improvement of the assessability and verifiability of the trustworthiness of the system, and (iv) the adaptive integration of the GEMOM solution that consists of a continuous cycle of monitoring, measurement, assessment, optimization, self-healing, adaptation and evolution to meet the challenges in the changing environments.

Message-Oriented Middleware (MOM) provides the functionality of interoperability, portability, and flexibility of architectures that enables applications to exchange messages with other applications without having to know what platform the other application resides on [2][3][4]. MOMs provide a service that allows content providers and consumers to concentrate on the production and consumption of transmitted information. In essence, MOM is compatible with, and can be viewed as, a central component of the Enterprise Service Bus (ESB) [5] architecture, where the MOM message broker acts as the ‘bus’ between applications. The term Publish/Subscribe (PS) MOM does not necessarily imply the broad collection of concepts and standards of ESB. The key advantage of the MOM architecture is that it reduces the number of point-to-point connections in a complex business-critical IT system.

However, existing commercial MOM technologies are expensive and lack scalability. In addition, there are no solutions that provide the required levels of robustness, reliability and resilience appropriate for future real-time and business-critical systems. Moreover, because they must be self-organizing, modern autonomous MOM platforms have stringent requirements for resilience (ability to keep going in given scenarios by learning, evolving, etc., over time), self-healing (ability of the system to preserve its capabilities even in the event of failure of any individual or multiple components), self-learning and self-optimization, self-adaptation and evolution, fault-tolerance, self-active-vulnerability assessment, adaptive autonomic security. GEMOM (Genetic Message-Oriented Secure Middleware) provides solutions to overcome these limitations to secure messaging to support a communications framework that can be deployed for a wide range of applications [6][7].

Complex, distributed business-critical systems are virtually impossible to implement without the heavy use of a messaging infrastructure. The most common variant of these systems is the scheme utilizing the PS messaging paradigm. Synchronous Request/Reply is easily overlaid on top of PS, making PS the right proxy for overall messaging. GEMOM uses the PS messaging paradigm and further supports better interoperability and integration of business-critical systems by allowing actual instances to be configured so various functions are subcontracted to one or more separate, external or federated entities. GEMOM [7][1] has made advances in the following areas: resilience and self-healing, scalability and resilience, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement.

The rest of this paper is organized as follows. Section II gives an overview of related work. Section III presents the GEMOM system architecture with a brief overview of key properties of the system. In Section IV, we describe the self-optimization, self-healing and scalability of GEMOM. The holistic and systematic adaptive security approach is presented in Section V. In Section VI, we describe the enhanced interoperability and integration of distributed business-critical systems. Section VII shows how the self-healing, adaptive security and the different tool-sets are integrated. A brief introduction to GEMOM prototypes and validation is given in Section VIII. The paper closes with a conclusion and future work in Section IX.

II. RELATED WORK

This section gives a rundown of related work and comparisons of our work with that of whose work is most closely related to ours.

A. MOM Systems

MOM platforms are available in a wide range of implementations such as JMS, WebSphereMQ, TIBCO, Herald, Hermes, SIENA, Gryphon, JEDI and REBECCA where each of these MOMs has been designed to achieve specific goals, and employs unique functionality to meet specific messaging challenges [8]. However, the current state-of-the-art technologies do not allow security mechanisms to actually predict or anticipate future threats, and to adapt to rapidly changing behaviours and threats over time. Table I describes the key functionalities of MOMs, the limitations of existing MOM Systems, and the GEMOM advances as comparison as shown in the table below.

TABLE I. CONTRASTING MOM FUNCTIONALITIES

Key functionalities	Limitations of existing systems	GEMOM advances
Performance - throughput & latency. (*Throughput* represents the number of requests served by the MOM per second). *Latency* is the time between publishing a message and the subscriber receiving it	Insufficient information and control over performance	Externalised architecture for monitoring resilience, which limits impedance of processing rates while offering control
Increasing interoperability, portability, and flexibility of architectures	Data-loss prone - no means to compensate for the reliability loss and to integrate limitation of risk of loss into the system offered to the user	Compensation for the reliability loss by automatically finding another source of redundancy
Publish/subscribe – asynchronous	No system re-factoring at runtime	Hot standby brokers with instant switch-over and no data loss
Resilience, self-healing and scalability	Prone to feed failures, arbitrary resilience by a brute-force approach, self-healing is either rudimentary or non-existent, and risk is not quantified	Integration of the tool-sets for the management of threats and of vulnerabilities, intelligent techniques to support security assurance, clustering of namespaces/ topics into namespace with namespace replication, and quantification of risk
Security management	No holistic or systematic adaptive security approach	Adaptive security management based on security evidence information offered by security metrics

B. Self-Healing and Self-Adaptation

Self-healing systems attempt to 'heal' themselves in the sense that they recover from faults and regain normative performance levels by employing models, whether external or internal, to monitor system behaviour and by using inputs to adapt themselves to the run-time environment [9]. Self-adaptive systems aim at anticipating changes which occur in a complex environment and automatically dealing with them at run-time, on the basis of the knowledge of what is happening in the system, guided by objectives and needs of stakeholders [10]. Self-adaptive software evaluates its own behaviour and changes it when the evaluation indicates that the software is not accomplishing what it is intended to do, or when this will lead to better functionality or performance [11]. Self-adaptive systems are characterized by three core functionalities: monitoring (sensing) the environment to recognize problems, making decisions on which behaviour to exhibit, and realizing the behaviour change by adaptation [10][11][12].

A number of surveys of mechanisms and techniques to achieve self-healing and self-adaptation exist. Kramer [13] gives a survey of self-adaptive parameter control in evolutionary computation, classifies self-adaptation in the taxonomy of parameter-setting techniques, gives an overview of automatic online-controllable evolutionary operators, and provides a coherent view of search techniques in the space of strategy parameters, and concludes that self-adaptation is an efficient way to control the strategic parameters of an evolutionary optimization algorithm automatically during optimization. In [14], a classification of adaptation on the basis of the mechanisms used and the level at which adaptation operates within the evolutionary algorithm has been developed. Their classification covers all forms of adaptation in evolutionary computation. Ghosh, Sharman, Rao, and Upadhyaya present a survey and synthesis of self-healing systems and propose a strategy of synthesis and classification [9]. Miorandi, Yamamoto, and Pellegrini present a survey of evolutionary and embryogenic approaches to autonomic networking, applicable to network-level functionalities [15]. They give an overview of the major technical challenges to be met by anyone applying the surveyed techniques to autonomic systems.

A number of self-healing and self-adaptive systems have been recently developed supporting healing and adaptation at different levels. Rodero-Merino, Fernandez, Lopez, and Cholvi propose a topology self-adaptation mechanism for efficient resource location that makes the network change its topology to maintain an efficient configuration that depends on the system load and the peer's capacities [16]. Dustdar, Goeschka, Truong, and Zdun have also proposed self-adaptation techniques for complex service-oriented systems, which comprise model-driven compliance support, runtime interaction mining, run-time management of requirements, and explicit control-loop architecture [17]. Alencar and Weigand [18] present the challenges involved in the predictive self-adaptation of service bundles in a service-oriented scenario in terms of time and cost involved in the adaptation that can be useful to enhance the decision-making process in a business strategic or tactical context. Gjørven, Rouvoy, and Eliassen describe a technology-agnostic self-adaptation middleware for service-oriented architectures that can support a cross-layer adaptation of SOA systems and they show that their middleware is able to exploit both the technologies of the service interface and application layers to support a coordinated adaptation of both layers [19]. Reinecke, Wolter, and Moorsel [20] propose a framework and methodology for the definition of benefit-based adaptivity metrics that allow an informed choice between systems based on their adaptivity to be made, and provide a broad survey of related approaches that may be used in the study of adaptivity and to evaluate their respective merits in relation to the proposed adaptivity metric. Giannakopoulos and Palpanas [21] propose an adaptive subscription service architecture, concerning the update of the clients of an entity name system with information on entity changes, using information from user feedback to model user needs, taking into account both the type and the content of changes.

Our self-healing and secure adaptive messaging middleware is inspired by the work above but is focussed more on providing resilience, self-healing, scalability, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement.

C. Adaptive Security

There have been a number of adaptive security systems that have been developed recently supporting adaptation at different levels and for a number of reasons. Chess, Palmer, and White outline a number of security and privacy challenges facing those designing and developing autonomic systems, and also a number of ways that autonomic principles can be used to make systems more secure than they are today [12]. Hager [22] has in his dissertation developed a context-aware and adaptive security for wireless networks, with application to a pervasive networking environment. Shnitko describes an approach to the design of complex secure systems based on the formalization of adaptive functions in an information-security context, and both practical and theoretical aspects related to the usage of adaptive security in complex systems [23]. Son, Zimmerman, and Hansson propose an adaptable security manager for real-time transactions featuring adaptability and multi-level security services that can be applied in a soft real-time environment in order to achieve performance gains [24]. Schneck and Schwan [25] present an adaptive authentication for networked applications with a novel security control abstraction with which trade-offs in security versus performance may be made explicit.

Zou, Lu, and Jin [26] present an architecture and fuzzy adaptive security algorithm in an intelligent firewall where a fuzzy controller is the core module and the characteristics of packets are fuzzified as its inputs. Abie, Spilling, and Foyn [27] and Abie [28] propose self-contained objects for secure information-distribution systems that carry with them usage rights and enforce on their own behalf these rights assigned to them, preserving their confidentiality and integrity. Pietzowski, Satzger, Trumler, and Ungerer propose a bio-inspired self-protecting organic message-oriented middleware with artificial antibodies that evaluates optimal parameter-setting techniques to minimize the memory space needed for storing the antibodies and to reduce the time needed for detecting malicious messages [29]. Djordjevic, Nair, and Dimitrakos present a virtualized trusted computing platform for adaptive security enforcement of web-services interactions by providing virtual machine-level separation that maps from logical domains imposed by web-service-level enforcement policies [30]. Luo, Ni, and Yong [31] and Ma, Abie, Skramstad, and Nygaard [32] propose trustworthiness assessment methods for the calculation of the degree of trust in a grid computing environment and digital records management over time, respectively. Boukerche and Ren present a trust-based security system for ubiquitous and pervasive computing environments, a trust model that assigns credentials to nodes, updates private keys, manages the trust value of each node, and makes appropriate decisions about nodes' access rights [33]. Goovaerts, Win, and Joosen present a bus-based architecture for integrating security middleware services for achieving flexible and adaptive security middleware with a qualitative comparison of the flexibility of the approach with an alternative aspect-oriented-middleware-based approach [34].

A survey of approaches to adaptive application security, and adaptive middleware can be found in [35] and [36], respectively. A taxonomy of compositional adaptation and a comparison of two approaches for achieving flexible and adaptive security middleware can also be found in [37] and [34], respectively. Presentations of semantic and logical foundations of an adaptive security infrastructure can be found in [38].

It was the work of, inter alia, the above researchers that convinced us of the viability of adaptive security, and therefore gave us confidence in the productivity of our research in this direction. Table II gives a brief comparison of our adaptive security work with other closely related work with their special features and benefits categorized according to their types of adaptation.

TABLE II. BRIEF SURVEY OF ADAPTIVE SECURITY AND TRUST

Adaptation type	References	Features and benefits	Limitations	Advances in our approach
Risk	McGraw [39]	Risk-adaptable access control that bases its access decision on a computation of security risk and operational need	Lack of assessability and verifiability of the trustworthiness of the system	Combination of trust-based security and security-based trust. The integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating.
	Qu and Hariri [40]	Anomaly-based self-protection against network attacks	Lack of models for trust and policy adaptation	Combination of trust-based security and security-based trust
Trust	Ryutov, Zhou, Neuman, Leithead, and Seamons [41]	Adaptive trust negotiation and access control for flexible policy adaptation and capturing dynamically changing system security requirements using user and system suspicion levels.	Lack of integration of a continuous cycle of monitoring, assessment and evaluation models.	Integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating,
	Shrobe, Doyle, and Szolovits [42]	An active trust management for autonomous adaptive survivable systems for compromise-based trust management model	Lack of close integration with adaptive security to minimizing the rate and severity of compromises	Combines risk-based security and a security-based trust model using an adaptive control loop for the provision of a secure communication environment.
Security	Djordjevic, Nair, and Dimitrakos [30]	Trusted computing for security enforcement of web services	Lack of models for integration of assessability and verifiability models	Integration of a continuous cycle of monitoring, assessment and evaluation, and tools and processes for pre-emptive vulnerability testing and updating,
	Weise [43]	A security architecture and adaptive security which is capable of reducing threats and anticipating threats before they are manifested, and uses biological and eco-system metaphors	Lack of models for trust building and for integration of assessability and verifiability models	Combines a compromise-based trust model to maximize the value of risk-taking, and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.
Policy	Venkatesan and Bhattacharya [44]	Threat-adaptive security policy that adapts security policies according to threats	Lack of trust model to maximize the value of risk-taking, and integration of assessability and verifiability models	Combines a compromise-based trust model to maximizing the value of risk-taking and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.
	Lamanna [45]	Adaptive security policies enforced by software dynamic translation	Lack of trust model to maximize the value of risk-taking and to minimize the rate and severity of compromises	Combines a compromise-based trust model to maximize the value of risk-taking and integrates a continuous cycle of monitoring, assessment and evolution, and tools and processes for pre-emptive vulnerability testing and updating.

D. Adaptive Security Metrics

Security metrics provide the on-line means with which to score different security solutions in adaptive security management. In addition, metrics can be used off-line for security engineering decision-making during the whole lifecycle of the system.

The security metrics development approaches that are most valuable in adaptive security management, focus on security-enforcing mechanisms and the quality of the overall security of the system, are briefly discussed here. Wang and Wulf describe their general-level security metrics development framework based on a decomposition approach in [46]. Heyman, Scandariato, Huygens, and Joosen [47] use a security objective decomposition approach and associate the metrics with security patterns. Savola and Abie apply Wang and Wulf's approach to security requirements and enhance it by a complete description of their entire methodology from the analysis of threats and vulnerabilities to a balanced and detailed collection of metrics, and present an initial collection of security metrics for GEMOM in [48]. The Common Vulnerability Scoring System (CVSS) [49] is an initiative aiming at providing an open and standardized method for rating vulnerabilities. The CVSS, along with some other security vulnerability and weakness metrics systems, has been integrated by the U.S. National Institute of Standards and Technology (NIST) into Security Content Automation Protocol (SCAP) [50]. The purpose of this effort is to develop solutions that will be widely-accepted, but it is not complete; it lacks means to obtain evidence of the security level of security-enforcing mechanisms and methodologies to relate the metrics to security objectives. Howard, Pincus, and Wing [51] and Manadhata, Kaynar, and Wing [52] propose an abstract *attack surface* measurement method. Attack surface means the parts that can be accessed by unauthenticated users, such as attackers, including the set of entry points, exit points, the set of channels and the set of non-trusted data items. Further surveys of security metrics can be found in [53][54][55][56].

III. GEMOM System Architecture and Key Properties

GEMOM exploits the predominant PS [2][57] variant of MOM. For completeness, it provides a synchronous Request/Reply overlay as well. In GEMOM, publishers of messages do not send their messages directly to specific receivers. The published messages are positioned in a hierarchy of logical channels (called namespaces and topics in GEMOM) without the publishers having explicit knowledge of what subscribers there may be. Publishers are loosely coupled to subscribers and need not even know of their existence. Namespaces are a hierarchical classification of topics.

The GEMOM system achieves a considerable increase in the end-to-end resilience of complex distributed business-critical systems to ensure secure transmission of data and services across heterogeneous infrastructures and networks. The GEMOM platform consists of the following resilience and self-adaptive properties: (i) reliability of message sourcing and delivery, (ii) scalability in messaging, (iii) replication of structural and dynamic properties of security policies with adaptive authentication and authorization model, (iv) process-zoning and overall encapsulation to an arbitrary level, and (v) new techniques and tools for pre-emptive and automated checking a deployed system for robustness and vulnerabilities to faults, oversights and attacks, all of which are described in detail in the ensuing sections. In the following subsections, we briefly present the system architecture and key properties.

A. GEMOM System Architecture

The GEMOM [7] system architecture is composed of a set of communicating nodes, G-Nodes. Some of these G-Nodes are operational (micro) nodes and some managerial (macro) nodes, see Figure 1. The operational G-Nodes can be classified as Message Brokers (Bs), Clients (either publishing or subscribing messages, Publishers (Ps) or Subscribers (Ss)), Authentication and Authorization Modules (AAMs), Anomaly Detector (AD), Security Measurement Module (SMM), etc. They communicate with managerial nodes of different types. The managerial G-Nodes can be classified as Adaptive Security Managers (ASMs), Audit and Logging Modules (ALMs), and Security Monitoring Tools (SMTs) with associated Security Monitors (SMs) and Quality of Service (QoS) Monitors (QMs), Resilience Managers (RMs), Security Anomaly Managers (SAMs), etc. The managerial G-Nodes make decisions about the run-time operation of the system and require a wider perspective than the individual operational G-Nodes. In GEMOM, a Message Broker is a package consisting of an application server, numerous plug-and-play objects, configuration files, and database schemas.

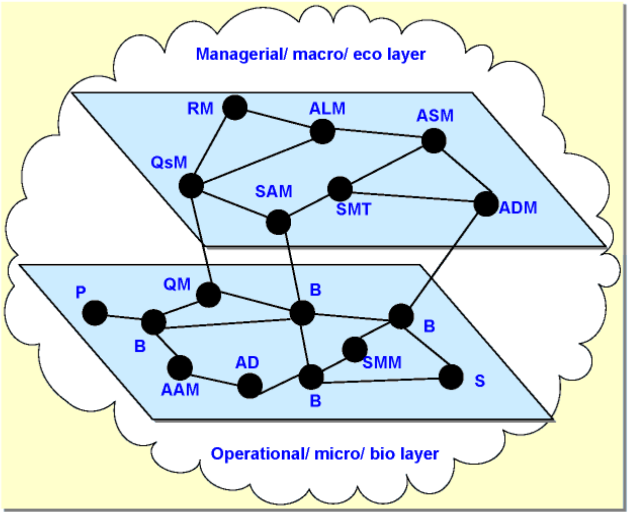


Figure 1. GEMOM system architecture [1]

Figure 1 depicts the GEMOM system architecture showing the main components. It supports mechanisms for adding G-Nodes, for measuring security and QoS between overlay components and publishers and subscribers and deciding what action is to be taken to mitigate loss of security or QoS, or breakdowns, for discovering and communicating with other components in the overlay network, for evaluating the performance of the system in the context of the monitored performance, for establishing the state of the overlay network, and for making decisions on the reconfiguration of routing and message-passing. It also learns from experience and uses its new knowledge in its prediction and decision-making [58].

The biological and ecosystem metaphors provide interesting parallels to the conceptualizations and descriptions of the G-Nodes. The overall GEMOM system architecture has a structure similar to that of a complex adaptive system that utilizes autonomic systems mimicking biological auto-immune systems at the microscopic level (operational level in this case) and that utilize the behaviours of an ecosystem of disparate entities at the macroscopic level (managerial level in this case). Biological and ecological systems maintain system integrity by reacting to foreseeable changes, adapting to unforeseeable changes, or dying. The adaptations and responses can be at a macroscopic ecosystem level (e.g., system or species) or a microscopic biological level (e.g., molecular, cellular) [43]. Hence we can consider GEMOM as having a genetic makeup [7][59].

B. *Reliability of Message Sourcing and Delivery*

GEMOM supports redundant message feeds (topics and namespaces) and redundant delivery paths (message communication architecture). In the event of failure, switch-over to a redundant resource would be transparent to the end users, with no information loss. As well as entire Message Broker redundancy, GEMOM offers the redundancy of certain subsets or messaging segments. As part of its self-healing functionality when a backup resource is mobilised to carry messages, other nodes, feeds or paths are identified as mirrors (backups) in case of further failure in order to maintain the same level of resilience. This ensures that there are no single points of failure even as new nodes become compromised and so rendered alien and isolated, or even as their rights are revoked.

C. *Replicating Structural and Dynamic Properties*

One particular GEMOM setup might be configured with a certain security layout or profile in place. GEMOM ensures that the security profiles of the overall system and individual message paths and dynamics are not compromised as a result of failovers. Namely, GEMOM is capable of fully replicating structural and dynamic properties of security policies representing different security layouts or profiles.

GEMOM utilizes a novel Adaptive Security and QoS model that consists of a continuous cycle of monitoring, assessment, and evolution to meet the challenges of the changing environments and threats [7]. This involves gathering contextual information both within the system and the environment, analyzing the collected information and responding to changes by adjusting security functions such as selecting suitable encryption schemes, security protocols, security policies, security algorithms, different authentication and authorization mechanisms, etc. Information gathering for adapting is implemented by using anomaly detection and security monitoring services that register external influences of the environment [60].

The GEMOM project has investigated a number of possibilities in connection with the self-learning capabilities and optimization approaches with respect to resilience. In that sense, the algorithmic approaches for this involved the use of genetic and evolutionary techniques at some level as partial elements for the overall solution.

D. *Notion of Faults in GEMOM*

The term 'fault' in GEMOM refers to a very general concept covering network faults, congestion, and security vulnerabilities, etc. Faults can manifest themselves in the deterioration of the functional profile of the informational system, of the volumetric profile, or of the security profile. Mitigation or resolution of faults requires the availability of support for a reconfiguration back to an efficiently working system.

GEMOM is able to rectify such vulnerabilities to faults by dynamically deploying a new instance of the messaging system. GEMOM is resilient and able to utilize redundant modules, hot-swap or switch-over without information loss. These resilience-features allow specialist, independent system actors (viz. watchdogs, security and situation monitors, routers, and MOM clients) to remove or replace compromised nodes from the broader network instantly and without compromising higher level functionality and security.

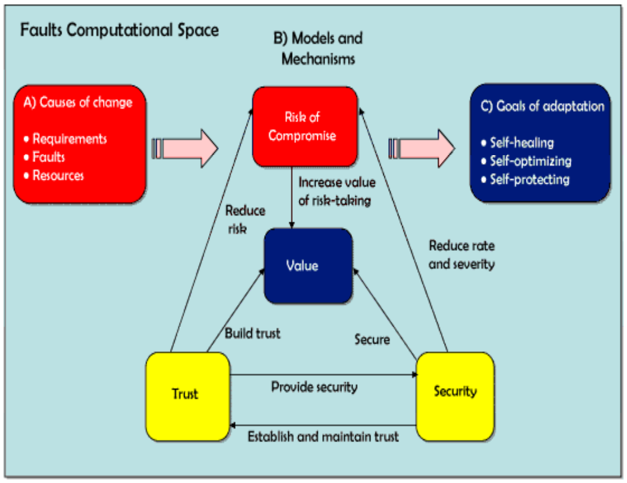


Figure 2. GEMOM Fault Computational Space

The GEMOM fault computational space in terms of risk, trust and security is shown in Figure 2. The space can be categorized as:

A) Causes of change are due to system complexity and environment, in which the GEMOM system has to deal with and needs to adapt to them. These include requirements, resources and faults. Fault tolerance ensures availability by guaranteeing maximum continuity of a service and an acceptable level of service when faults occur. Since the concept naturally lends itself to adaptability, fault and intrusion tolerance mechanisms can be used to increase the availability of a system. At an abstract level within the maintenance of high levels of QoS, reliability and resilience in the presence of threats to critical infrastructures, resources are sometimes restricted in terms of financial budgets and computing infrastructure. A comprehensive risk assessment might therefore be appropriate to assess what components and/or levels require the highest level of protection in order to reduce the overall effect, or at least to mitigate, the worst threats of those events whose effects it would take the longest time to recover from. The ideas within GEMOM can play a considerable role in this area as some of this intelligence can be covered by Service Level Agreements (requirements) through the use of Artificial Intelligent approaches such as naive Bayes and conditional probabilities to predict where within the overall system the highest level of risk - in terms of cause and effect - might be concentrated.

B) Models and mechanisms: A model here means a set of functions used to describe features of either a particular element or multiple elements of a client, e.g., the range of a particular element's possible values, the probability distribution of the number of sub-elements, or the presence and absence of a subset of elements, the structure of elements in a message body. Our models here consist of risk, trust and security adaptations with associated algorithms. Risk is an inherent part of any security or trust system. Risk adaptive security is an emerging technology that adapts its decision based on a computation of security risk. Trust is a necessary prerequisite basis for a decision to interact with an entity. Trusting an entity is always associated with risk since there is always a chance that the entity will behave contrary to expectations. Trust reduces risk, builds confidence in the value of a business and provides security. Security supports the process of establishing and maintaining trust through the provision of a secure and trustworthy environment. Security also reduces the rate and severity of compromises by continuously adjusting and responding to constantly emerging and changing threats. Based on the above described relationships of cause and effect as a foundation, GEMOM adapts and combines adaptive risk-based security, trust-based security, and security-based trust. The effect of this combination is to increase the strength of security and the degree of trust in the messaging system, and to reduce the rate and severity of compromises [59]. Many promising approaches exist, bringing together tools from control theory, biology, economics, utility theory, artificial intelligence, etc. A model here means 'a set of functions used to describe features of either a particular element or multiple elements of a client, e.g., the range of a particular element's possible values, the probability distribution of the number of sub-elements or the presence and absence of a subset of elements, the structure of elements in a message body'.

C) Goals of the adaptation (self-healing, self-optimizing, self-protecting) are adapting topology, resource usage, 'fidelity', etc. The self-healing capabilities can prevent and recover from failure by automatically discovering, diagnosing, circumventing, and recovering from things that might cause service disruptions. The self-optimizing capabilities enable the system to continuously tune itself – proactively to improve on existing processes and reactively in response to environmental conditions. Its self-protecting capabilities enable the system to detect, identify, and defend against viruses, unauthorized access, and denial-of-service attacks [12]. The driving factors and the needs for dynamic adaptation can be summarized from [59] as follows. The driving factors for adaptation are (i) the convergence of advanced electronic technologies (wireless, handheld, sensors, etc) and the Internet, (ii) the promise of instant access to data and computing no matter where or when, (iii) the changing nature and behaviour of the environment, and (iv) the need for systems to operate in the face of failures and attacks. The need for dynamic adaptation is due to (i) the heterogeneity of hardware, network, software, etc., (ii) the dynamics of the environmental conditions, especially at the wireless edge of the Internet, (iii) the limited resources (such as battery lifetime), and (iv) the software adaptation technologies for detecting and responding to environmental changes, and strengthening self-auditing capabilities of 'always-on' systems.

E. *Self-Adaptive Agent System*

The combination and integration of MOM and self-adaptive agent-based systems in GEMOM, resulting in resilient MOM, render a number of advantages. The self-adaptive agent has the following properties [61]: (i) autonomy, which allows it to operate without the direct intervention of humans or other external systems and to have some kind of control over its actions and internal state, (ii) social ability, which allows it to interact with other agents (possibly humans), (iii) reactivity, which allows it to perceive its environment and respond in a timely fashion to changes that occur in it (the environment), and (iv) pro-activeness, learning, and adaptiveness, which allow it to exhibit goal-directed behaviour by taking the initiative, to learn when reacting and/or interacting with its external environment, and to modify its behaviour based on its experience.

Consequently, the GEMOM components incorporating all these properties have self-adaptive behaviours, built-in capabilities for autonomous operation, monitoring their environments, reasoning, and communicating with other agents and human users. Self-adaptive systems require high-level dependability, robustness, adaptability, and availability. GEMOM meets these requirements by reaping the benefits of agent-based message brokers and overlay nodes: reliability via self-healing, performance via self-adaptation, security via self-protection.

IV. Self-optimization, Self-healing and Scalability

In this section, we investigate the self-optimization, self-healing and scalability functionalities of GEMOM.

A. *Optimizing Security and Protecting Networked Systems*

In GEMOM, self-optimization means making run-time adjustments to the operation of the system so that the values of certain selected operational parameters meet, or get closer to, their preferred range. Typical parameters are the usage of bandwidth, computational power and speed of message delivery.

GEMOM allows for the persistence of an optimized setup: new sessions can be established over a newly evolved topology. Redundancy can be used as a safety measure to secure continued, uninterrupted operation in cases of hardware failure or overload, or a DoS attack, yet being utilized at the expense of computing power, hardware and bandwidth. Self-healing can be seen as a ‘sibling’ of self-optimization, where the structure of running nodes, tasks, and communication paths are adjusted as a response to failure-type events, in order to re-establish an initial system structure equivalent pattern. Equivalence in this case assumes functional, resilience and security parameters.

Optimization can be achieved by autonomous agents inside each node, by a central agent for the entire system, or a hybrid approach. The knowledge of the autonomous agents is typically limited to the node itself and its immediate neighbours. These agents normally follow a set of empirical rules that are known statistically to make the network perform reasonably well if all nodes adhere to them. If a central agent is deployed it will have knowledge about the entire network and all the communication paths, and thus be better equipped to make decisions that are globally optimal. The system is then, however, exposed to attacks or failures that could disrupt the communication between the agent and one of the nodes, whilst systems based on autonomous per-node optimization agents are more robust and self-healing than centrally managed systems.

B. *Evolution Algorithms*

The GEMOM system utilizes one overall managing entity per Broker core, running on the same host platform. The manager performs both optimisation and healing in terms of starting a replacement broker in case of malfunction. The manager runs as a parent process of the broker core. The manager process manages routing and group replication. This is based on communication with other brokers. Evaluation algorithms have been developed to decide optimal values for various metadata and routing properties, balancing considerations for:

Manageability: Each node performs a limited and well-defined set of functions, and only has responsibility for a manageable number of groups of nodes.

Scalability and resilience: In a system of cooperating brokers, publishers and subscribers, there have to be sufficient replication of paths and messages to avoid overloading specific servers, and to be able to sustain random and sudden fallout without interruption of service.

Economy: A system of co-operating nodes has to use as little bandwidth and hardware resources as possible.

Consequently, two approaches are used to achieve resilience and evolution in GEMOM, one being the management of reserve resources in such an overlay network, the other being empirical correlations.

C. *Redundant Publishers*

Published messages often originate from outside of the environment when a user is subscribing to the messaging system. They are received through feeds that can be compromised. GEMOM allows for the application of redundant feeds sourcing data from the same or different provider (publisher). Switch-over is instantaneous with minimal loss of other features providing feeds are compatible in terms of capabilities.

D. *Quality of Service*

GEMOM as middleware is well-suited to provide an abstraction for QoS towards the application. GEMOM addresses the QoS requirements of applications through service level agreements, which are managed by the middleware. The supported QoS metrics and parameters include message latency, transaction rate, loss rate, delivery semantics, message ordering, message delay variation, and expiration time. The management of the QoS requirements given by an application is performed by the same regime that manages security-related properties, i.e., using the extended concept of faults.

E. *Scalability and Resilience*

In GEMOM, scalability and resilience are achieved via co-operating message brokers, publishers and subscribers with sufficient replication of paths and namespaces, and clustering of topics into groups of one or more, with group replication. This allows the system to avoid overloading specific brokers, and to sustain random and sudden fallout without any interruption of service.

For scalability with respect to message volume, GEMOM provides switch-over to redundant components preserving, and not compromising, scalability.

V. Holistic and Systematic Adaptive Security

The GEMOM [7] system is a resilient and scalable MOM that supports adaptive security-management by a monitoring functionality based on security and QoS metrics. Adaptive security in GEMOM refers to a security solution that learns, modifies existing functions, and adapts to the changing threat environment without sacrificing too much of the efficiency, flexibility, reliability and security of the system.

A. *Tangible and Demonstrable Improvements in Security*

Within the current MOM technologies the security requirements are somewhat rigid and do not form an integral part of the overall capability in a scalable and flexible way. Unfortunately, the state of the art in developing credible and sufficient security requirements in a holistic way is still in its childhood. Improvements to security are through a security monitoring system supported by appropriate security metrics, explicit enhancements to the authorization process, the explicit provision of resilience, and the provision of an associated software suite to support the discovery of vulnerabilities in systems that deploy GEMOM. Multiple modes of authentication and management of the authentication strength during authorization processing and fine-grained authorization of GEMOM usage at broker, cluster, topic or message level are provided.

B. *Adaptive Authentication and Identification system*

Authentication in GEMOM is based on multiple, possibly redundant, mechanisms and may include passwords, smart cards, uni-modal biometrics, and their fusion. This solution allows for interoperable context-sensitive security mechanisms, where the security mechanisms adjust to the needs defined by applications and the security level requested by the transaction. The authentication mechanism developed in GEMOM makes use of authentication policies that can be dynamically adapted according to needs, for example to take into account application needs for authentication security level. Appropriate security metrics are being developed to offer evidence for the adaptive security management. Flexibility is achieved by adding a normalized strength of authentication to the actor, before it is authorized as a pair (*Actor*, *Authentication Strength*), the *Authentication Strength*, which is an aggregated metric that depicts the overall security level of the authentication solution.

GEMOM Identity and Authentication Management Components: as depicted in Figure 3, the GEMOM identity & authentication management has the following functional components:

- A GEMOM Identity Provider (IdP) Service in charge of managing all identity related data, users' authentication and the provision of an entities' Attribute Service, that is of a set of functionalities through which additional entities information can be searched and provided. The GEMOM IdP Service is in charge of managing trust management relationships between GEMOM systems and other systems with which a federation is configured. The GEMOM IdP Service also provides the capability of using different back-end silos so that existing entities data sets can be reused;
- A GEMOM Authentication Service Client to be used to access the GEMOM IdP Service for all entities authentication and attribute needs. As depicted in the figure the GEMOM Authentication Client has to be used both by the GEMOM Message Broker functional component, as well as from GEMOM application clients. The GEMOM Authentication Service Client takes care of managing all interactions with the GEMOM IdP Service, select the right authentication protocol, translation of security tokens, as well as submission of attribute queries and acquisition of entities attribute values.

As indicated in Figure 3, end-users can use different identity technologies and tokens, leaving to the GEMOM Authentication Service Client the job of properly managing the corresponding protocols, data and transactions.

Figure 4 provides a more fine-grained view of the functional components involved in end-user authentication and of the kinds of authentication credentials the end-user has at his/her disposal (username/password pair, X.509 [62] certificate, and smartcard).

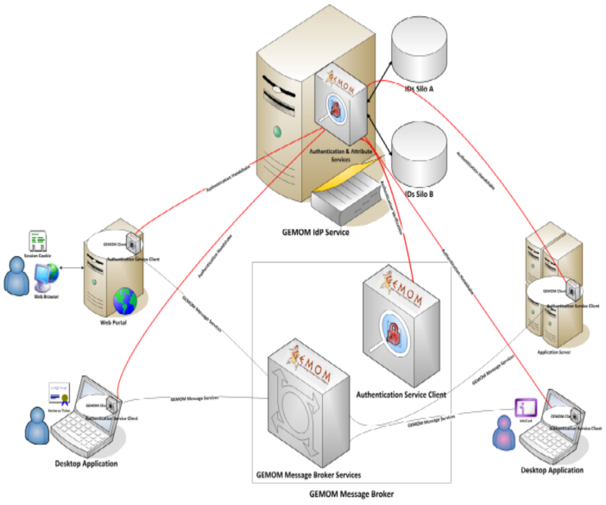


Figure 3. Identity Management and Authentication architecture

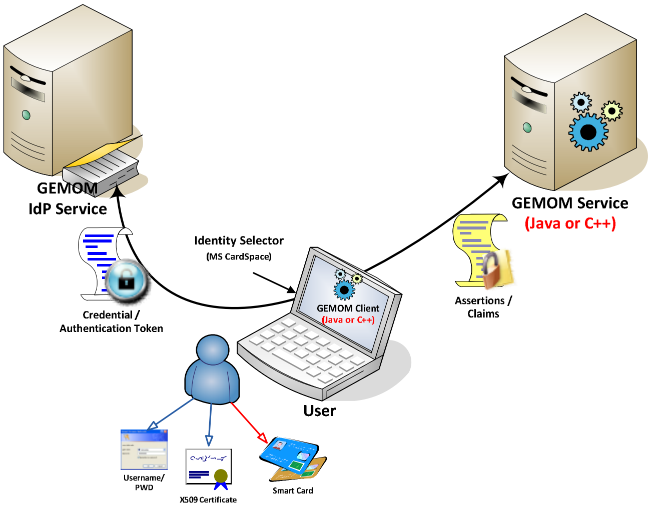


Figure 4. End-user authentication components

C. *Adaptive Authorization*

Authorization in GEMOM supports access rights to namespaces, cluster groups, topics and single messages, the application of access rights to a single message being the smallest level of granularity to which authorization rules can be applied. The GEMOM authorization model also supports multiple user roles, defining access rights and varying performance and reliability requirements depending on the type of user. It is the pair (*Actor*, *Authentication Strength*) by which the actor was authenticated that is a unit-entity that GEMOM authorizes. The GEMOM authorization process is carried out using this pair as a basic composite key, taking into account the following: (i) each user belongs to a group, and the basic strength of the user-authentication key is translated into a vector of strength of group-authentication pairs, (ii) the system is perceived as having certain multi-dimensional security profiles, and boundaries are defined in each dimension, (iii) an application is divided into an arbitrary set of modules, and an abstract notion of operation on a module is defined in which a module can allow an arbitrary number of operations to be performed on it. Access rights are defined for the pair (module, operation), and (iv) certain groups of users that are authenticated with strengths that fall into certain ranges are allowed to perform certain operations on application modules within certain periods of time, within defined context boundaries and within certain dynamic security boundaries. The development of adaptive features of the authentication, identity management and authorization processes is described in detail in [7][59].

D. *Adaptive Security Monitoring*

The GEMOM Security Monitoring System (SMS) is based on security level estimation mechanisms to enable resilience of the system. These mechanisms utilize security metrics, developed in a systematic security requirement decomposition process, introduced in [60], and enhanced in [48]. The security metrics compare the actual security level to the reference level set by security requirements of security-enforcing mechanisms or security functions [63]. Consequently, the definition of appropriate requirements, which address security, resilience, self-healing and evolution, has been the core activity in the development of GEMOM Adaptive Security Management (ASM) functionality.

The SMS includes measurement data collection mechanisms and interfaces to the system components under measurement, associated adaptive security knowledge repositories, metrics and trust, confidence and reputation information and suitable algorithms for using metrics. The SMS carries out security monitoring and supports ASM operations based on the on-line security metrics. The SMS is connected to the GEMOM Message Broker, Authentication and Authorization Module, Audit and Logging Module, QoS Accessory Module, Anomaly Detector Module, and memory elements, storage and network interfaces. In addition to the logs produced by the Message Broker, the monitoring system is able to monitor messages and metadata. Figure 5 depicts an example GEMOM subnet and information flow relevant to the SMS.

The collection of Basic Measurable Components (BMCs) of the security metrics for the GEMOM Security Monitoring System have been introduced in [48] along with a security metrics development methodology, analysis of its benefits and shortcomings and a framework for calculating trust, confidence and trustworthiness of the metrics. BMCs are the leaf components resulting from the security-requirement decomposition, an abstraction for a more detailed development of security metrics.

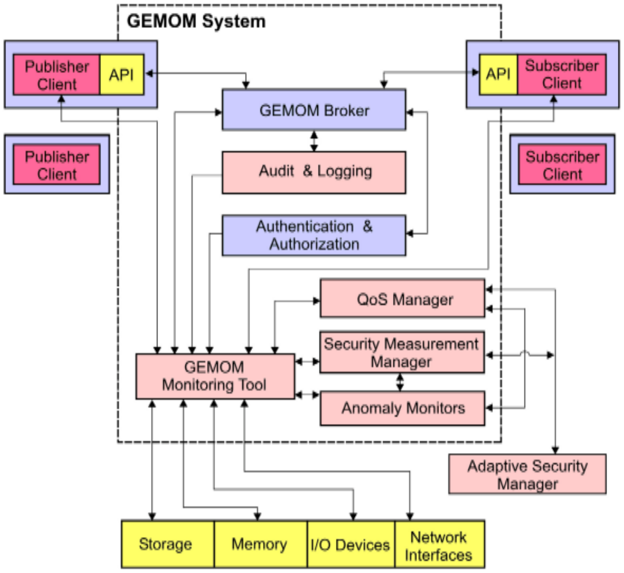


Figure 5. An example GEMOM subnet [64]

Metrics, rules and reputation information are configured by using the Monitoring Tools (MTs) user interface and are stored in a special database. An MT is connected directly to GEMOM Broker(s). Connection between the Monitors and Brokers is arranged via GEMOM Client Interface (GCI), see Figure 6 [65]. Other modules use the GEMOM PS mechanism for communication: publishing and subscribing to relevant topics in a *measurement namespace* [64]. Using this mechanism, the MTs connect to Authentication and Authorization modules, QoS Managers, Anomaly Detector modules, Security Measurement Managers, as well as relevant-use and free-memory entities, storages (hard disks, memory sticks), network interfaces and Input/Output devices (e.g., keyboard).

The following attributes form the minimum set of needed configuration parameters: metric ID, input and output data of the metric, metric calculation formula or heuristics, threshold value(s), and timing information. At the managerial G-Nodes level, the Monitoring Tools co-operate with the ASM. The ASM monitors security, analyses its details, plans adjustments, and executes the planned adjustments through a global control loop, using both manual and automated information. Thus, the ASM manages the behaviour of the overall system from the security point of view. The monitor modules can be updated and enhanced, and new modules can be integrated during runtime operation, supporting the ASM.

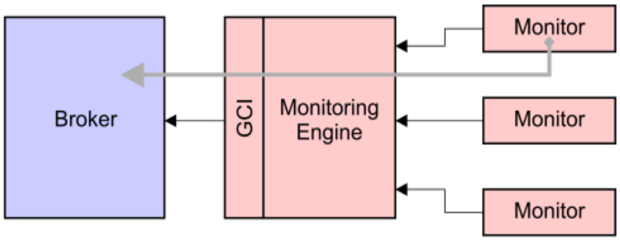


Figure 6. Communication between the Broker and the MT [66]

The GEMOM SMS and the ASM utilize a holistic State of Security (SoS) concept [64][65]. SoS is a time-dependent estimate of the system's security performance level based on the appropriate collection of security metrics that is calculated initially and when triggered. The concept can be used to configure the management of the used security metrics. There are five steps in the estimation process of the SoS:

1. Definition of the initial SoS is done using appropriate security metrics.
2. The current SoS is measured whenever triggered by a timer, an attack, an anomaly or a manual request.
3. Past and current SoS is compared to offer input to the trend estimation in decision-making.
4. The initial SoS is adapted according to decisions made by the ASM functionality.
5. A future SoS is predicted to enable proactive Adaptive Security Management.

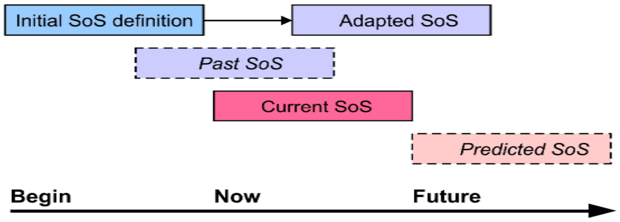


Figure 7. A timeline visualization for SoS estimates [64]

Figure 7 depicts the visualization of the different types of SoS estimates. The predicted SoS is based on the analysis of the past history of the SoS levels and threat and vulnerability trends. The predicted SoS is useful when carrying out proactive operations to ensure the high resilience of the system.

E. Self-Protection

A self-protecting system, as defined by IBM [66], can anticipate, detect, identify, and protect itself against threats, unauthorized access, and denial of service attacks. GEMOM as an autonomic MOM has to implement self-protecting capabilities that can detect hostile behaviours as they occur and take corrective actions to make the system less vulnerable. In the GEMOM setting (see Section III), the self-protection is managed either at a single entry point (a micro property), which gives each node authorization, by a coordinated defensive group attack of the other nodes alone (a macro property), or by a combination of the two (defence-in-depth). Figure 8 shows these entry points and their properties.

Most intrusions can be managed by triggering a one-shot behaviour of the GEMOM system. However, the GEMOM system has constantly to be alert, so the degree of protection over time (ongoing) is important [59]. The proactive identification of and protection from, arbitrary attacks are achieved via the combination of anomaly-based self-protection [40], and security monitoring and measurement.

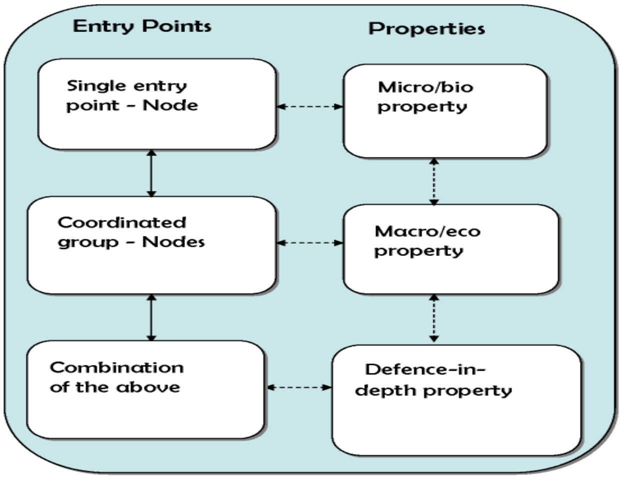


Figure 8. Self-protection entry points and properties

A key component in self-protection is the integration of mechanisms to support the detection of anomalies such as high message rates, degradation of broker performance, e.g., in the context of DoS, and of services to support the detection of anomalous message content in appropriate cases. Detectors are divided into different functions e.g., link-state detection, message-rate computation, bottleneck detection, and overall system representation. For example, the detection of and reaction to, link-state faults, brokers continually send probes to its clients and peers. Metrics such as loss rate and message delay are measured from probes. However, when anomalous metrics are detected, the frequency of probing is increased. A relevant action for example, when a sequence of probes are lost or metrics are above the acceptable range, is that another broker is set as a relay broker between the nodes experiencing a faulty connection in an attempt (not assured as the topology of the underlay is probably not known to the MOM system) to circumvent the failure. In a worst case, if a relay broker can not be found, then the workload of this original broker needs to be taken over by a pre-allocated mirror broker. The distinction between link-failure and broker-failure is established by monitoring between brokers either over the subnet or over what are expected to be disjoint paths in a wider network (Whether a subnet or a wider network is used to support the overlay depends on the application and the nature of the resilience required.). This probing approach maintains a view of the whole system, and scales to a limit of tens of nodes, which is considered enough for our applications. Different approaches to bottleneck-detection are under investigation. The component being integrated currently uses Markov models to predict the values of different individual measurable resources of the broker (broker CPU, message rate, subscription rate etc.) and uses a Naïve Bayes classifier, trained on system operational data, to detect a bottleneck based on the predictions.

The optimal allocation of the workload among brokers (see self-healing later), and redundant mirroring provides an enhanced toleration of burstiness from Flash Events (FE) and DoS attacks. DoS detectors are distributed among the overlay nodes to localize and mitigate DoS attacks. The analysis is achieved by collaboration between overlay nodes. The detection is employed both locally in each node, and by globally monitoring the correlated measurements. We adapt new detect and defence mechanisms to the architecture and context of our federated PSMOM, and some of the mechanisms being based on previous attempts for DoS defence, e.g., [67][68].

Anomalies in the messages in a MOM system can be caused by attacks that propagate through or target on the system. In the PSMOM domain we can profile the normal messages based on the collective characteristics among messages from similar topics, and detect outliers. We are employing a multi-model approach [69], and the models profiling normal messages are chosen based on the system requirements, e.g., whether content is encrypted or not for inspection, and statistical characteristics of messages.

Different detectors with different levels of functions can raise alarms simultaneously. The Management Layer will correlate these alerts and choose a proper reaction. An Overlay Manager is responsible for a range of functions to improve performance and resilience at the management layer. For example, a link fault with long delay and high loss rate might be accompanied by a simultaneous DoS attack alert in this case the Overlay Manager will prioritize the response to DoS attack and suppress the link fault alert for a short time. Experiments based on DoS attacks are being created and the detection and reaction mechanisms validated.

Figure 9 depicts the GEMOM data collection architecture. The sensors and detectors are distributed both inside the GBroker, monitoring messages and extracting features of each topic.

Inside the GBroker, data collection is performed for each topic by computing and updating it during the message-processing stage. An anomaly in a topic can be poisoning the entire cluster. Topics in the same cluster (i.e., topics in the same messaging path connected by the switching GBrokers) will also exchange anomaly detection information through the GBrokers. This can be seen as a simple form of dynamic-taint analysis. That is why there should be a cluster-correlator to decide on the actions of the whole cluster (i.e., Cluster Correlator). This suppresses any actions proposed by the individual switching GBrokers in the cluster, and replaces the actions with the actions decided on by the cluster manager. Correlation of anomalies between switching GBrokers in a cluster is performed by a single cluster correlator, which can be centrally located, or elsewhere, e.g., the last switching GBroker in the path of the cluster. The consumer and producer clients can only send their information to the cluster correlator, which sits on top of the base anomaly detector in the GBroker. The cluster anomaly detector is the anomaly detector that is responsible for taking decisions, and it has the capability to override/suppress reactive settings in the decision-making policies for the topic of the individual GBrokers. Reactive actions may be necessary at individual GBrokers, as some anomalies may be so clear cut and so dangerous, that to wait for the decision of the cluster correlator may be too late. The location of the Correlator is outside of the GBrokers, as shown in Figure 9.

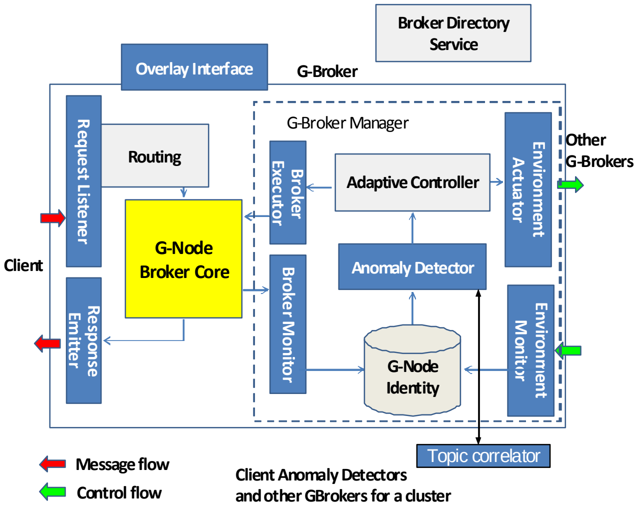


Figure 9. Data collection in context of the GEMOM Architecture [1]

F. *Pre-Emptive and Automated Run-time Vulnerability Management*

Including errors to force a reaction from the system can be a way of making the system more robust. Mechanisms to induce error conditions in the services and to expose vulnerabilities before they happen have been developed. The GEMOM vulnerability management toolkit has been developed to detect vulnerabilities in the deployed GEMOM system as configured by the users. To identify previously unknown faults and loopholes, effective techniques to generating inputs that induce failures have been developed. This is called fault injection or fuzzing technique. Compared with traditional software-testing techniques, fuzzing has been found effective and cost-efficient. It is becoming a legitimate aspect of robustness and security testing.

There are several available fuzzing software and libraries for widely used protocols. However, considerable research is needed into the construction of fuzzers that can capture, without too much manual intervention, the diversity of applications associated with it. Each situation, protocol, or application causes new issues that need to be addressed. The best approach to testing varies between projects. The experience of the testers has a significant bearing on the efficacy of the testing [70].

G. GEMOM Software Security Assurance in General

A secure and resilient system solution based on best practices is not a sufficient security solution by itself. The security, trust, dependability and privacy requirements of a system and applications must be comprehensively analyzed from a security risk perspective and adequate security assurance methods must be used. Security assurance includes a wide variety of activities from security analysis to security testing and monitoring [64]. In recent years, the understanding and tools for security assurance have developed in leaps and bounds, enabling the functional testing and monitoring of security to be part of the normal product development and maintenance processes. Comprehensive risk-aware security analysis guides testing and monitoring activity. Security analysis may include: the investigation of threats, the specification of security requirements, the modelling of attack, the investigation of vulnerability, and the assessment of security level and performance using adequate security metrics. Most of the on-line and off-line security metrics developed for GEMOM can be utilized for the security assurance.

H. Self-Healing

The atomic unit to offer resilience is at the topic level. For performance reasons, QoS monitoring in GEMOM is at the namespace level by default, though monitoring at the topic level is possible; this is particularly relevant for monitoring anomalous individual message content when appropriate. Namespaces are also further grouped into items, to support scalability of the resilience decision-making [71].

The mirror and relay concepts are used in resilience management. The primary namespaces are the namespaces that are handled by the GBroker. A GBroker will also act as a mirror for other broker namespaces and these namespaces form the mirror namespace set, also held at the broker. A primary namespace tree can be partitioned into many sub-namespaces for mirroring, and each sub-namespace assigned to a different GBroker for mirroring. The namespaces assigned to a broker for mirroring is called the mirror namespace set.

In practice, assured delivery is a common requirement of MOM users. This means that if a subscriber loses connection (e.g., through border gateway failure between the broker and the subscriber, or subscriber site failure) then the MOM has to retain messages until the subscriber can later pick up the messages. A time limit can be put on the retention in the MOM, but the relevant policy is application dependent. This means that if a broker has to take over the function of all or part of another broker then a lot of state information may need to be available. This also means that, in contrast to working on P2P systems, it is sensible to pick a mirror candidate prior to failure and not delay the decision of alternatives till the time of failure so that state can be tracked.

There are three steps. First, the whole workload of namespaces is partitioned into items, and each item is a subset of namespaces. Items are disjoint. Assuming those items are known, we are interested in the allocation or re-allocation of such items to each broker in the MOM system. Second, an optimal allocation of items over the overlay is determined. A combinatorial auction mechanism has been implemented for finding the optimal combinations of items to be allocated to each broker (i.e., the winner determination problem). The brokers act as bidders and bid for sets of nodes and the MOM system acts as the auctioneer. In a complex problem like providing resilient service, this auction based allocation mechanism gives brokers some degree of freedom in applying different preferences to choose the items they bid on. The system is able to find an optimal solution, from possible combinations of all the bids placed by brokers. By optimal combinations we mean the best allocation of items, where the risk of brokers being saturated is estimated to be the least, and where the chance of brokers generating the maximum revenue to the system is estimated to be optimal. The price to bid is based on the risk function that estimates the probability of exceeding the GBroker's capacity by exploiting the correlation between different items using the variance covariance matrix of the namespaces of the workload. Since positively correlated items have a super-additive effect on consuming resources of the system, the bidding function put preference bidding on non-positively correlated item combinations that posses less risk.

The third step is to provide redundancy and reactive solutions to adapt to system faults and degradations. After the initial allocation of the workload in order to react to possible failures and service degradation, we compute solutions to re-allocate workload and introduce redundant mirror items with available resources. This is done by applying either extra rounds of auctions or by an optimal [67] search again based on the risk function. The solutions are saved in a case data base to support timely reaction.

VI. Enhanced Interoperability and Integration

A PS MOM-based system can be modelled and re-factored with ease at run-time as well as at design time. The exchange of messages is connectionless and asynchronous. The PS MOM system is inherently extensible, etc. These features make PS MOM a powerful base for resource efficient implementation of scalability, resilience and management of vulnerabilities in a distributed system. GEMOM as a PS-based MOM has these properties.

GEMOM further supports better interoperability and integration of information systems by allowing actual instances to be configured, so various functions are subcontracted to one or more separated external or federated entities. This separation allows the use of different security layouts for different individual services or clusters of services. The most important advantages of this approach are:

- Focus of different functional clusters on different issues and core competences. For example, consider the dynamism of messaging. The rapid changes in message volume is often such that, in terms of economy of the offered solution, it is common to separate pure messaging, authentication and security related services. Pure messaging (e.g., without bundling) in highly scalable environments can be very resource intensive;
- Message brokering is not compromised while an incident is flagged on one or more security monitors awaiting resolution;
- Security functions (e.g., authentication, authorization, key management, security metrics processing) can be implemented to far higher standards and be less resource intensive by separating them from the other parts of the system;
- Non-intrusiveness of the monitoring system: the monitoring system does not cause any harm to the normal operation of the measurement target system and does not affect the measurement results; and
- Bridges and adapters for industry standard messaging systems: GEMOM deploys a framework for integration with other messaging platforms and information service busses. GEMOM is interoperates with platforms such as JMS [72], Tibco's RV [73], Reuter's Triarch, and IBM's MQ Series [74] through the provision of bridges and adapters.

VII. Adaptive Integration Architecture

The GEMOM framework includes adaptive integration functions and tool-sets. This section briefly describes the integration of these tool-sets using the Adaptive Security Manager (ASM) as an example of how these tools can be integrated. Figure 10 shows the adaptive integration architecture.

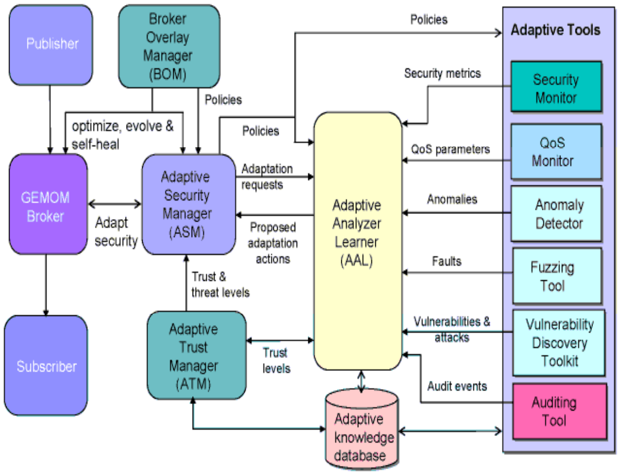


Figure 10. Adaptive integration architecture

The Broker Overlay Manager (BOM) provides resilience and adaptability by utilising an overlay network of publish/subscribe MOMs and can function despite lack of privileged knowledge of the underlying infrastructure. Figure 11 shows the conceptual structure of the BOM. As shown in Figure 11, the BOM uses models for supporting resilience, optimisation, evolution, QoS and security. It provides two methods for interactions with the models: (i) model can be a plug-and-play rule loaded into BOM process, and/or (ii) model can be hosted in an independent application and communicate with BOM over dedicated GEMOM message broker. It also provides global mechanisms (such as global policies for adaptation, optimization, evolution, and self-healing) for models' functionality to be able to alter the behaviour of the GEMOM system at the level of machine, broker, client, namespace or topic.

The BOM performs autonomous adjustments to the run-time configuration of the system in order to preserve and maintain optimal and uninterrupted operation, recover from partial breakdowns and use newly acquired information about the topology of the system and its surrounding in order to evolve into better system. It considers the provision of self-optimisations of the system both at the level of the overlay manager and at the sub-components of the overlay manager. In the context of MOM system, different approaches are used to support resilience, multi-path redundancy, reactive routing, path disjointedness and namespace mirroring. A Case Based Reasoning (CBR) approach that allows reactive response was also developed, where the case base is created by repeated use of the optimisation process. Additionally, an approach based on the analysis of system correlations is proposed as a technique to respond to a complex dynamic system.

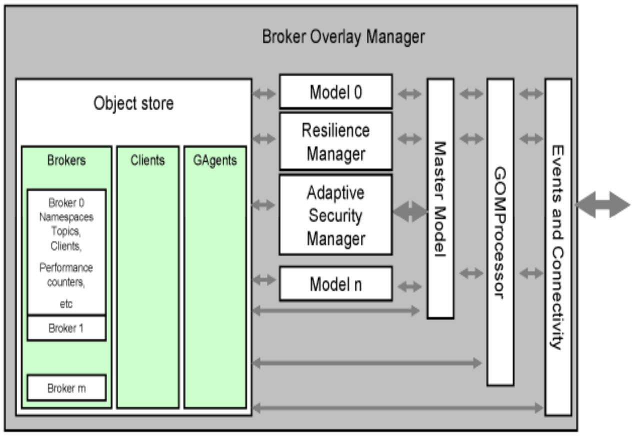


Figure 11. Conceptual structure of the Broker Overlay Manager

Finally, the BOM supports mechanisms for adding G-Nodes; for measuring QoS between overlay components, publishers and subscribers and deciding what action to be taken to mitigate loss of QoS or breakdowns; for discovering and communicating with other components in the overlay network; for evaluating the performance of the system in the context of the monitored performance; for establishing the state of the overlay network; and for making decisions on the reconfiguration of routing and message passing. It also learns from experience and uses its new knowledge in its prediction and decision-making.

The ASM [59] manages and controls all the security components as an integrated GEMOM security infrastructure. Its security services adapts to the rapidly changing contexts of the GEMOM environment. The ASM model consists of a continuous cycle of monitoring, assessment, and evolution to meet the challenges in the changing environments and threat situation. It utilizes contextual information and decision-making to select the 'best' security model for a given situation. The ASM includes the integration of adaptive control loop functions (monitoring, analysis and response), and tool-set, elastic-fine-grained adaptive authorization, adaptive authentication and federated identity management, and tools and processes for pre-emptive vulnerability testing and updating. While each component implements a local adaptation control loop, the ASM implements a global adaptation control loop. Here the sensors are Anomaly Detectors, Security Monitors, Fault Detectors, QoS Monitors, and Auditing and Logging.

The ASM component provides adaptive security and trust through changing security policies, algorithms, protocols and encryption schemes according to context parameters, such as environment, system threats, user threats, trust levels, usage, security and trust metrics, faults, and quality of service. Fault and intrusion tolerance mechanisms are used to increase the availability of a system, and previous faults caused by the user are used to increase suspicion-level. The system threat-level and the user suspicion-level are maintained by and obtained from the Adaptive Tools (like Security Monitor, Anomaly Detector, and Fuzzing Tool). Figure 10 depicts the relationships between the ASM and other components.

The Adaptive Analyzer and Learner (AAL) component implements the analysis function of the adaptive control loop and analyses the collected information using established analysis and decision-making methods. It processes the collected data, along with other information (e.g., security policy, threat levels, or trust levels boundaries) and proposes actions to bring about a new stage. The Adaptive Tools sense and gather contextual information both from within the system and from the environment. They distribute information about the security environment to the AAL and adaptive database. The Vulnerability Discovery Toolkit allows the identification and understanding of the risks and vulnerabilities of the GEMOM system and the forming of trust solutions to address the risks and vulnerabilities. The Fuzzing Tool allows an effective black box testing technique to be used for finding security flaws from software.

The Adaptive Trust Management (ATM) model [59] is a compromise-based trust model that provides information about any attack on the system and the nature of that attack for the purpose of establishing whether, and if so, how different properties of the system have been compromised. In addition, it establishes whether these properties can be trusted for a particular purpose in spite of being compromised and to what degree these judgments should be suspected or monitored. It also incorporates a framework for calculating trust, confidence and trustworthiness of the trust and risk impact metrics.

This adaptive integration demonstrates GEMOM's solution that consists of a continuous cycle of monitoring, measurement, assessment, optimization, self-healing, adaptation and evolution to meet the challenges in the changing environments by (i) provision of self-optimisations of the system both at the level of the overlay manager and at its sub-components, (ii) combining adaptive risk-based security, trust-based security, and security-based trust, and (iii) integrating different metrics, assessment and observation tools.

VIII. GEMOM Prototyping and Validation Results

The GEMOM project has prototyped: a full featured message broker, transparent completion and encapsulation publishing framework, adaptive security implementation (such as authentication, authorization, key management, and identity management), MOM Intelligent Fuzzing Tool, Security Monitoring Tool, and configuration and deployment of management and development process tools. The project has also developed the following demonstrators: Interfaces for enhanced resilience, QoS and security, security and QoS monitoring system, Integrators with well-known commercial MOM systems (JMS, Tibco's, Reuters, and IBM's MQ Series), and Broker Manager Agent without and with optimization.

These GEMOM prototypes have been validated in five case studies: a collaborative business portal, a dynamic linked exchange, a financial market data delivery system, a dynamic road management system, and a banking scenario for transaction processing. This validation and evaluation process allowed the core GEMOM platform and innovations to be tested. Different scenarios represent differing specific requirements and needs. GEMOM maintains some core requirements: security, performance, speed, and scalability; as a result, the overall approach through the delivery of all the specified enhancements will achieve these top-level needs. The validation and evaluation within real application use cases means that features and enhancements in terms of guaranteed delivery, security, QoS, and resilience were tested against the specific requirements of each use case.

By looking at a number of diverse applications, GEMOM can be tried for scalability, applicability and effectiveness across a wide set of market sectors. A typical way of thinking about this is guaranteed delivery – in some cases, as long as the transaction is completed in the Banking scenario, the requirement is met. However, in the financial market data scenario, guaranteed delivery can have strong time constraints, i.e., if a message is not sent within a given time period – it becomes redundant and the next one carrying the updated data should be sent. By looking at more than one application scenario, it allows us to test scalability, performance, etc., across many user scenarios.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, we have described a self-healing and secure adaptive messaging middleware for business-critical systems that is designed to adapt to dynamically changing environments. This middleware system, GEMOM, makes advances in the areas of resilience, self-healing, self-adaptation, scalability, integrated vulnerability management, better interoperability and integration of distributed business-critical systems, and holistic and systematic adaptive security monitoring and measurement. The combination and integration of MOM and agent-based systems, resulting in resilient MOM, advances the state-of-the-art. The system is capable of autonomously adjusting the run-time configuration of the system in order to preserve and maintain optimal, uninterrupted operation, recover from partial breakdowns and use newly-acquired information about the topology of the system and its surroundings in order to evolve into a better system, improving and increasing the strength of security and degree of trust in the system by combining adaptive risk-based security, trust-based security, and security-based trust, and improving the assessability and verifiability of the trustworthiness of the system by integrating different metrics, assessment and observation tools.

In our future work we plan to enhance the intelligent algorithms to improve the robustness, self-healing, self-adaptive, holistic and systematic assurance of adaptive security of the overall integrated system. In order for governments to fulfil their functions and do their job properly, it is important that critical infrastructures be resilient and secure in order to operate reliably and dependably in the presence of threats to them. The resilience and security of infrastructures are a high-priority requirement for governments. In our future work we intend to address this matter and apply our solutions to meet this requirement.

ACKNOWLEDGMENT

This research is in the context of the EU project GEMOM (Genetic Message-Oriented Secure Middleware, www.gemom.eu), 2008-2010, Grant Agreement No. 215327, approved by the EU Commission. The authors acknowledge the efforts of all partners who participated in the development of the GEMOM framework. Mr. Trenton Schulz of Norwegian Computing Centre, Mr. Petri Heinonen of VTT Technical Research Centre, and Dr. Peng Jiang and Mr. Jinfu Wang of Queen Mary University of London are acknowledged for their valuable and professional comments, and Mr. Cristoforo Seccia of TXT e-solutions Spa for his support and perseverance.

REFERENCES

[1] H. Abie, R. Savola, and I. Dattani, Robust, Secure, Self-Adaptive and Resilient Messaging Middleware for Business-critical Systems. In: The First International Conference on Adaptive and Self-adaptive Systems and Applications, ADAPTIVE 2009, November 15-20, 2009 - Athens/Glyfada, Greece.

[2] H. Li and G. Jiang, Semantic Message-Oriented Middleware for Publish/Subscribe Networks. Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III. In proceedings of the SPIE. In Proceedings of SPIE,5403:24–133, 2004.

[3] D. Lewis, J. Keeney, D. O'Sullivan, and S. Guo, Towards a Managed Extensible Control Plane for Knowledge-based Networking. Lecture Notes in Computer Science, Large Scale Management of Distributed Systems, Springer Berlin / Heidelberg, 4269/2006 (0302-9743):98–111, 15 October, 2006.

[4] S. Parkin, D. Ingham, and G. Morgan, A Message-oriented Middleware Solution Enabling Non-repudiation Evidence Generation for Reliable Web Services. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 4526/2007(0302-9743):9–19, 06 June, 2007.

[5] ESB, Enterprise Service Bus (ESB). Accessed May 30th, 2010, from http://www.sonicsoftware.com/psm/enterprise-service-bus/index.ssp.

[6] GEMOM, Genetic Message-Oriented Secure Middleware Technical Annex, Grant Agreement No: 215327, approved by the EU Commission, October, 2007.

[7] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham, and R. Savola, GEMOM - Significant and Measurable Progress Beyond the State of the Art. in Proc. ICSNC 2008, 26-31 October, 2008.

[8] E. Curry, D. Chambers, and G. Lyons, Extending Message-Oriented Middleware Using Interception, Proc. 3rd Int'l Workshop on Distributed Event-Based Systems (DEBS 04), 2004, pp. 32–37.

[9] D. Ghosh, R. Sharman, H. R. Rao, and S. Upadhyaya, Self-healing Systems - Survey and Synthesis, Decision Support Systems, Volume 42, Issue 4, January 2007, Pages: 2164-2185.

[10] M. Morandini, L. Penserini, and A. Peri, Towards Goal-oriented Development of Self-adaptive Systems. Proceedings of the 2008 international workshop on Software Engineering for Adaptive and Self-managing Systems, Pages:9-16, Leipzig, Germany, 2008.

[11] R. Laddaga, Self-Adaptive Software Problems and Projects. In SOFTWARE-EVOLVABILITY '06: Proceedings of the Second International IEEE Workshop on Software Evolvability (SE'06), Washington, DC, USA, 2006. IEEE Computer Society, pp. 3–10.

[12] D. M. Chess, C. C. Palmer, and S. R. White, Security in an Autonomic Computing Environment. IBM Systems Journal, Vol. 42, No 1, 2003 107-118.

[13] O. Kramer, Evolutionary Self-Adaptation: A Survey of Operators and Strategy Parameters. Evolutionary Intelligence, Springer Berlin / Heidelberg, Saturday, February 06, 2010.

[14] R. Hinterding, Z. Michalewicz, and A. E. Eiben, Adaptation in Evolutionary Computation: a Survey. In Proceedings of the Fourth IEEE Conference on Evolutionary Computation, Indianapolis, IN (1997), pp. 65-69.

[15] D. Miorandi, L. Yamamoto, and F. D. Pellegrini, A Survey of Evolutionary and Embryogenic Approaches to Autonomic Networking. Computer Networks, Volume 54, Issue 6, 29 April 2010, pp. 944-959.

[16] L. Rodero-Merino, A. Fernandez, L. Lopez, and V. Cholvi, A Topology Self-adaptation Mechanism for Efficient,Resource Location. In Lecture Notes in Computer Science, Proceedings of the 4th International Symposium on Parallel and Distributed Processing and Applications (ISPA 2006). Springer-Verlag, 2006, pp. 660–671.

[17] S. Dustdar, K. M. Goeschka, H. L. Truong, and U. Zdun, Self-Adaptation Techniques for Complex Service-oriented Systems.Fifth International Conference on Next Generation Web Services Practices (NWESP '09), 2009, pp. 37–43.

[18] P. D. Alencar and H. Weigand, Challenges in Predictive Self-Adaptation of Service Bundles. WI-IAT, Vol. 3, pp.457-461, 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, 2009.

[19] E. Gjørven, R. Rouvoy, and F. Eliassen, Cross-layer Self-Adaptation of Service-oriented Architectures. In Proceedings of the 3rd workshop on Middleware for service-oriented computing, 2008, pp. 37–42.

[20] P. Reinecke, K. Wolter, and A. V. Moorsel, Evaluating the Adaptivity of Computing Systems. Performance Evaluation, Special Issue on

Software and Performance, Volume 67, Issue 8, August 2010, pp. 676-693.

[21] G. Giannakopoulos and T. Palpanas, Adaptivity in Entity Subscription Services. 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009, Attents/Glyfada, Greece. IEEE Computer Society, pp. 61–66.

[22] C. T. R. Hager, Context-Aware and Adaptive Security for Wireless networks. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, November, 2004.

[23] A. Shnitko, Practical and Theoretical Issues on Adaptive Security, Novosibirsk State Technical University, and Adaptive security in complex information systems. In Proc. 7th Korea-Russia International Symposium on Science and Technology, (8023863):206– 210, 28 June - 6 July, 2003.

[24] S. H. Son, R. Zimmerman, and J. Hansson, An Adaptable Security Manager for Real-time Transactions. In Proc. 12th Euromicro Conference on Real-Time Systems,, 19-21 June, 2000, pp. 63–70.

[25] P. A. Schneck and K. Schwan, Dynamic Authentication for High-Performance Networked Applications. In Proc. 6th International Workshop on Quality of Service, 18-20 May, 1998, pp. 127–136

[26] J. Zou, K. Lu, and Z. Jin, Architecture and Fuzzy Adaptive Security Algorithm in Intelligent Firewall. In Proc. MILCOM, 2:1145–1149, October 7-10, 2002.

[27] H. Abie, P. Spilling, and B. Foyn, Rights-Carrying and Self-enforcing Information Objects for Information Distribution Systems. Lecture notes in computer science, Information and Communications Security: 6th International Conference, ICICS 2004, Malaga, Spain, LNCS 3269(0302-9743):546–561, 27-29 October, 2004.

[28] H. Abie, Distributed Digital Rights Management: Frameworks, Processes, Procedures and Algorithms. VDM Verlag, 02 October 2009, Paperback, ISBN-10:3639202961, ISBN-13:978-3639202960.

[29] A. Pietzowski, B. Satzger, W. Trumler, and T. Ungerer, A Bio-inspired Approach for Self-protecting an Organic Middleware with Artificial Antibodies. Self-Organizing Systems, LNCS, Springer Berlin / Heidelberg, Vol. 4124/2006, September 21, 2006, pp. 202–215.

[30] I. Djordjevic, S.K. Nair, and T. Dimitrakos, Virtualised Trusted Computing Platform for Adaptive Security Enforcement of Web Services Interactions. IEEE Int. Conference on Web Services (ICWS 2007), 9-13 July 2007, pp. 615–622.

[31] J. Luo, X. Ni, and J. Yong, A trust degree based access control in grid environments. Information Sciences: an International Journal, Volume 179 , Issue 15, July 2009, pp. 2618–2628.

[32] J. Ma, H. Abie, T. Skramstad, and M. Nygaard, Requirements for Evidential Value for the Assessment of the Trustworthiness of Digital Records over Time. In the Proc. of the MASS 2009, IEEE Symp on Trust, Security and Privacy for Pervasive Applications (TSP 2009), Macau SAR, P.R.China, October 12-14, 2009, pp. 796–803.

[33] A. Boukerche and Y. Ren, A Trust-based Security System for Ubiquitous and Pervasive Computing Environments. Computer Communications, Vol. 31, Issue 18, December 2008, pp. 4343–4351.

[34] T. Goovaerts, B. D. Win, and W. Joosen, A Comparison of Two Approaches for Achieving Flexible and Adaptive Security Middleware. In Proc. of the 2008 workshop on Middleware Security, Leuven, Belgium, December 2, 2008. ACM 2008, pp.19–24.

[35] A. Elkhodary and J. Whittle, A Survey of Approaches to Adaptive Application Security. International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '07), 20-26 May 2007.

[36] S. Sadjadi, A Survey of Adaptive Middleware. Technical Report MSU-CSE-03-35, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, December 2003.

[37] P. McKinley, S. Sadjadi, E. Kasten, and B. Cheng, A Taxonomy of Compositional Adaptation. Technical Report, MSU-CSE-04-17, 2004.

[38] L. Marcus, Semantics of Static, Adaptive, and Incremental Security Policies. First Symposium on Requirements Engineering for Information Security (SREIS) March 2001, Indianapolis, Technical Report ATM 2001(8104-05)-1, The Aerospace Corporation July 2001.

[39] R. W. McGraw, Risk-Adaptable Access Control (RAdAC). September 2009, accessed May 26th, 2010, http://csrc.nist.gov/news_events/privilege-management-workshop/radac-Paper0001.pdf.

[40] G. Qu and S. Hariri, Anomaly-Based Self-Protection against Network Attacks. In Autonomic Computing: Concepts, Infrastructure, and Applications, Ed.: M. Parashar and S. Hariri, CRC Press, 2007, pp. 493–521.

[41] T. Ryutov, L. Zhou, C. Neuman, T. Leithead, and K. Seamons, Adaptive Trust Negotiation and Access Control. In Proc. 10th ACM symposium on Access control models and technologies, 1-3 June, 2005, pp. 139–146

[42] H. Shrobe, J. Doyle, and P. Szolovits, Active Trust Management for Autonomous Adaptive Survivable Systems. Self-adaptive Software, 2000, pp. 40–49

[43] J. Weise, Security Architecture and Adaptive Security. SSA (Information Systems Security Association) Journal, July, 2008, pp. 10–15.

[44] R. M. Venkatesan and S. Bhattacharya, Threat-Adaptive Security Policy. In Proc. IEEE International Performance,Computing, and Communications Conference, 5-7 February, 1997, pp. 525–531

[45] P. Lamanna, Adaptive Security Policies Enforced by Software dynamic Translation. A Thesis in TCC 402 25 March, 2002.

[46] C. Wang and W. A. Wulf, Towards a Framework for Security Measurement. In the Proc. of the 20th National Information Systems Security Conference, Baltimore, MD, Oct. 1997, pp. 522–533.

[47] T. Heyman, R. Scandariato, C. Huygens, and W. Joosen, Using Security Patterns to Combine Security Metrics. In the Proc. of the 3rd Int. Conf. on Availability, Reliability and Security (ARES ‘08), pp. 1156–1163.

[48] R. Savola and H. Abie, Development of Measurable Security for a Distributed Messaging System. In: International Journal on Advances in Security, Vol. 2, No. 4, 2009, ISSN 1942-2636, pp. 358–380 (Published in March 2010).

[49] M. Schiffman, G. Eschelbeck, D. Ahmad, A. Wright, and S. Romanosky, CVSS: A Common Vulnerability Scoring System. National Infrastructure Advisory Council (NIAC), 2004.

[50] M. Barrett, C. Johnson, P. Mell, S. Quinn, and K. Scarfone, Guide to adopting and using the Security Content Automation Protocol (SCAP). NIST Special Publ. 800-117 (Draft), U.S. National Institute of Standards and Technology, 2009.

[51] M. Howard, J. Pincus, and J. M. Wing, Measuring Relative Attack Surfaces. Workshop on Advanced Developments in Software and Systems Security, 2003.

[52] P. K. Manadhata, D. K. Kaynar, and J. M. Wing, A Formal Model for a System’s Attack Surface. Technical Report CMU-CS-07-144, July 2007.

[53] D. S. Herrmann, Complete Guide to Security and Privacy Metrics – Measuring Regulatory Compliance, Operational Resilience and ROI. Auerbach Publications, 2007, 824 p.

[54] A. Jaquith, Security Metrics: Replacing Fear, Uncertainty and Doubt. Addison-Wesley, 2007.

[55] N. Bartol, B. Bates, K. M. Goertzel, and T. Winograd, Measuring Cyber Security and Information Assurance: a State-of-the-art Report. Information Assurance Technology Analysis Center IATAC, May 2009.

[56] R. Savola, A Novel Security Metrics Taxonomy for R&D Organisations. In the Proc. of the 7th Annual Information Security South Africa (ISSA ’08) Conference, Johannesburg, South Africa, July 7-9, 2008, pp. 379–390.

[57] Middleware Resource Center. Basic Message-oriented Middleware, Commercial and Open Source., Accessed May 26th, 2010 from http://www.middleware.org/mom/basicmom.html.

[58] S. A. Macskassy and F. Provost, A Brief Survey of Machine Learning Methods for Classification in Networked Data and an Application to Suspicion Scoring, Statistical Network Analysis: Models, Issues, and New Directions. LNCS Springer Berlin / Heidelberg, Vol. 4503/2007, April 12, 2008, pp. 172–175.

[59] H. Abie, Adaptive Security and Trust Management for Autonomic Message-Oriented Middleware. IEEE Symposium on Trust, Security and Privacy for Pervasive Applications (TSP) 2009, October 12-14, 2009 in Macau SAR, P.R.China.

[60] R. Savola and H. Abie, Identification of Basic Measurable Security Components for a Distributed Messaging System. The 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE) 2009, June 18-23, 2009 - Athens/Vouliagmeni, Greece, IEEE Computer Society, pp. 121–128.

[61] E. Grishikashvili, Investigation into Self-Adaptive Software Agents Development. Distributed Multimedia Systems Engineering Research Group, Technical Report, Liverpool john Moors University, 27 April 2001.

[62] ITU Recommendation X.509 (08/05), Information technology – Open systems interconnection – the directory: public-key and attribute certificate frameworks, International Telecommunication Union, 2005.

[63] R. Savola and H. Abie, On-Line and Off-Line Security Measurement Framework for Mobile Ad Hoc Networks. Journal of Networks . Vol. 4, No. 7, September 2009, Academy Publisher, ISSN 1796-2056, pp. 565–579.

[64] R. Savola and H. Abie, Development of Security Metrics for a Distributed Messaging System, The 3rd Int. Conference on Application of Information and Communication Technologies, AICT2009, Azerbaijan, Baku, 14-16 October, 2009.

[65] R. Savola and P. Heinonen, Security-Measurability-Enhancing Mechanisms for a Distributed Adaptive Security Monitoring System, SECURWARE ’10, Venice, Italy, 18-25 July, 2010.

[66] IBM White Paper Autonomic Computing, an Architectural Blueprint for Autonomic Computing, Third Edition, June 2005.

[67] J. Mirkovic and P. Reiher, A Taxonomy of DDOS Attacks and DDOS Defense Mechanisms. Computer Communication Review 2004, Vol. 34; No. 2, ACM Press, USA, ISSN 0146-4833, pp. 39–54.

[68] T. Bu, S. Norden, and T. Woo, A Survivable DoS-resistant Overlay network. Computer Networks 50(9): 1281-1301 (2006)

[69] J. Wang and J. Bigham, Anomaly Detection in the Case of Message-oriented Middleware. In Proceedings of the 2008 Workshop on Middleware Security (Leuven, Belgium, December 02 - 02, 2008). MidSec '08. ACM, New York, NY, 40-42.

[70] J. DeMott, The Evolving Art of Fuzzing, 1 June, 2006, accessed May 26th, 2010, from http://www.vdalabs.com/tools/The_Evolving_Art_of_Fuzzing.pdf.

[71] J. Wang, J. Peng, J. Bigham, B. Chew, B. V. Murciano, M. Novkovic, and I. Dattani, Adding Resilience to Message-oriented Middleware. In Proc. Serene 2010 ACM, 13-16 April, 2010 London, UK.

[72] Sun Microsystems, Java Message Service API Specification v1.1. Sun Microsystems, April, 2002, accessed May 26th, 2010 from http://java.sun.com/products/jms/.

[73] TIBCO. Rendezvous. Accessed May 26th, 2010, from http://www.tibco.com/software/messaging/rendezvous/default.jsp.

[74] IBM. IBM WebSphere MQ support, accessed May 26th, 2010 from http://www-01.ibm.com/software/integration/wmq/support/.

Information Geometrical Approximation of Quantum Channel Security

Laszlo Gyongyosi
Department of Telecommunications, Budapest
University of Technology
Magyar tudosok krt. 2., Budapest, H-1117, Hungary
gyongyosi@hit.bme.hu

Sandor Imre
Department of Telecommunications, Budapest
University of Technology
Magyar tudosok krt. 2., Budapest, H-1117, Hungary
imre@hit.bme.hu

Abstract— **The problem of quantum cloning is closely connected to quantum cryptography. While an eavesdropper on a quantum channel cannot copy perfectly the sent quantum states, in many cases a cloning machine is known to be the most powerful eavesdropping strategy with which to counter quantum cryptographic protocols. In this paper, effective computational geometrical methods are used to analyze cloning activity on a quantum channel. A geometric approach is demonstrated which analyzes the security of the quantum channel, based on quantum relative entropy and Delaunay triangulation on the Bloch sphere. In the security analysis, an approximation algorithm derived from classical computational geometry is used to determine the smallest enclosing ball of balls using core-sets. An improved version is presented which is able to obtain a more effective approximation algorithm in quantum space, while the performances of the proposed geometric algorithms are compared.**

Keywords - quantum cryptography; quantum cloning; quantum informational distance

I. INTRODUCTION

In the past few years, quantum key distribution schemes have invited much study [1]. Quantum theory takes advantage of quantum mechanical principles such as the superposition of states and their no-cloning principle, introduced by Wooters and Zurek [2]. According to the no-cloning theorem, an unknown quantum state cannot be copied perfectly. The method of imperfect cloning was first published by Buzek and Hillery in 1996 [3], while Mozyrsky and Privman have subsequently analyzed other possible imperfect quantum copying approaches and their practical realization [4].

The problem of quantum cloning is closely connected to quantum cryptography, a cryptographic method based on the principles of quantum theory. Using current network technology, interfaces able to manage both quantum and classical channels simultaneously must be implemented in order to spread quantum cryptography.

In our fundamentally new security analysis of quantum cryptography, the fidelity of the eavesdropper's cloning machine is derived from Laguerre-type Delaunay diagrams on the Bloch sphere. Voronoi diagrams are the geometric dual of ordinary Delaunay diagrams, and their application in quantum space has been studied by Kato et al. [5]. In computational geometry, many complex high dimensional problems can be expressed with graphs and tessellation diagrams. Using Laguerre diagrams [6][7], the radii of the smallest enclosing balls of mixed states on the Bloch sphere can be calculated efficiently, while the level of eavesdropping activity can also be measured. The geometric interpretation of quantum states investigates informational distances between two different quantum states [8][9]. The fidelity of the quantum cloning transformation is here computed using a classical informational geometric algorithm presented by Badoui and Clarkson [10], and the Laguerre Delaunay triangulation on the Bloch sphere. The geometric interpretation of the smallest ball problem in informational space has been investigated previously by Nielsen and Nock [11].

As classical cryptographic methods used in wired and wireless security have been found to have vulnerabilities, new methods based on quantum mechanical principles have been deployed [12]. Quantum cryptography is an emerging technology that may offer new forms of security protection. However, quantum cloning-based attacks against quantum cryptography protocols will play a crucial role in the future [13][14][15][16].

Our goal is to identify these imperfect quantum cloning-based attacks, and find potential and efficient solutions for their detection in secret quantum communications. Analysis of the fidelity of the eavesdropper's cloning machine indicates how far the eavesdropper preserves the size of the space of quantum states. In the proposed method, quantum informational distance plays an important role in the estimation of this fidelity. Also presented is a fundamentally new method of deriving quantum relative entropy based on Delaunay tessellation on the Bloch ball and the computation of the radius of smallest enclosing quantum informational ball, to detect eavesdropping activity on the quantum channel.

This paper is organized as follows. In Section II, basic facts about computational geometry and quantum information theory are discussed. In Section III, the basic properties of quantum cloners are explained, while in Sections IV and V, the application of the method for the security analysis of eavesdropping detection on the quantum channel is shown. In Section VI, the optimized algorithm is presented, and the performances of the proposed methods compared. In Section VII, an illustrative example is provided which presents the main steps of the proposed quantum informational geometric security analysis. Finally, in Section VIII a conclusion is drawn from the results.

II. Attacker Model and Geometric Background

The map of the quantum cloner compresses the Bloch ball as an affine map. This affine map has to be a completely positive, trace-preserving map which shrinks the Bloch ball along the *x*, *y* and *z* axes. Quantum informational theoretical analysis of the eavesdropper's cloning machine indicates to what extent the size of the space of quantum states is preserved. In the proposed model, due to eavesdropper activity the sent pure quantum states become mixed states. Eve's output is represented by a 2×2 density matrix, her operation being a trace-preserving completely positive (CP) map. Eve's map is denoted by L , which is trace-preserving if $Tr\left(\mathrm{L}\left(\rho\right)\right) = Tr\left(\rho\right)$ for all density matrices ρ , and positive if the eigenvalues of $\mathrm{L}\left(\rho\right)$ are non-negative, whenever the eigenvalues of ρ are non-negative. As Eve's map L has to be CP, $\mathrm{I}_n \otimes \mathrm{L}$ is thus a positive map for all *n*, where I_n is the identity map on $n \times n$ matrices [15].

A computational geometric method is used to analyze cloning activity on a quantum channel, using the Bloch ball representation. The activity of an eavesdropper on a single qubit in the Bloch sphere representation can be expressed by an affine map as

$$\mathbf{r}_E = \mathrm{L}\left(\mathbf{r}\right) = A\mathbf{r} + \vec{b}, \tag{1}$$

where A is a 3×3 real matrix, $\vec{b}$ is a three-dimensional vector, $\mathbf{r}$ is the initial Bloch vector of the sent pure quantum state, and $\mathbf{r}_E$ is the Bloch vector of the cloned state.

A. Related Work

To analyze the informational geometric impacts of cloning activity on the quantum channel, the mathematical results of Aurenhammer and Klein [6], Badoiu and Clarkson [10], Panigrahy [17], Badoiu et al. [18], Kumar et al. [19], Kato et al. [5] and Nielsen and Nock [11] are used. The proposed geometric analysis of this paper is based on the most important studies of the security analysis of QKD protocols. Our approach integrates the results of Cerfand Bourennane [14], D'Ariano and Macchiavello [15], Acín et al. [16], Kraus et al. [20], Hillery et al. [21], Cerf et al. [22], D'Ariano and Macchiavello [23], Gisin et al. [24], Gisin and Popescu [25], Niederberger et al. [26] and Mozyrsky and Privman [4]. The geometric approach of Kato et al. [5] is based on the computation of Voronoi diagrams using a lower envelope technique. However, the current paper shows that an enhanced version of the method presented by Nielsen and Nock [11] can be applied in quantum space, and by the introduction of quantum Delaunay diagrams for convex hull calculation, a more powerful method can be achieved.

In our security analysis, the informational theoretical meaning of quantum cloning activity in the quantum channel is studied. Alice's side is modeled by the random variable $X = \left\{p_i = P\left(x_i\right)\right\}, i = 1, \ldots N.$, while Bob's side can be modeled by another random variable *Y*. The Shannon entropy for the discrete random variable *X* is denoted by $H\left(X\right)$, which can be defined as

$$H\left(X\right) = -\sum_{i=1}^{N} p_i \log\left(p_i\right). \tag{2}$$

For conditional random variables, the probability of the random variable *X* given *Y* is denoted by $p\left(X|Y\right)$. Alice sends a random variable to Bob, who subsequently produces an output signal with a given probability.

The effects of Eve's quantum cloner on Bob's received quantum state are then geometrically analyzed. The presence of Eve's cloner in the quantum channel increases the uncertainty in *X*, given Bob's output of *Y*. The informational theoretical noise of Eve's quantum cloner increases conditional Shannon entropy $H\left(X|Y\right)$, where

$$H\left(X|Y\right) = \sum_{i=1}^{N_X}\sum_{j=1}^{N_Y} p\left(x_i, y_j\right) \log p\left(x_i|y_j\right). \tag{3}$$

The general model for the quantum cloner based attack is illustrated in Figure 1.

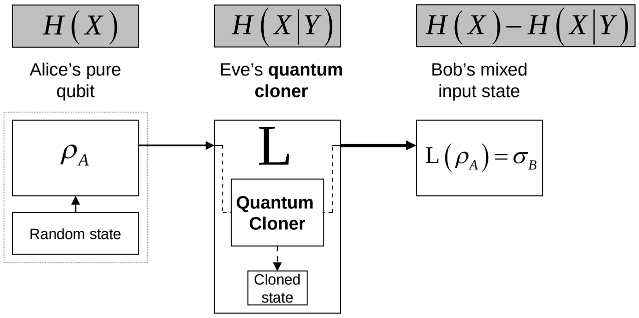


Figure 1. The analyzed attacker model and entropies.

The proposed geometric security analysis focuses on the cloned mixed quantum state, which is received by Bob. The type of quantum cloner machine depends on the actual protocol. For example, in the four-state protocol Eve chooses the phase-covariant cloner, while in the six-state protocol she uses the universal quantum cloner (UCM) machine. Alice's pure state is denoted by ρ_A, Eve's cloner modeled by an affine map L , and Bob's mixed input state denoted by $\mathrm{L}(\rho_A) = \sigma_B$.

Our calculations utilize the fact that for random variables *X* and *Y*, $H(X,Y) = H(X) + H(Y|X)$, where $H(X)$, $H(X,Y)$ and $H(Y|X)$ are defined by probability distributions. Information which can be transmitted in the presence of an eavesdropper on the quantum channel is measured in a geometric representation. In a secret quantum channel, $H(X)$ and $H(X|Y)$ are sought to be maximized and minimized, respectively, in order to maximize the radius r^* of the smallest enclosing ball of Bob, whose radius describes the maximum transmittable information from Alice to Bob in the *attacked* quantum channel:

$$r^* = Max_{\{all\ possible\ x_i\}} H(X) - H(X|Y). \qquad (4)$$

To calculate the radius r^* of the smallest informational ball of quantum states and the entropies between the cloned quantum states, von Neumann entropy and quantum relative entropy are used rather than classical Shannon entropy.

Geometrically, the presence of an eavesdropper causes detectable mapping to change from a noiseless one-to-one relationship to a stochastic map. If there is no cloning activity on the channel, then $H(X|Y) = 0$ and the radius of the smallest enclosing quantum informational ball on Bob's side will be maximized.

B. Properties of Quantum States

In the quantum space, a quantum state can be described by their density matrix $\rho \in \mathbb{C}_{d\times d}$, which is a $d \times d$ matrix, where *d* is the level of the given quantum system. For an *n* qubit system, the dimension is $d = 2^n$. The two-level quantum states can be given by their density matrices in the following way:

$$\rho = \frac{1}{2}\begin{pmatrix} 1+z & x-iy \\ x+iy & 1-z \end{pmatrix},\ x^2 + y^2 + z^2 \le 1, \qquad (5)$$

where *i* denotes the complex imaginary $i^2 = -1$. In quantum cryptography, the encoded pure quantum states are sent through a quantum communication channel. Using the Bloch sphere representation, the quantum state ρ can be given as a three-dimensional point $\rho = (x, y, z)$ in $\mathbb{R}^3$, and it can be represented by spherical coordinates

$$\rho = (r, \theta, \varphi), \qquad (6)$$

where *r* is the radius of the quantum state to the origin, while θ and φ represents the latitude and longitude rotation angles. On the Bloch ball B , the *pure* states are on the boundary of the Bloch ball B , while the *mixed* states are inside the Bloch ball.

C. Measuring Distances between Quantum States

For two *pure* quantum states ρ and σ , the geometrical distance $d(\rho,\sigma)$ is defined as

$$\cos d_{Pure}(\rho,\sigma) = \sqrt{Tr(\rho\sigma)}\,, \qquad (7)$$

respectively. For the $d_{Pure}(\rho,\sigma)$ geometrical distance between two pure quantum states, an obvious condition can be given in the following way:

$$0 \le d_{Pure}(\rho,\sigma) \le \frac{\pi}{2}. \qquad (8)$$

We can define geometrical distances between mixed states and pure states in the following way:

$$d_{P,M}(\rho,\sigma) = \sqrt{1 - Tr\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}}\ , \qquad (9)$$

where the quantum states ρ and σ are arbitrary quantum states. If both ρ and σ are *pure* states, then the distance $d_{P,M}(\rho,\sigma)$ between the quantum states can be given by

$$d_{P,M}(\rho,\sigma) = \sqrt{1 - \mathrm{Tr}(\rho\sigma)}, \qquad (10)$$

as we illustrated it in Figure 2.

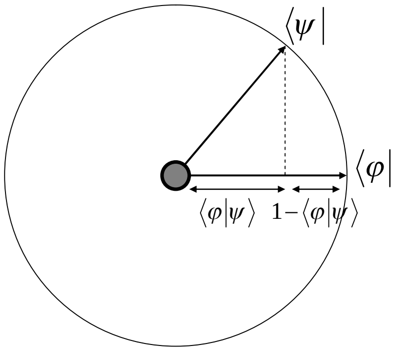


Figure 2. Geometrical representation of distance between pure quantum states.

Hence, for pure quantum states there is a coincidence between distances, since

$$d_{Pure}(\rho,\sigma) = d_{P,M}(\rho,\sigma), \qquad (11)$$

thus, the distances $d_{P,M}\left(\rho,\sigma\right)$ and $d_{Pure}\left(\rho,\sigma\right)$ are the same for pure quantum states.

In our security analysis, the distances between quantum states is defined by the quantum relative entropy. The relative entropy of quantum states measures the informational distance between quantum states.

The classical Shannon-entropy of a discrete *d*-dimensional distribution *p* can be given by

$$H\left(p\right)=\sum_{i=1}^{d}p_i\log\frac{1}{p_i}=-\sum_{i=1}^{d}p_i\log p_i\ . \qquad (12)$$

The von Neumann entropy S of quantum states is a generalization of the classical Shannon entropy to density matrices [8][9]. The entropy of quantum states can be given by:

$$\mathrm{S}\left(\rho\right)=-Tr\left(\rho\log\rho\right). \qquad (13)$$

The quantum entropy $\mathrm{S}\left(\rho\right)$ is equal to the Shannon entropy for the eigenvalue distribution:

$$\mathrm{S}\left(\rho\right)=\mathrm{S}\left(\lambda\right)=-\sum_{i=1}^{d}\lambda_i\log\lambda_i, \qquad (14)$$

where *d* is the level of the quantum system.

The relative entropy in classical systems is a measure, that quantifies how close a probability distribution *p* to a model or candidate probability distribution *q* [8][9]. For *p* and *q* probability distributions the relative entropy can be given by

$$D\left(p\|q\right)=\sum_{i}p_i\log_2\frac{p_i}{q_i}\ . \qquad (15)$$

The relative entropy between quantum states can be measured by

$$D\left(\rho\|\sigma\right)=Tr\left[\rho\left(\log\rho-\log\sigma\right)\right]. \qquad (16)$$

The quantum relative entropy plays a key role in the description of the quantum state space. The quantum informational distance has some distant-like properties, however it is not commutative [8][9], thus $D\left(\rho\|\sigma\right)\neq D\left(\sigma\|\rho\right)$, and $D\left(\rho\|\sigma\right)\geq 0$ iff $\rho\neq\sigma$, and $D\left(\rho\|\sigma\right)=0$ iff $\rho=\sigma$. The quantum information theoretical distance is not symmetric, nor do they satisfy the triangular inequality of metrics.

D. Quantum Informational Distance between Quantum States

The quantum relative entropy reduces to the classical Kullback-Leibler relative entropy for simultaneously diagonal matrices. Using the negative entropy of quantum states, the relative entropy of quantum states can be described by a strictly convex and differentiable generator function $\mathbf{F}$:

$$\mathbf{F}\left(\rho\right)=-\mathrm{S}\left(\rho\right)=Tr\left(\rho\log\rho\right). \qquad (17)$$

The quantum informational distance $D\left(\rho\|\sigma\right)$ for density matrices ρ and σ can be given by generator function $\mathbf{F}$ in the following way:

$$D\left(\rho\|\sigma\right)=\mathbf{F}\left(\rho\right)-\mathbf{F}\left(\sigma\right)-\left\langle\rho-\sigma,\nabla\mathbf{F}\left(\sigma\right)\right\rangle, \quad (18)$$

where $\left\langle\rho,\sigma\right\rangle=Tr\left(\rho\sigma^{*}\right)$ is the inner product of quantum states, and $\nabla\mathbf{F}\left(\cdot\right)$ is the gradient. In Figure 3, we have depicted the geometrical interpretation of quantum informational distance between quantum states ρ and σ [1][5][6]. The point of intersection of quantum state ρ on $H\left(\sigma\right)$ is denoted by $H_{\sigma}\left(\rho\right)$. The tangent hyperplane to hypersurface $\mathbf{F}\left(x\right)$ at quantum state σ is

$$H_{\sigma}=\mathbf{F}\left(\sigma\right)+\left\langle\nabla\mathbf{F}\left(\sigma\right),x-\sigma\right\rangle. \qquad (19)$$

We have depicted the quantum informational distance, $D\left(\rho\|\sigma\right)$, as the vertical distance between the generator function $\mathbf{F}$ and $H\left(\sigma\right)$, the hyperplane tangent to $\mathbf{F}$ at σ .

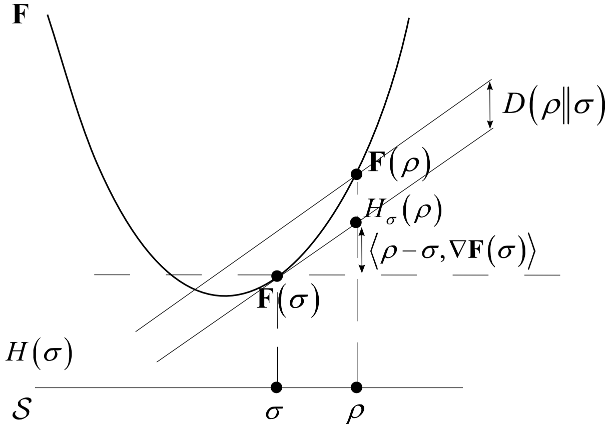

$\mathbf{F}$
$D\left(\rho\|\sigma\right)$
$\mathbf{F}\left(\rho\right)$
$H_{\sigma}\left(\rho\right)$
$\left\langle\rho-\sigma,\nabla\mathbf{F}\left(\sigma\right)\right\rangle$
$\mathbf{F}\left(\sigma\right)$
$H\left(\sigma\right)$
$\mathcal{S}$
σ
ρ

Figure 3. Depiction of generator function as negative von Neumann entropy.

We note, if σ has zero eigenvalues, quantum relative entropy function $D(\rho\|\sigma)$ may diverge, otherwise it is a finite and continuous function.

The quantum relative entropy for general quantum state $\rho=(x,y,z)$ and mixed state $\sigma=(\tilde{x},\tilde{y},\tilde{z})$, with radii $r_\rho=\sqrt{x^2+y^2+z^2}$ and $r_\sigma=\sqrt{\tilde{x}^2+\tilde{y}^2+\tilde{z}^2}$, can be expressed as

$$\begin{aligned} D(\rho\|\sigma) &= \frac{1}{2}\log\frac{1}{4}\left(1-r_\rho{}^2\right)+\frac{1}{2}r_\rho\log\frac{(1+r_\rho)}{(1-r_\rho)} \\ &\quad -\frac{1}{2}\log\frac{1}{4}\left(1-r_\sigma{}^2\right)-\frac{1}{2r_\sigma}\log\frac{(1+r_\sigma)}{(1-r_\sigma)}\langle\rho,\sigma\rangle, \end{aligned} \tag{20}$$

where $\langle\rho,\sigma\rangle=(x\tilde{x}+y\tilde{y}+z\tilde{z})$. For a maximally mixed state $\sigma=(\tilde{x},\tilde{y},\tilde{z})=(0,0,0)$ and $r_\sigma=0$, the quantum divergence can be expressed as

$$D(\rho\|\sigma)=\frac{1}{2}\log\frac{1}{4}\left(1-r_\rho{}^2\right)+\frac{1}{2}r_\rho\log\frac{(1+r_\rho)}{(1-r_\rho)}-\frac{1}{2}\log\frac{1}{4}. \tag{21}$$

Using Bloch vector representation $\rho=\frac{1}{2}\left(\mathbf{1}+\vec{r}_\rho\cdot\vec{\sigma}\right)$ and $\sigma=\frac{1}{2}\left(\mathbf{1}+\vec{r}_\sigma\cdot\vec{\sigma}\right)$ of two mixed states ρ and σ, the quantum relative entropy between them can be given by the following formula:

$$\begin{aligned} D(\rho\|\sigma) &= \frac{1}{2}\log\left(\frac{1-r_\rho^2}{1-r_\sigma^2}\right)+\frac{1}{2}r_\rho\log\left(\frac{1+r_\rho}{1-r_\rho}\right) \\ &\quad -\frac{1}{2}\left(r_\rho\cos\theta\right)\log\left(\frac{1+r_\sigma}{1-r_\sigma}\right), \end{aligned} \tag{22}$$

where r_ρ and r_σ denote the lengths of Bloch vectors, and θ is the angle between the two mixed quantum states. The quantum informational distance between two mixed quantum states depends on the lengths of their Bloch vectors and the angle θ between them, as illustrated in Figure 4.

The density matrices of quantum states are represented by 3D points in the Bloch ball. The eavesdropper's cloner transformation is modeled by an affine map, which maps quantum states to other quantum states. Geometrically, the eavesdropper maps the Bloch ball to a deformed ball. The cloning activity in the channel can then be analyzed by the radius of the deformed Bloch ball, which can be computed by the proposed geometric methods.

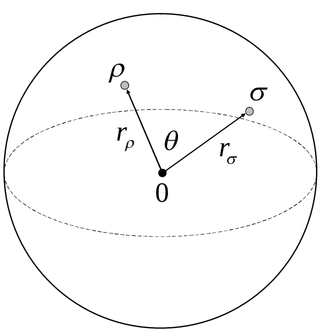


Figure 4. The quantum relative entropy between two mixed quantum states.

In our security analysis a Delaunay tessellation is used, whose diagram is symmetrical only for pure states, and asymmetrical for quantum states if the lengths of the radii differ. To construct a Delaunay triangulation, three-dimensional Laguerre diagrams are used.

E. The Smallest Enclosing Quantum-Information Ball

We would like to compute the radius r of the smallest enclosing ball, thus first we have to seek the center $\mathbf{c}^*$ of the point set S. The set S of quantum states is denoted by $\mathrm{S}=\{\rho_i\}_{i=1}^n$. The distance function $d(\cdot,\cdot)$ between any two quantum states of S is measured by quantum relative entropy, thus the minimax mathematical optimization is applied to quantum informational distance to find the center $\mathbf{c}$ of the set S of quantum states. We denote the quantum informational distance from $\mathbf{c}$ to the furthest point of S by distance function $d(\cdot,\cdot)$ as

$$d(\mathbf{c},\mathrm{S})=\max_i d(\mathbf{c},s_i). \tag{23}$$

Using the minimax optimization, we can minimize the maximal quantum relative entropy from $\mathbf{c}$ to the furthest point of S by

$$\mathbf{c}^*=\arg\min_{\mathbf{c}} d(\mathbf{c},\mathrm{S}) \tag{24}$$

In Figure 5, we illustrated the circumcenter $\mathbf{c}^*$ of S, and the smallest enclosing balls for the Euclidean distance and quantum relative entropy [11].

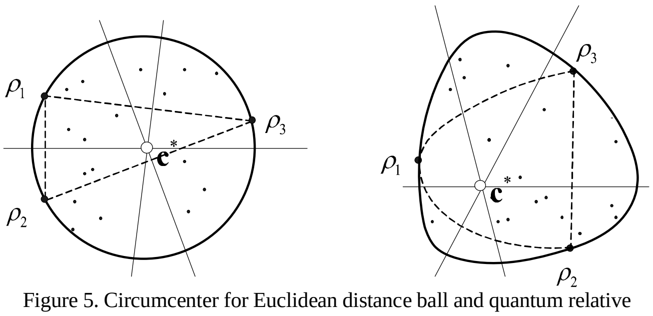


Figure 5. Circumcenter for Euclidean distance ball and quantum relative entropy ball.

In Figure 6, we compare the smallest quantum informational ball and the ordinary Euclidean ball [11]. We can conclude that the quantum states ρ_1, ρ_2 and ρ_3, which determine the Euclidean smallest enclosing ball are different from the states of the quantum informational ball.

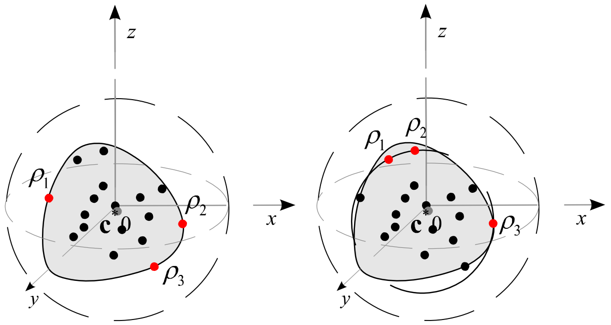


Figure 6. The maximum distance states of the smallest balls differ for the quantum informational distance and Euclidean distance.

The informational theoretical effect of the eavesdropper's cloning machine is described by the *radius* of the smallest enclosing quantum informational ball, denoted by r^*. This quantum informational theoretical radius is equal to the maximum quantum informational distance from the center, and can be expressed as:

$$r^* = \min_{\sigma \in \mathrm{S}(\square^2)} \max_{\rho \in \mathrm{S}(\square^2)} D\left(\mathrm{L}(\rho) \| \mathrm{L}(\sigma)\right). \tag{25}$$

In the proposed procedure of calculating the smallest enclosing information ball, a Delaunay tessellation is used because it is the *fastest* known tool with which to seek the center of a smallest enclosing ball of points.

III. Quantum Cloning in Quantum Cryptography

The quantum cloning machine does a trace-preserving, completely positive map $\{U, |\mathrm{M}\rangle\}$. When apriori information about the input states is given so that input quantum states can be particularly characterized, we call it state-dependent quantum cloning. By the outputs, there are symmetric and asymmetric cloning machines. A cloning machine is called symmetric if at the output all the clones have the same fidelity, and asymmetric if the clones have different fidelities.

The quantum channel's error rate is denoted by D_C^{attack}, and the maximal information theoretic fidelity of the eavesdropper's cloning machine is denoted by F_{Eve}. The parameter F_{Eve} represents the theoretical upper bound on the cloning machine's fidelity. The cloning machine based attack has a critical value for the error rate

$$D_C^{attack} = 1 - F_{Eve}, \tag{26}$$

thus Eve's cloning machine fidelity is

$$F_{Eve} = 1 - D_C^{attack}. \tag{27}$$

For example, if Eve uses universal quantum cloner, then the value of parameter F_{Eve} is independent of input quantum state $|\psi\rangle$, and the fidelity of her optimal quantum cloning machine can be defined by

$$F_{Eve} = \langle\psi|^{(in)} \rho^{(out)} |\psi\rangle^{(in)} = \frac{1}{2}(1+\eta), \tag{28}$$

where η is the reduction factor. The quantum cloning transformation is optimal [14][15], if $\eta = \frac{2}{3}$, hence the maximal fidelity of optimal universal cloning is $F_{Eve} = \frac{5}{6}$, and the maximal radius $r_{Eve}^{universal}$ is

$$r_{Eve}^{universal} = \frac{2}{3}. \tag{29}$$

The quantum informational theoretical radius can be defined as

$$r_{Eve}^{*universal} = 1 - \mathrm{S}\left(r_{Eve}^{universal}\right), \tag{30}$$

where S is the von Neumann entropy of corresponding quantum state with radius length $r_{Eve}^{universal}$.

In general, the universal cloning machine [14][15][16] output state is can be given as

$$\rho^{(out)} = F_{Eve} |\psi\rangle_a \langle\psi| + (1 - F_{Eve}) |\psi_\perp\rangle_a \langle\psi_\perp|. \tag{31}$$

Asymmetric cloning has direct application to eavesdropping strategies in quantum cryptography. The best-known example of state-dependent quantum cloning machine is the phase-covariant cloning machine. Here, the states lie in the equator $(x-y)$ of the Bloch sphere, thus the fidelity of the cloning will be independent of φ.

A. *Optimal Individual Eavesdropping Strategy for QKD*

The phase-covariant cloning machine has a remarkable application in quantum cryptography, namely its use in optimal individual eavesdropping strategies in the BB84 protocol [20][22][23]. Using an individual-type cloning-based attack, Eve applies the same unitary transformation to each sent quantum state. She does not introduce correlation among the copies, measuring her state after she clones it [9]. Alice, Bob and Eve immediately measure their respective quantum states, since the parties have no ability to store qubits.

The results of their measurements can be described by a probability distribution $P(A,B,E)$. Eve's quantum state is denoted by $|E\rangle$, while the unitary operation, which describes the interaction between the sent qubit and Eve's state, is denoted by L, and thus the whole transformation can be expressed as [24][26]:

$$\begin{aligned}|E\rangle\otimes|0\rangle &\xrightarrow{\mathrm{L}} |E_{0,0}\rangle|0\rangle+|E_{0,1}\rangle|1\rangle,\\ |E\rangle\otimes|1\rangle &\xrightarrow{\mathrm{L}} |E_{1,0}\rangle|0\rangle+|E_{1,1}\rangle|1\rangle,\end{aligned} \tag{32}$$

where $|E_{i,j}\rangle$ denotes Eve's cloned quantum state. $|E\rangle$ can be written as a 2×2 matrix, whose elements are Eve's states $|E_{i,j}\rangle$ [9].

Eve's cloning activity can be written in the following form:

$$\begin{aligned}|E\rangle\otimes|0\rangle &\xrightarrow{\mathrm{L}} \sqrt{F}\,|E_{0,0}\rangle|0\rangle+\sqrt{D}\,|E_{0,1}\rangle|1\rangle,\\ |E\rangle\otimes|1\rangle &\xrightarrow{\mathrm{L}} \sqrt{D}\,|E_{1,0}\rangle|0\rangle+\sqrt{F}\,|E_{1,1}\rangle|1\rangle,\end{aligned} \tag{33}$$

where $|E_{i,j}\rangle$ represents Eve's normalized state in case Alice sent an *i*-bit and Bob detected a *j*-bit. F is the fidelity parameter, while D is the disturbance. The fidelity F provides the probability that Bob detected Alice's bit correctly using the same basis, while D provides the probability of an incorrect detection [9].

B. Different Types of Quantum Cloners

In quantum cryptography the optimal eavesdropping attack is done by a phase-covariant cloning machine, which clones the *x* equator. The importance of equatorial qubits lies in the fact, that quantum cryptography requires these states, rather than the states that span the whole Bloch sphere [9].

The transformations were restricted for pure input states of the form

$$|\psi_\phi\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+e^{i\phi}|1\rangle\right), \tag{34}$$

where the parameter $\phi\in[0,2\pi)$ represents the angle between the Bloch vector and the *x*-axis. These qubits are called equatorial qubits, because the *z*-component of their Bloch vector is zero. The phase-covariant quantum cloners [14][15][16] can clone arbitrary equatorial qubits, and they keep the quality of the copies same for all equatorial qubits. The reduced density operator of the copies at the output can be expressed as [9]

$$\rho^{(out)}=\left(\frac{1}{2}+\sqrt{\frac{1}{8}}\right)|\psi_\phi\rangle\langle\psi_\phi|+\left(\frac{1}{2}-\sqrt{\frac{1}{8}}\right)|\psi_{\phi,\perp}\rangle\langle\psi_{\phi,\perp}|, \tag{35}$$

where $|\psi_{\phi,\perp}\rangle$ is orthogonal to state $|\psi_\phi\rangle$. Thereby, the optimal fidelity of 1 to 2 phase-covariant cloning transformation is given by

$$F_{1\to2}^{phase}=\frac{1}{2}+\sqrt{\frac{1}{8}}\approx0.8535. \tag{36}$$

If Eve has a phase-covariant quantum cloner, then the maximal value of her *radius* r_{Eve}^{phase} is

$$r_{Eve}^{phase}=2\sqrt{\frac{1}{8}}. \tag{37}$$

The quantum informational radius r_{Eve}^{*phase} of the phase-covariant cloner can be defined as

$$r_{Eve}^{*phase}=1-\mathrm{S}\left(r_{Eve}^{phase}\right), \tag{38}$$

where S is the von Neumann entropy of corresponding quantum state with radius length r_{Eve}^{phase}.

The phase-covariant quantum cloning transformation produces two copies of the equatorial qubit, with optimal fidelity. The phase-covariant cloning transformation without ancilla is a two-qubit unitary transformation, it can be given by $|0\rangle|0\rangle\to|0\rangle|0\rangle$ and $|1\rangle|0\rangle\to\cos\eta|1\rangle|0\rangle+\sin\eta|0\rangle|1\rangle$, where $\eta\in[0,\pi/2]$ is the shrinking parameter, which is related to the fidelity [14][15][16][23].

IV. Computation of Quantum Channel Security

In this section the informational geometric model of the proposed security analysis is defined, fundamentally based on the results of Gyongyosi and Imre [1]. In the computation of the smallest enclosing quantum informational ball, the results of Kato et al. [5] and Nielsen and Nock [11] are also used.

In the proposed model, the fidelity of the eavesdropper's cloning machine is derived from the quantum informational theoretical radius r^* of the smallest enclosing quantum informational ball and the theoretical upper bound of the quantum informational radius of the eavesdropper's cloning machine, denoted by r_{Eve}^*.

Theorem

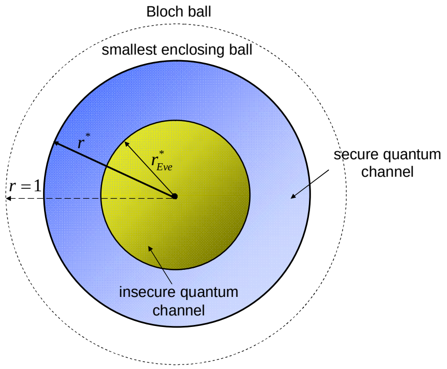
For a secure quantum channel, the radius r^* of the smallest enclosing quantum informational ball of mixed states must be greater than r_{Eve}^*, thus

$$r^* > r^*_{Eve} , \tag{39}$$

where r^* is the radius of the smallest enclosing informational ball, computed by the proposed method. In terms of the second part, for an insecure quantum channel the radius r^* is smaller than or equal to r^*_{Eve} and thus

$$r^* \le r^*_{Eve} . \tag{40}$$

Figure 7 shows a geometric interpretation of the proposed model for a secure quantum channel.

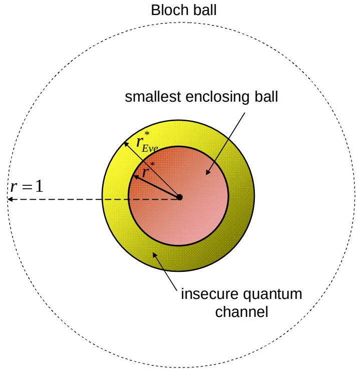


$r^* > r^*_{Eve}$

Figure 7. The radius of the smallest enclosing informational ball for secure quantum communication.

In our security analysis, a spherical Delaunay tessellation is used to compute the quantum informational theoretical radius r^*, since it can be simply obtained as an ordinary Euclidean Delaunay triangulation mesh. The quantum relative entropy-based Delaunay tessellation of pure states is identical to conventional spherical Delaunay tessellation [1][6].

Figure 8 shows a geometric interpretation of the proposed model for an insecure quantum channel [1].



$r^* \le r^*_{Eve}$

Figure 8. The radius of the smallest enclosing informational ball for insecure quantum communication.

In this scheme, the radii r_{UCM} and $r_{phasecov}$ of corresponding quantum cloners are derived from the smallest enclosing quantum informational balls.

The informational theoretical radius r^*_{UCM} of a universal quantum cloner can be expressed as

$$r^*_{UCM} = 1 - \mathrm{S}\left(r_{UCM}\right) . \tag{41}$$

For a phase-covariant cloner, the informational theoretical radius $r^*_{phasecov}$ can be defined as

$$r^*_{phasecov} = 1 - \mathrm{S}\left(r_{phasecov}\right) , \tag{42}$$

where S is the von Neumann entropy of the corresponding quantum state. In our numerical calculations, the notations r_{UCM} and $r_{phasecov}$ are used for the maximal radii.

The values of r_{UCM} and $r_{phasecov}$ can be computed from the informational theoretical radii of the smallest enclosing quantum informational balls, denoted by r^*_{UCM} and $r^*_{phasecov}$.

Theorem

For UCM and phase-covariant cloner machines, the connection between informational theoretical radii r^*, r^*_{Eve} and the Bloch vectors r and r_{Eve} can be defined as:

$$r^* = 1 - \mathrm{S}\left(r\right), \tag{43}$$

and

$$r^*_{Eve} = 1 - \mathrm{S}\left(r_{Eve}\right), \tag{44}$$

where S is the von Neumann entropy, and r and r_{Eve} are the Euclidean lengths of the vectors from the Bloch sphere origin.

The smallest quantum informational ball with radius r^* is shown in gray, with the ball of the ideal UCM cloner with radius r^*_{UCM} shown in light gray.

It can be concluded that if $r_{\mathrm{E}} \ge r_{\mathrm{E},UCM}$, then $r^* \le r^*_{UCM}$, and hence the informational theoretical radius will be smaller.

Figure 9 compares ideal and imperfect universal and phase-covariant cloner quantum balls.

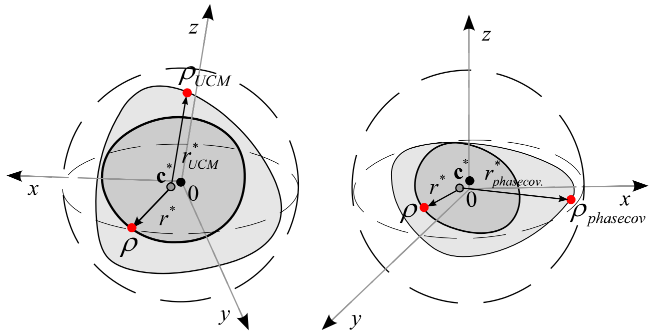


Figure 9. Smallest enclosing quantum informational balls of optimal and imperfect universal and phase-covariant cloners.

It can be concluded that the informational theoretical radii for ideal and imperfect phase-covariant cloning are different.

Figure 10 illustrates the radii r^*_{UCM} and $r^*_{phasecov}$ of the smallest enclosing quantum informational ball for UCM-based and phase-covariant cloner based attacks, in Bloch sphere representation.

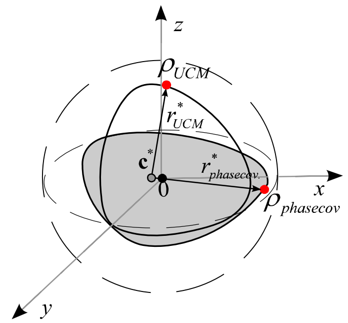


Figure 10. Comparison of smallest enclosing quantum informational ball of UCM-based and phase-covariant cloners.

Geometrically, the smallest quantum informational ball can be computed from the intersection of the contours of the quantum relative entropy ball with the ellipsoid of the secret channel, the latter being generated by the eavesdropper's cloner machine.

A. Geometrical Calculation of Channel Security

In the Bloch sphere representation, the smallest value of $D(\rho\|\sigma)$ corresponds to the contour closest to the location of density matrix.

In Figure 11, the smallest quantum informational ball with radius $r^* = D_{\max}\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$, intersects the channel ellipsoid at the magnitude m_ρ of Bloch vector $\mathbf{r}_\rho$. The Euclidean distance between the origin and center $\mathbf{c}^*$ is denoted by m_σ. Similarly, the Euclidean distance between the origin and state ρ is denoted by m_ρ, respectively.

We note, that for a perfect UCM cloner and a perfect phase-covariant cloner, the center of the channel ellipsoid is equal to the Bloch sphere origin. In our next example, we will use an imperfect quantum cloner model, where the center slightly differs from the Bloch sphere origin.

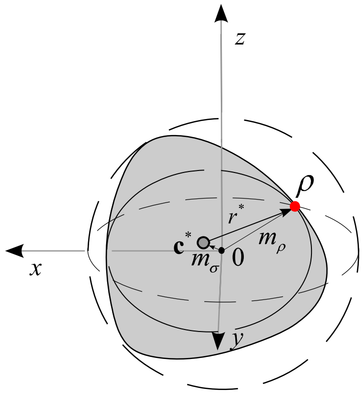


Figure 11. Intersection of radius of smallest enclosing quantum informational ball and channel ellipsoid.

In our geometrical iteration algorithm, we would like to determine the location of vector $\mathbf{r}_\sigma$ inside the channel ellipsoid such that, the largest possible contour value $D_{\max}\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$ touches the channel ellipsoid surface, and the remainder of the $D_{\max}$ contour surface lies entirely outside the channel ellipsoid. The point on the channel ellipsoid surface is defined as the set of channel output ρ. The vector $\mathbf{r}_\sigma$ is defined in the interior of the ellipsoid, as the convex hull of the channel ellipsoid.

To determine the optimal length of the radius, the algorithm moves point σ. As we move vector $\mathbf{r}_\sigma$ from the optimum position, a larger contour corresponding to the larger value of the quantum relative entropy D will intersect the channel ellipsoid surface, thereby $\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$ will increase. We can conclude that vector $\mathbf{r}_\sigma$ should be adjusted to minimize $\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$, as we illustrated it in Figure 12.

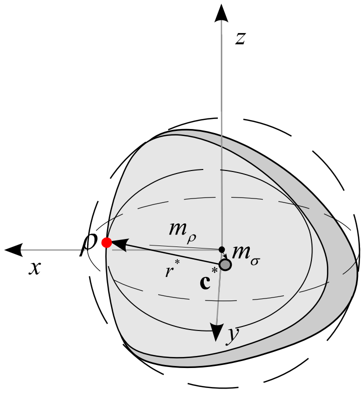


Figure 12. Moving of the vector from the optimum position will increase quantum relative entropy. The optimal ball is shown in light-grey.

The maximum length radius $\mathbf{r}_\rho$ can be determined by an iterative algorithm, suing the quantum relative entropy as distance measure. The computed radius is equal to the radius of the smallest quantum informational ball, hence the quantum informational radius can be used to derive the fidelity of the eavesdropper's quantum cloner machine.

B. *Geometrical Representation of the Iteration Process*

The vector $\mathbf{r}_\sigma$ should be adjusted to minimize the value of $\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$. To find the optimal value of vector $\mathbf{r}_\sigma$ in our geometrical analysis, we choose a start point for vector $\mathbf{r}_\sigma$ in the interior of the ellipsoid.

In Figure 13(a), we show the initial start point inside the channel ellipsoid chosen by the algorithm. The vector of state σ is denoted by $\mathbf{r}_\sigma$. In the next step, the algorithm determines the set of points to the vector $\mathbf{r}'_\rho$ on the ellipsoid surface most distant from $\mathbf{r}_\sigma$, using the quantum relative entropy as distance measure. In Figure 13(b), the new state is notated by ρ'.

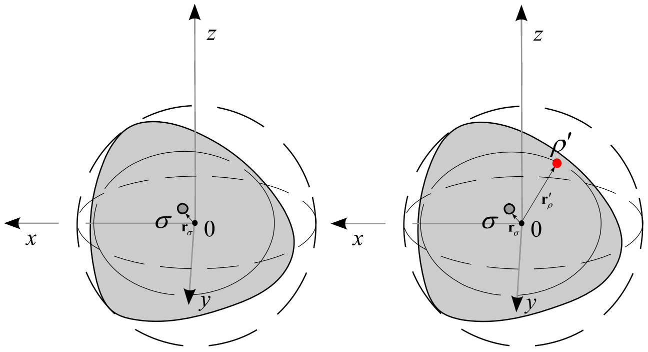


Figure 13. The algorithm determines the points on the ellipsoid surface most distant from the point, using the quantum relative entropy.

The maximum distance between the states can be expressed as

$$\max_{\mathbf{r}_\rho} D\left(\mathbf{r}'_\rho \| \mathbf{r}_\sigma\right). \tag{45}$$

We choose a random Bloch sphere vector from the maximal set of points according to vector $\mathbf{r}'_\rho$. The selected point is denoted by $\mathbf{r}''_\rho$. The algorithm makes a step from $\mathbf{r}_\sigma$ towards the surface point vector $\mathbf{r}''_\rho$ in the Bloch sphere space. In this step, the algorithm updates vector $\mathbf{r}_\sigma$ to

$$\mathbf{r}^*_\sigma = (1-\gamma)\mathbf{r}_\sigma + \gamma \mathbf{r}''_\rho, \tag{46}$$

where γ denotes the size of the step.

In Figure 14(a), the updated state and the vector of the state are denoted by ρ'' and $\mathbf{r}''_\rho$. The center of the quantum informational ball is denoted by $\mathbf{r}^*_\sigma$.

In Figure 14(b), we illustrate the quantum informational distance between the final center point and the maximal distance state ρ, using the notation $\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$.

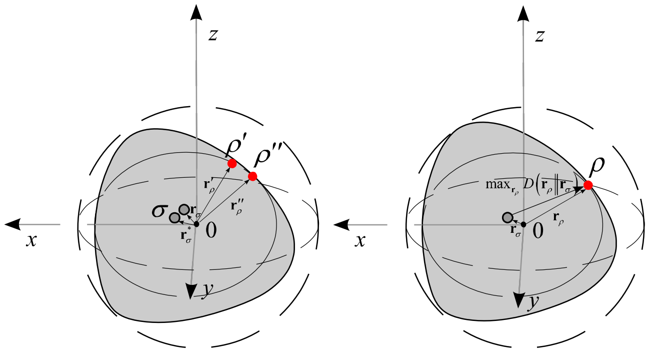


Figure 14. The algorithm makes a step towards the found surface point vector and updates the vector.

Using the updated vector $\mathbf{r}^*_\sigma$, the algorithm continues to loop until $\max_{\mathbf{r}^*_\rho} D\left(\mathbf{r}'_\rho \| \mathbf{r}^*_\sigma\right)$ no longer changes. We conclude that the iteration converges to the optimal $\mathbf{r}_\sigma$, because the algorithm minimizes $\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right)$. At the end of the iteration process, the radius of the smallest quantum informational ball can be expressed as

$$\min\max_{\mathbf{r}_\rho} D\left(\mathbf{r}_\rho \| \mathbf{r}_\sigma\right). \tag{47}$$

We would like to compute the radius r^* of the smallest enclosing ball, thus first we have to seek the center $\mathbf{c}^*$ of the point set S.

C. *Delaunay Triangulation on the Bloch Sphere*

The mesh of the Bloch sphere B can be described as a number of points connected in some way by lines, the points and the lines of the mesh are referred to as *edges* and *vertices*.

The triangle t is said to be Delaunay, when its circumcircle is empty. For an empty circumcircle, the circle passing through the quantum states of a triangle $t \in T$, encloses no other vertex of the set S [6][7].

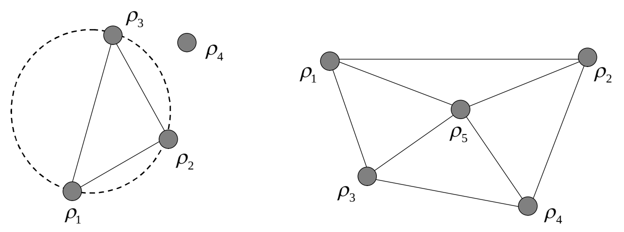


Figure 15. The Delaunay triangulation of a set of quantum states.

The Delaunay triangulation $Del(\mathrm{S})$ of a set of quantum states S is unique, if at most three quantum states $\rho \in \mathrm{S}$ are co-circular. The Delaunay triangulation $Del(\mathrm{S})$ of a set of quantum states $\mathrm{S} = \{\rho_1, \rho_2, \ldots \rho_n\}$ maximizes the minimum angle among all triangulation of a given set of quantum states, and the circle centered at vertex $\mathbf{c}$, gives an empty circumcenter for quantum states $\rho_1\rho_2\rho_3$, as we illustrated it in Figure 16.

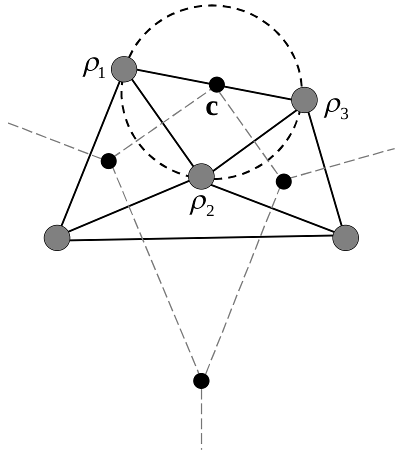


Figure 16. The triangle of quantum states corresponds to the vertex c.

In our paper we use the Laguerre Delaunay diagrams [6][7] to compute the radius of the smallest enclosing ball. In general, the Laguerre distance for generating point x_i with weight r_i^2 , is defined by

$$d_L(\rho, x_i) = \|\rho - x_i\|^2 - r_i^2 . \qquad (48)$$

The Delaunay diagram for the Laguerre distance is called the Laguerre Delaunay diagram. We illustrated the dual diagram of the Laguerre Delaunay tessellation in Figure 17.

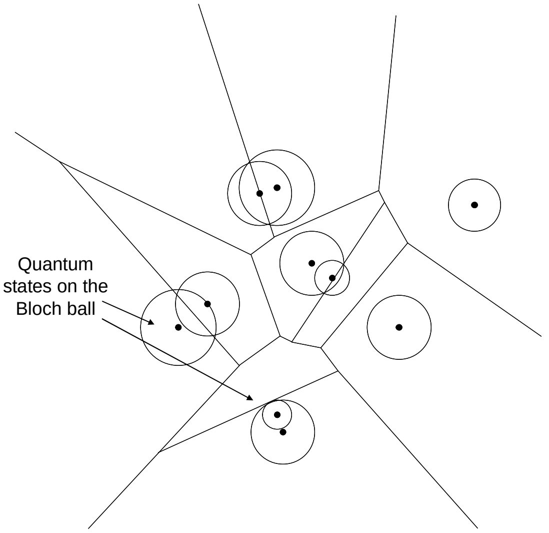


Figure 17. Laguerre diagram for quantum states on the Bloch ball.

We show a fundamentally new method for deriving the quantum relative entropy based Delaunay tessellation on the Bloch ball B , to analyze the informational theoretical impacts of eavesdropping activity on the quantum channel.

V. Geometrical Computation of Smallest Quantum Informational Ball

In our algorithm we present an effective solution to seek the center $\mathbf{c}$ of the set of smallest enclosing quantum information ball, using Laguerre diagrams. Our geometrical-based security analysis has two main steps:

1. *We construct quantum Delaunay triangulation from Laguerre diagrams on the Bloch ball.*
2. *We seek the center of the smallest enclosing ball.*

In our procedure, we use Delaunay tessellation, because it is the *fastest* tool to seek the center of a smallest enclosing ball of points. By the usage of quantum Delaunay triangulation on the Bloch sphere, the convex hull of the quantum states, and the radius of the smallest enclosing quantum informational ball could be determined very efficiently.

A. Delaunay Triangulation from Laguerre Diagrams

The Laguerre distance of a point x to an Euclidean ball $b = b(\rho, r)$ is defined as $d_L(\rho, x) = \|\rho - x\|^2 - r^2$. For n balls, the Laguerre diagram of $b_i = b(\rho_i, r_i)$, where $i = 1, \ldots, n$, b_i can be defined as the minimization diagram of the corresponding n distance functions as [6][7]

$$d_L^i(x) = \|\rho - x\|^2 - r^2 . \qquad (49)$$

We use the result of Aurenhammer to construct the quantum relative entropy based dual diagram of the Delaunay tessellation, by the Laguerre diagram of the n Euclidean spheres of equations [6]

$$\langle x-\rho_i', x-\rho_i'\rangle = \langle \rho_i', \rho_i'\rangle + 2\left(\mathbf{F}\left(\rho_i\right) - \langle \rho_i, \rho_i'\rangle\right). \quad (50)$$

In Figure 18, we illustrated the Laguerre diagram on the Bloch ball, and the construction of dual Delaunay diagram.

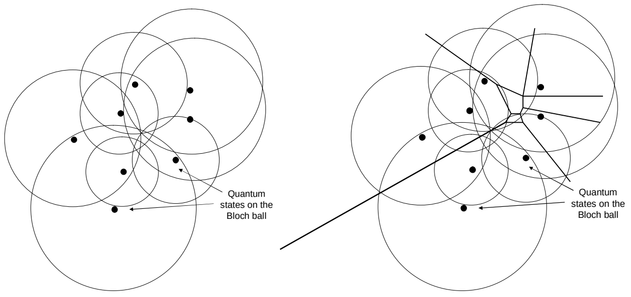

Quantum states on the Bloch ball
Quantum states on the Bloch ball

Figure 18. Tessellation on the Bloch ball, obtained by Laguerre diagram.

The most important result of this equivalence is that we can efficiently construct a quantum relative entropy-based Delaunay triangulation on the Bloch sphere, using fast methods for constructing the classical Euclidean Laguerre diagrams.

In Figure 19, we show the ordinary Delaunay triangulation. The quantum Delaunay triangulation is derived from the Laguerre diagram of Euclidean balls.

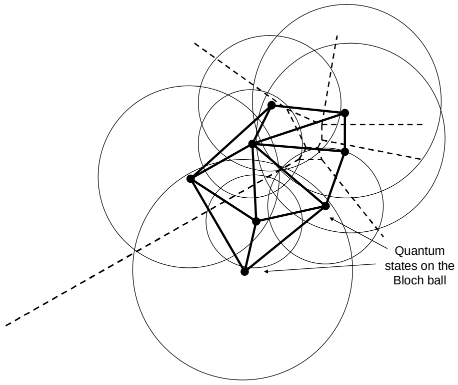


Figure 19. The regular triangulation on the Bloch ball.

We note, the image of quantum relative entropy based Delaunay triangulation by the inverse of gradient ∇_F^{-1}, is a curved triangulation whose vertices are the points of S .

B. Center of the Smallest Quantum Informational Ball

In our security analysis, we apply the approximation algorithm presented by Badoui and Clarkson [10][18], however in our algorithm, the distance measurement between quantum states is based on quantum relative entropy.

The E -core set C is a subset of the set $\mathrm{C} \subseteq \mathrm{S}$, such that for the circumcenter $\mathbf{c}$ of the minimax ball [10][18]

$$d\left(\mathbf{c}, \mathrm{S}\right) \le \left(1+\mathrm{E}\right) r, \quad (51)$$

where r is the radius of the smallest enclosing quantum information ball of the set of quantum states S [10][18].

The approximating algorithm, for a set of quantum states $\mathrm{S} = \{\rho_1, \ldots, \rho_n\}$ and circumcenter $\mathbf{c}$, first finds the farthest point ρ_m of ball set *B*, and moves $\mathbf{c}$ towards ρ_m in $\mathrm{O}(dn)$ time in every iteration step.

We denote the set of n d-dimensional balls by $B = \{b_1, \ldots, b_n\}$, where $b_i = Ball\left(S_i, r_i\right)$, S_i is the center of the ball b_i and r_i is the radius of the *i*-th ball. The smallest enclosing ball of set $B = \{b_1, \ldots, b_n\}$ is the unique ball $b^* = Ball\left(\mathbf{c}^*, r^*\right)$ with minimum radius r^* and center $\mathbf{c}^*$ [11].

In Figure 20, we illustrate the smallest enclosing ball of balls in the quantum space.

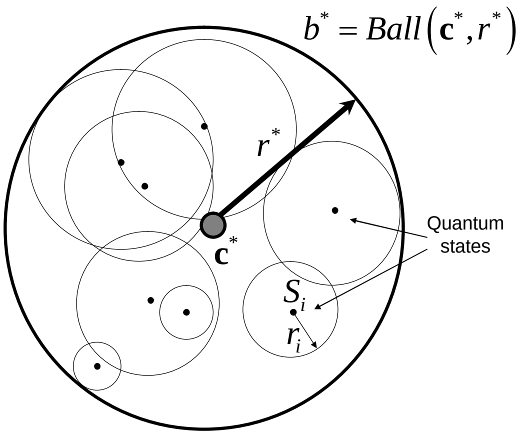


Figure 20. The smallest enclosing ball of a set of balls in the quantum space.

The *main steps* of our algorithm can be summarized as:

Algorithm 1.

```
1. Select a random center $\mathbf{c}_1$ from the set of quantum states S
                $\mathbf{c}_1 = S_1$
    for $\left(i = 1, 2, \ldots, \left\lceil \frac{1}{\mathrm{E}^2} \right\rceil\right)$
    do
2. Find the farthest point $S$ of S
                $S \leftarrow \arg\max_{s' \in \mathrm{S}} D_F\left(\mathbf{c}_i, S'\right)$
3. Update the circumcircle:
                $\mathbf{c}_{i+1} \leftarrow \nabla_F^{-1}\left(\frac{i}{i+1}\nabla_F\left(\mathbf{c}_i\right) + \frac{1}{i+1}\nabla_F\left(S\right)\right).$
4. Return $\mathbf{c}_{i+1}$.
```

The smallest enclosing ball of ball set $B=\{b_1,\ldots,b_n\}$, fully enclosing B, thus $B\subseteq Ball(\mathbf{c}^*,r^*)$. The algorithm does

$$\left\lfloor \frac{1}{\mathrm{E}^2} \right\rfloor \tag{52}$$

iterations to ensure an $(1+\mathrm{E})$ approximation, thus the overall cost of the algorithm is $\mathrm{O}\left(\frac{dn}{\mathrm{E}^2}\right)$ [18]. The smallest enclosing ball of a ball set B can be written as

$$\min{}_{\mathbf{c}} \mathbf{F}_B(\mathbf{c}), \tag{53}$$

where $\mathbf{F}_B(X)=d(X,B)=\max_{i\in\{1,\ldots,n\}} d(X,B_i)$, and the distance function $d(\cdot,\cdot)$ measures the relative entropy between quantum states [6].

The minimum ball of the set of balls is unique, thus the circumcenter $\mathbf{c}^*$ of the set of quantum states is:

$$\mathbf{c}^*=\arg\min{}_{\mathbf{c}} \mathbf{F}_B(\mathbf{c}). \tag{54}$$

At the end of our algorithm, the radius r^* of the smallest enclosing ball B^* with respect to the quantum informational distance is equal to $\min_{\sigma\in S(\square^2)} \max_{\rho\in S(\square^2)} D(\mathrm{L}(\rho)\|\mathrm{L}(\sigma))$.

The *security* of the quantum channel is determined by our geometrical model, with assumptions

$$r^*>r^*_{Eve}, \tag{55}$$

and

$$r^*\leq r^*_{Eve}, \tag{56}$$

as we have defined it at Eq. (39) and Eq. (40).

Finally, the approximated value of the fidelity parameter F_{Eve}, can be expressed as:

$$F_{Eve}=\langle\psi|^{(in)}\rho^{(out)}|\psi\rangle^{(in)}=\frac{1}{2}(1+r), \tag{57}$$

where r can be derived from the quantum informational theoretical radius r^* by $r^*=1-\mathrm{S}(r)$, where S is the von Neumann entropy.

C. The Computational Complexity of our Algorithm

The quantum relative entropy-based algorithm at the *i*-th iteration provides an $\mathrm{O}(1+\sqrt{i})$-approximation of the real circumcenter, thus to obtain an $(1+\mathrm{E})$ approximation, a time

$$\mathrm{O}\left(\frac{dn}{\varepsilon^2}\right)=\mathrm{O}\left(\frac{d}{\varepsilon^2}\frac{1}{\varepsilon}\right)=\mathrm{O}\left(\frac{d}{\varepsilon^3}\right) \tag{58}$$

is required, by first sampling $n=\frac{1}{\varepsilon}$ points [10]. Based on the computational complexity of the smallest enclosing ball, the $(1+\varepsilon)$ approximation of the fidelity of the eavesdropper cloning machine can be computed in a time

$$\mathrm{O}\left(\frac{d}{\varepsilon^2}\right). \tag{59}$$

VI. Optimized Algorithm

In this section an improved core-set algorithm is shown which determines the smallest quantum informational ball in a more efficient manner [18]. This improved method obtains a

$$\mathrm{O}\left(\frac{d}{\varepsilon}\right) \tag{60}$$

time $(1+\varepsilon)$ approximation algorithm in quantum space. In comparison with current geometric methods in the literature, this approach has a lower complexity than that presented by Kato et al. [5] or Nielsen and Nock [11].

In Figure 21, we illustrate the improved algorithm on a set of quantum states. The approximate ball has radius r, the enclosing ball has radius $r+\delta$. The approximate center $\mathbf{c}$ is denoted in black, the core-set is denoted by grey circles. The optimal radius between the center $\mathbf{c}$ and the farthest quantum state is denoted by r^* [18].

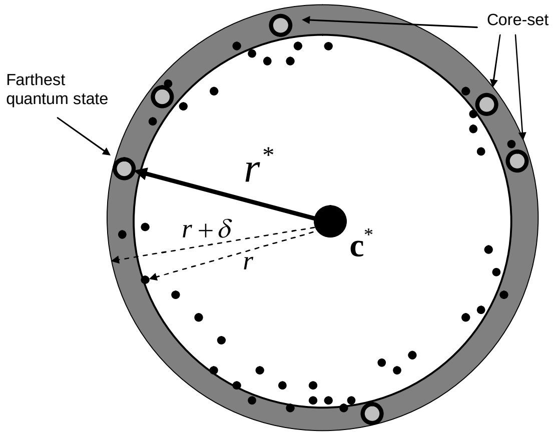


Figure 21. The approximate (light) and enclosing quantum information ball (darker).

The main steps of our improved algorithm can be summarized as:

Algorithm 2.

```
Select a random center $\mathbf{c}_1$ from the set of quantum states S
                    $\mathbf{c}_1 \in S$
$r = \frac{1}{2}\max_i D_F(\mathbf{c}_1, S)$;
$\delta = \frac{1}{2}\max_i D_F(\mathbf{c}_1, S)$;
for $\left(i = 1, 2, \ldots, \left(\frac{1}{\delta}\right)\right)$
    do
$S = \arg\max_i D_F(c, S)$;
Move $Ball(c, r)$ on the geodesic until it touches
  the farthest point $S$;
$s = \max_i D_F(c, S_i) - r$;
    if $s \le \frac{3\delta}{4}$ then
                $\delta = \frac{3\delta}{4}$
    else
                $r = r + \frac{\delta}{4}$;
                $\delta = \frac{3\delta}{4}$;
until $\delta \le \varepsilon$.
```

The improved algorithm increments the radius of the quantum information ball from a lower bound r of the optimal radius r^* [18]. In this algorithm, the optimal radius is between $r \le r^* \le r + \delta$, and the process is terminates as $\delta \le \varepsilon$, in at most

$$O\left(\frac{1}{\varepsilon}\right) \tag{61}$$

iterations.

A. Rate of Converge

We made simulations and numerical analysis on the performances of the proposed algorithms. We have compared the core-set algorithm and the improved core-set algorithm to find the smallest enclosing quantum information ball. We have analyzed the approximation algorithms for 30 center updates.

In the simulation work, the quantum information ball is generated random, and the quantum relative entropy based approximation used uniformly sampled quantum states. In our numerical analysis, we have measured the quality of the approximation with respect to quantum relative entropy. At first we have simulated the algorithm presented in Section V, then we analyzed the improved algorithm presented in Section VI.

The results of our simulation are shown in Figure 22. The x-axis represents the number of center updates to find the center of the smallest enclosing quantum informational ball, the y-axis represents the quantum informational distance between the found center $\mathbf{c}$ and the optimal center $\mathbf{c}^*$.

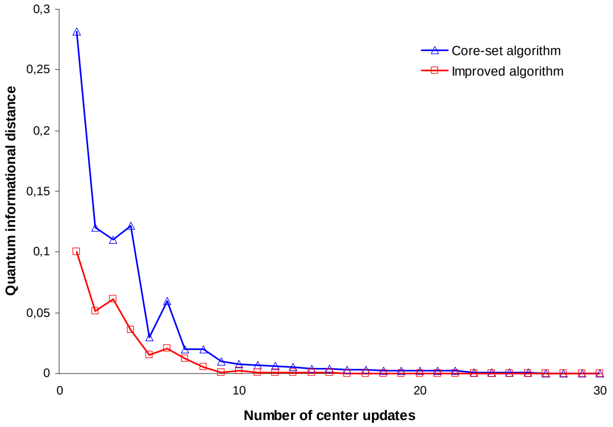


Figure 22. The converge rate of the approximation algorithms.

From the results, it can be seen that each presented algorithm finds the approximate center $\mathbf{c}$ to the optimal center $\mathbf{c}^*$ extremely quickly. The quantum relative entropy-based approximation algorithms have a very accurate convergence of $\mathbf{c}$ towards $\mathbf{c}^*$, but the improved core-set algorithm presented in Section VI converges more rapidly with a smaller number of center updates.

It can be concluded that only a small set of quantum states are required to compute the center of the quantum informational ball, with the proposed algorithms providing very solid approximations of the smallest quantum informational ball.

VII. Illustrative Example

In this section, the steps of the proposed algorithm are summarized. Here the six-state quantum cryptography protocol is used, together with the universal (UCM) quantum cloner-based attacker model. The smallest enclosing quantum informational ball for six mixed quantum states is computed. The proposed example illustrates the theoretical results of Section IV, integrating the mathematical background investigated by Gyongyosi and Imre [1], Kato et al. [5] and Nielsen and Nock [11].

As declared in Section IV, the quantum channel is secure if $r^* > r^*_{UCM}$. In the first step, the convex hull for the cloned states is computed by Delaunay tessellation, with respect to quantum informational distance.

Figure 23 illustrates the Voronoi cells for the cloned states, using the six-state quantum cryptography protocol. The cloned states were generated by Eve's universal quantum cloner machine. Cloned mixed states are denoted by $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ and ρ_6.

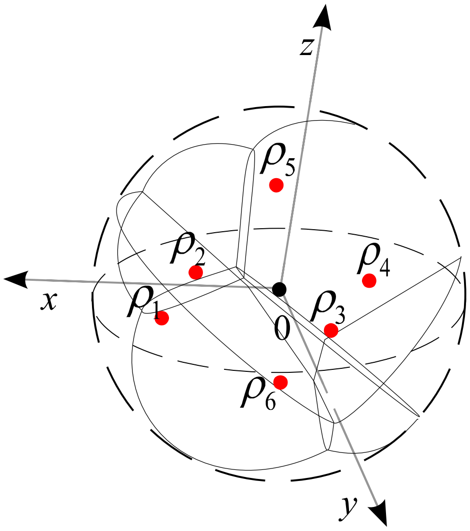


Figure 23. The dual Delaunay diagram in Bloch sphere representation.

In the next step of security analysis, the quantum Delaunay triangulation for the six cloned states is computed. Using quantum relative entropy-based Delaunay tessellation, the three-dimensional convex hull for cloned mixed quantum states $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ and ρ_6 can be calculated.

Figure 24 illustrates the three-dimensional *convex hull* (light-gray) of cloned states $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ and ρ_6.

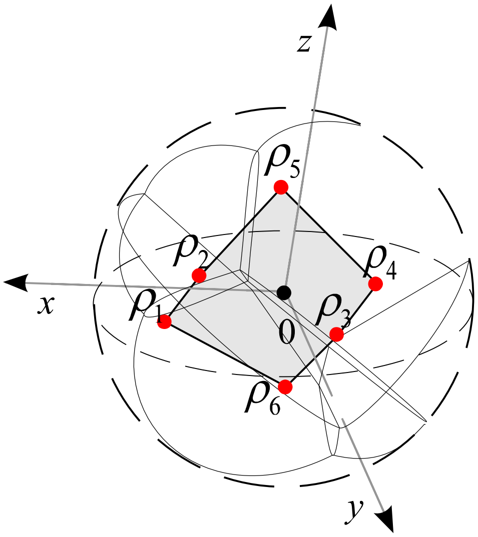


Figure 24. The convex hull computed by quantum Delaunay triangulation.

Finally, the center of the convex hull and radius r^* of the smallest enclosing quantum informational ball are computed.

In Figure 25, the center and informational theoretical radius of the smallest enclosing quantum informational ball are denoted by $\mathbf{c}^*$ and r^*, respectively.

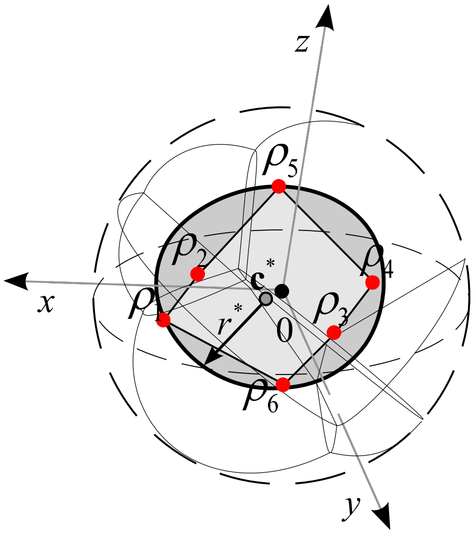


Figure 25. The smallest enclosing quantum informational ball.

Figure 26 shows an example of a two-dimensional smallest enclosing quantum informational ball.

The center point is $\mathbf{c}^* = (0.3287, 0.3274)$, and the radius r^* of the smallest enclosing quantum informational ball is $r^* = 0.4907$.

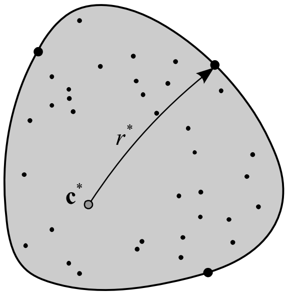


Figure 26. The smallest enclosing quantum informational ball inside the Bloch sphere.

As can be noticed, the center $\mathbf{c}^*$ of the smallest enclosing quantum informational ball differs from the center of an Euclidean ball.

VIII. Conclusion And Future Work

This paper proposes a fundamentally new method of calculating the security of a quantum channel, based on Laguerre diagrams and quantum relative entropy. Using a Delaunay tessellation on the Bloch sphere, the geometric space can be divided and can be calculated extremely efficiently, in a reasonable computational time. A generalization of an effective approximation algorithm of the smallest enclosing quantum information ball of the set of quantum states was presented, equipped with quantum relative entropy as a distance measure. This improved approximation algorithm was demonstrated to obtain a more efficient optimized geometric method. Finally the performances of the proposed methods were compared.

In this paper a new algorithm with which to compute the quantum informational theoretical impacts of an eavesdropper's cloning machine on the quantum channel was also proposed. In this analysis, the fidelity of the eavesdropper's cloning machine is numerically computed by tessellation on the Bloch sphere. A very effective core-set algorithm and an optimized core-set method were also designed. The adaptability of classical computational geometric methods in the quantum space was presented, while the performance of the basic and improved core-set algorithms were analyzed and compared. It was shown that both of the proposed algorithms provide very strong results concerning the approximation of the smallest quantum informational ball.

In terms of future work, it is hoped that the method will be extended to analyze coherent and individual attacks for quantum cryptography, while an in-depth study on the geometric impacts of physically allowed quantum cloning transformations is also intended.

REFERENCES

[1] L. Gyongyosi and S. Imre, Quantum Informational Geometry for Secret Quantum Communication, In: Proceedings of the First International Conference on Future Computational Technologies and Applications, FUTURE COMPUTING 2009, International Academy, Research and Industry Association (IARIA), pp. 580-585, 2009.

[2] W. K. Wootters and W. H. Zurek, 1982, Nature London 299, 802.

[3] V. Bužek and M. Hillery, 1996, Phys. Rev. A 54, 1844.

[4] D. Mozyrsky and V. Privman, Quantum Signal Splitting that Avoids Initialization of the Targets, Modern Phys. Lett. B 11, 1277-1283 (1997); e-print quant-ph/9609010;

[5] K. Kato, M. Oto, H. Imai, and K. Imai, "Voronoi diagrams for pure 1-qubit quantum states,quant-ph/0604101, 2006.

[6] F. Aurenhammer and R. Klein. Voronoi Diagrams. In J. Sack and G. Urrutia (Eds), Handbook of Computational Geometry, Chapter V, pp. 201–290. Elsevier Science Publishing, 2000. 03.

[7] J.-D. Boissonnat, C. Wormser, and M. Yvinec. Curved Voronoi diagrams. In J.-D.Boissonnat and M. Teillaud (Eds) Effective Computational Geometry for Curves and Surfaces, pp. 67–116. Springer-Verlag, Mathematics and Visualization, 2007.

[8] P. W. Lamberti, A. P. Majtey, A. Borras, M. Casas, and A. Plastino. Metric character of the quantum Jensen-Shannon divergence. Physical Review A (Atomic, Molecular, and Optical Physics), 77(5):052311, 2008.

[9] M. A. Nielsen and I. L. Chuang: Quantum Computation and Quantum Information. Cambridge University Press, 2000.

[10] M. Badoiu and K. L. Clarkson. Smaller core-sets for balls. In Proceedings 14th ACM-SIAM Symposium on Discrete Algorithms, pages 801–802, 2003.

[11] F. Nielsen and R. Nock, "On the smallest enclosing information disk," Information Processing Letters, vol. 105, no. 3, pp. 93–97, 2008.

[12] S. Imre and F. Balazs: Quantum Computing and Communications – An Engineering Approach, Published by John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005, ISBN 0-470-86902-X, 283 pages

[13] L. Gyongyosi and S. Imre: Computational Geometric Analysis of Physically Allowed Quantum Cloning Transformations for Quantum Cryptography, In Proceedings of the 4th WSEAS International Conference on Computer Engineering and Applications (CEA '10). Harvard University, Cambridge, USA. 2010. pp. 121-126. Paper 18.

[14] N.J. Cerf and M. Bourennane, A. Karlsson and N. Gisin, 2002, Phys. Rev. Lett. 88, 127902.

[15] G.M. D'Ariano and C. Macchiavello, 2003, Phys. Rev. A 67, 042306.

[16] A. Acín, N. Gisin, L. Masanes and V. Scarani, 2004, Int. J. Quant. Inf. 2, 23.

[17] R. Panigrahy. Minimum enclosing polytope in high dimensions. CoRR, cs.CG/0407020, 2004.

[18] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In Proceedings 34th ACM Symposium on Theory of Computing, pages 250–257, 2002.

[19] P. Kumar, J. S. B. Mitchell, and A. Yıldırım. Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In Algorithm Engineering and Experimentation, LNCS, pages 45–55. Springer-Verlag, 2003.

[20] B. Kraus, N. Gisin, and R. Renner, 2005, Phys. Rev. Lett. 95, 080501.

[21] M. Hillery, M. Ziman, and V. Bužek, 2004, Phys. Rev. A 69, 042311.

[22] N. J. Cerf, O. Krüger, P. Navez, R. F. Werner, and M. M. Wolf, 2005, Phys. Rev. Lett. 95, 070501.

[23] G. M. D'Ariano and C. Macchiavello, 2003, Phys. Rev. A 67, 042306.

[24] N., G. Gisin, Ribordy, W. Tittel, and H. Zbinden, 2002, Rev. Mod. Phys. 74, 145.

[25] N. Gisin and S. Popescu, 1999, Phys. Rev. Lett. 83, 432.

[26] A. Niederberger, V. Scarani, and N. Gisin, 2005, Phys. Rev. A 71, 042316.

Security Patterns for Untraceable Secret Handshakes with optional Revocation

Annett Laube*, Alessandro Sorniotti†‡, Paul El Khoury§‡, Laurent Gomez‡ and Angel Cuevas¶
* *Bern University of Applied Science, Switzerland,*
Email: annett.laube@bfh.ch
† *Institut Eurecom Sophia-Antipolis, France*
‡ *SAP Research Sophia-Antipolis, France*
Email: alessandro.sorniotti@sap.com, laurent.gomez@sap.com
§ *LIRIS University of Claude Bernard Lyon 1, France*
Email: paul.el-khoury@liris.cnrs.fr
¶ *University Carlos III of Madrid, Spain*
Email: acrumin@it.uc3m.es

***Abstract*—A security pattern describes a particular recurring security problem that arises in specific contexts and presents a well-proven generic solution for it. This paper describes Untraceable Secret Handshakes, cryptographic protocols that allow two users to mutually verify another's properties without revealing their identity or other sensitive information. The complex security solution is split into smaller parts, which are described in an abstract way. The identified security problems and their solutions are captured as SERENITY security patterns. The structured description together with motivating scenarios from three different domains makes the security solution better understandable for non-security experts and helps to disseminate the security knowledge to application developers.**

***Keywords*-security patterns; secret handshake; cryptographic protocols; mutual authentication;**

I. Introduction

Today's pioneer organizations recognize that performance accelerates when information security is driven into the very framework of business [2]. In the past, security experts developed secure systems and standardized sufficient security solutions to satisfy security requirements for various contexts. Most of these standards provide comprehensive methodologies for specifying, implementing and evaluating security of IT products. Rarely non-specialists are capable of correctly understanding and realizing such standards. The description of these standards in natural language limits their ability in passing knowledge to novice security users. The fundamental added value of adopting the security patterns approach is providing security for non-security experts [3][4].

Application developers transform clients' requirements into business applications. Business solutions and security solutions are often designed and developed at different coordinates of space and time. Considering the lack of expertise application developers are often incapable of being compliant to security regulations protecting the clients' business.

Security patterns capture security expertise abstract and concrete at the same time. This approach fits well as candidate link between security experts and application developers to encompass business applications with a security shield.

This paper focuses on capturing the security solution for secret handshakes described in [5] and [6] as security patterns, following the SERENITY methodology [4][7]. Parties cooperating in hostile networked environments often need to establish an initial trust. Trust establishment can be very delicate when it involves the exchange of sensitive information, such as affiliation to a secret society or to an intelligence agency.

Secret handshakes are first introduced in 2003 by Balfanz et al. [5] as mechanisms designed to prove group membership and share a secret key between two fellow group members. The purpose of these protocols is – as pointed out in [8] – to model in a cryptographic protocol the folklore of real handshakes between members of exclusive societies or guilds.

With a secret handshake, two users can simultaneously prove to each other possession of a property, for instance membership to a certain group. The ability to prove and to verify is controlled by a certification authority that issues credentials and matching values respectively. Users are not able to perform a successful handshake without the appropriate credentials and matching values; in addition protocol exchanges should be untraceable and anonymous.

Even though there are several ways for implementing the secret handshake protocol. Our purpose is to capture the properties and functionalities that are common to all implementations. Therefore, any particular solution could be derived from the defined security patterns. Furthermore, capturing expertise as a pattern makes security solutions for a given problem more general. It is easier for non-security experts to find a suitable solution for a particular problem by searching into the patterns' library through properties and features.

With the capturing of Untraceable Secret Handshakes as security patterns, we increase the number of standard security solutions available to application developers. In this paper, we describe not only the secret handshake protocol as in [1], we also discuss variants of the security solution adapted to specific contexts and use cases. The security patterns and their relationships are described in detail. The SERENITY pattern template is extended and complemented by standard UML models. The three scenarios from different domains illustrate the broadness of possible use cases for secret handshakes.

The paper is organized as follows. Work related to secret handshakes is discussed in Section II. In Section III, we overview security patterns and introduce the SERENITY security pattern template. Section IV proposes three scenarios to illustrate the use of secret handshakes in real-world applications. We describe the proposed security solution in detail in Section V and define an abstract model used to capture this security solution as a combination of patterns. Next, Section VI describes the security patterns and integration scheme of the proposed solution. Section VII highlights advantages of the SERENITY pattern approach and points out weaknesses. In Section VIII, we conclude the paper.

II. Related Work

After the introduction of secret handshakes in 2003 by Balfanz et al. [5] many papers have further investigated the subject. New schemes have been introduced, achieving for instance reusable credentials (the possibility to generate multiple protocol exchanges out of a single credential with no loss in untraceability) and dynamic matchings (the ability to verify membership for groups different from one's own).

Castelluccia et al. in [9] introduce the concept of CA-Oblivious encryption and show how to build a secret handshake scheme from such a primitive. Users are equipped with credentials and matching references (in this particular case embodied by a public key and a trapdoor) that allow them to pass off as a group member and to detect one. In [10], Meadows introduces a scheme that is similar to secret handshakes, despite the fact that the security requirements are slightly different – for instance, untraceability is not considered. In [11], Hoepman presents a protocol, based on a modified Diffie-Hellman key exchange [12], to test for shared group membership, allowing users to be a member of multiple groups. In [8], Vergnaud presents a secret handshake scheme based on RSA [13]. In [14], Xu and Yung present the first secret handshake scheme that achieves unlinkability with reusable credentials: previous schemes had to rely upon multiple one-time credentials being issued by the certification authority. However, the presented scheme only offers a weaker anonymity. In [15], Jarecki et al. introduce the concept of affiliation-hiding authenticated key exchange, very similar to group-membership secret handshakes; the authors study the security of their scheme under an interesting perspective, allowing the attacker to schedule protocol instances in an arbitrary way, thus including MITM attacks and the like.

In [16], Ateniese et al. present the first secret handshake protocol that allows for matching of properties different from the user's own. Property credentials are issued by a certificate authority.

Already the Balfanz' original scheme [5] supports revocation, but has a number of drawbacks, for instance the fact that it relies on one-time credentials to achieve untraceability. After this seminal work, many papers have further investigated the subject of secret handshake, considerably advancing the state of the art. The work by Castelluccia et al. [9] has shown how, under some specific requirements (namely CA-obliviousness), secret handshakes can be obtained from PKI-enabled encryption schemes. Other schemes have followed this approach [8][17] offering similar results, albeit with different nuances of unlinkability. Almost all the schemes in this family support revocation of credentials; however the functionalities offered are limited to proving and verifying membership to a common group. In [6], the authors present the first secret handshake with dynamic matching of properties under stringent security requirements. In [18], the revocation support for this kind of secret handshakes is presented.

All the mentioned publications about secret handshakes focus on the cryptographic details of the secret handshake protocols and their formal proofs. Our goal is to capture the common properties and functionalities of secret handshake as general security solution and to make them available for non-security experts. The abstraction from the implementation details allows us to provide a functional view of the needed operations of all involved parties. Our conceptual model and the UML sequence diagrams together with different application scenarios make secret handshakes easier understandable for solution architects and application developers and allow the easy integration of secret handshake in existing or emerging applications.

III. Security Patterns Overview

Research techniques for security patterns are interestingly different from other kinds of research. In software engineering and security engineering innovative results are measured by new solutions brought to market, whereas differently, innovative research in techniques for (*security*) patterns is measured by the successful provision of existing best practices, standards or well proven solutions from experts to lay users. The added value of these techniques for patterns is devoted to the format, validation techniques and means used to promote experts' knowledge to novice security users. Populating a collection of patterns is indeed time-consuming, but once realized the invested effort pays off. To accomplish the security patterns' 'mission', a list of objectives is summarized in four fundamental steps [3]. First, most of the novice security users should understand how experts approach key security problems. Second, security experts should be able to identify, name, discuss and teach both problems and solutions efficiently. Third, problems should be solved in a structured way. Fourth, dependencies and side-effects should be identified and considered appropriately. The connotation of these objectives emerged as appealing for research studies.

In [19], the authors summarize the pattern engineering life cycle, from creation until deployment. In a nutshell this process is presented hereafter as several steps for the creation process, i.e., numbered by ‘E’, and the deployment one, i.e., numbered by ‘A’:

E1 Finding a pair of recurring problem and its corresponding solution from knowledge and/or experiences of software development.
E2 Writing the found pair with forces in a specific pattern format.
E3 Reviewing and revising the written pattern.
E4 Publishing the revised pattern via some public or private resource (WWW, book or paper, . . .).
A1 Recognizing context and security problems in software development.
A2 Selecting software patterns that are thought to be useful for solving the recognized problems.
A3 Applying the selected patterns to the target problem.
A4 Evaluating the application result.

The usual natural language description for security patterns opens room for different interpretation of solutions provided and problems described by these patterns. Hence, none of the previously four objectives of the patterns’ mission can be achieved. First known contribution to security patterns, is the work from J. Yoder and J. Barcalow proposing to adapt the object-oriented solutions to recurring problems of information security [20]. Seven patterns were presented to be used when dealing with application security. A natural evolution of this work is the proposal presented by Romanosky in [21]. It takes into consideration new questions that arise when securing a networked application. Following this particular path, Schumacher et al. [22] presented a set of security patterns for the development process. Fernandez and Pan [23] describe patterns for the most common security models such as Authorization, Role-Based Access Control and Multilevel Security. Recently in [24], the same authors highlighted the need to develop additional security patterns for database systems in order to integrate it into secure software development methodology. These security patterns were rarely adopted in the security field. Indeed their description in natural language limits their applicability and forbid any reasoning mechanism.

The SERENITY EU project through a list of narrow yet complex studies [4][7][25][26] tackles the security patterns objectives. The SERENITY partners presented in [4] the SERENITY model of secure and dependable applications. Moreover, using security patterns they showed how to address, along with the tools provided, the challenge of developing, integrating and dynamically maintaining security mechanisms in open, dynamic, distributed and heterogeneous computing systems.

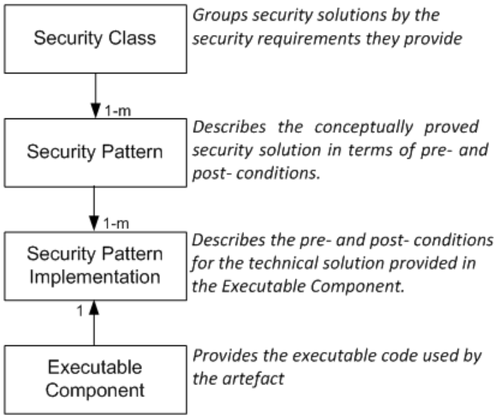

Security Class
Groups security solutions by the security requirements they provide
1-m
Security Pattern
Describes the conceptually proved security solution in terms of pre- and post- conditions.
1-m
Security Pattern Implementation
Describes the pre- and post- conditions for the technical solution provided in the Executable Component.
1
Executable Component
Provides the executable code used by the artefact

Figure 1. SERENITY Security Artefacts

One of the essential proposals from SERENITY is to provide novice users the SERENITY Security & Dependability pattern package. This package comprises *expert-proofed* security solutions and *tested* plug-and-play deployable implementations. The research interest in security patterns focuses in particular on capturing solutions for recurring security problems that arise in different contexts. The granularity of security problems analyzed and captured in a pattern, can be quite different. Usually a complex security solution is not captured in a single pattern. Solutions consisting of several patterns cover better the generality aspect of the abstract solution. To have an intuitive description for the solution proposed in this paper, we adopt the SERENITY approach using four artefacts. The description of these artefacts enables selection, adaptation, usage and monitoring at runtime by automated means. The hierarchy is composed of four artefacts (depicted in Figure 1): Security Classes, Security Patterns, Security Implementations and Executable Components. Although this paper emphasized the use of the Security Pattern artefact, in different studies [27][28][29][30] an intuitive and extensive description of all artefacts is presented.

In SERENITY, *security patterns* are detailed descriptions of abstract security solutions that contain all the information necessary for the selection, instantiation and adaptation performed on them. Such descriptions provide a precise foundation for the informed use of the solution and enhance the trust in the model.

This paper relies on the SERENITY representation of security patterns [4][26] to transfer the first three objectives of security patterns for the Untraceable Secret Handshakes to non-security experts. The most important parts of a security pattern description are the following:

Problem and context: The problem is the vulnerable part in an asset that can also be described as requirements, which need to be solved. The context defines the recurring situation where the problem/requirement can be found.

Solution: The solution is defined as a mechanism that is used to resolve the corresponding requirement/problem. It defines the sequential flow of operations in solving the security problem.

Role: The entity applying the pattern is described together with its interactions with other entities from the pattern context.

Pre-Conditions: They indicate assumptions and restrictions related to the deployment of the pattern. Before applying a pattern, users or applications in some cases should check the satisfiability of these pre-conditions. Obviously, pre-conditions are elements used during the selection of suitable patterns for a particular problem.

Properties: They describe which security elements the pattern provides. This is the basic element used to discriminate whether a pattern is useful for a security problem or not.

Features: They are additional characteristics to the patterns' properties used to select suitable patterns.

Consequences: They are the effects (benefits and drawbacks) of the compromise resulting from the application of the pattern's solution. In general, using security solutions implies an increase in cost (economic, more complex mechanisms, etc.).

Variants: This describes variants and possible extensions of the pattern.

Related patterns: They name related patterns, integration schemes and the kind of the relationship (e.g., similarity, dependency, extension).

Often security solutions are too complex to be captured in a single security pattern. Therefore an additional artefact, the *integration scheme (IS)*, was introduced. An IS defines the combination of security patterns and their relationships.

IV. Scenarios

In this section, we want to show how untraceable secret handshakes are used in real-world applications. Our first example is a use case from the EU Project R4eGov [31] for **Mutual Legal Assistance** in international crimes.

Several EU justice forces led by Europol [32] cooperate in order to solve cross-boundary criminal cases (in Figure 2, a workflow example is shown). EU regulations define official processes that must imperatively be followed by operating officers: in particular, these processes mandate which institutions must cooperate upon each particular case. During such collaboration, for instance, a member of France's *Ministère de la Défense* must cooperate with a member of the *Bundesnachrichtendienst*, Germany's intelligence service, to investigate on an alleged internal scandal. The two officers may need to meet secretly and to authenticate themselves on-the-fly. Both are definitely reluctant to disclose their affiliation and purpose to anybody but the intended recipient.

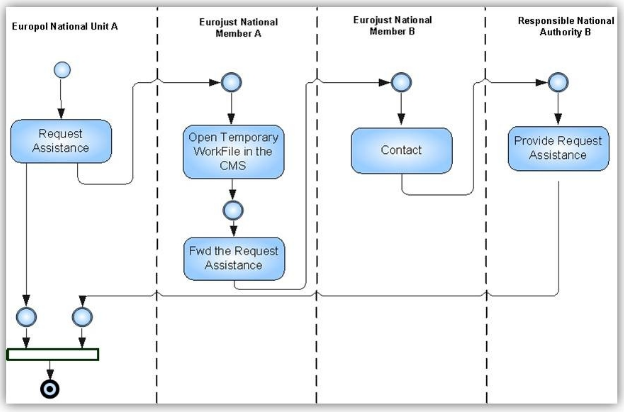

Europol National Unit A
Eurojust National Member A
Eurojust National Member B
Responsible National Authority B
Request Assistance
Open Temporary WorkFile in the CMS
Fwd the Request Assistance
Contact
Provide Request Assistance

Figure 2. Mutual legal assistance scenario

A scenario from another business domain is the **Incompatible chemicals in proximity** use case from the CoBIs EU project [33]. Let us assume that drums are stored in a warehouse (see Figure 3); each drum contains a liquid chemical and is equipped with a wireless sensor that is able to perform a secret handshake with other sensors in proximity. Drums can contain some reactive chemicals: the proximity of these drums must be considered dangerous. The goal is to generate safety-critical alerts based on an untraceable secret handshake. The drums get credentials related to their containing chemicals and a list of references to match the reactive liquids. After a successful matching, an alert is generated and sent to the storage manager. The security features of the secret handshakes described in [6] allow to exchange information of the containing chemicals without revealing them on a wireless channel. Drums with dangerous contents cannot be identified or traced by unauthorized persons.

In the third scenario, we regard **Online Social Networks** (OSNs), like Facebook. A problem, which is particularly felt among social network users, is identity theft and identity spoofing [34]. The root of the problem is that in many OSNs there is little or no verification that a person that joins the social network is really who he or she claims to be. Additionally is the social network users' decision on whether to accept a friendship request based on name, pictures and fragments of text, information that is often easily retrievable elsewhere on the Internet. A viable solution consists on users creating ad-hoc, trusted groups outside of the social network, issuing group membership credentials and presenting such credentials upon friendship invitations within the social network. A natural evolution of the aforementioned trusted friend groups are **Secret Interest Groups (SIGs)** [35], user-created groups with particular attention to confidential or simply privacy-sensitive topics. Indeed users of online

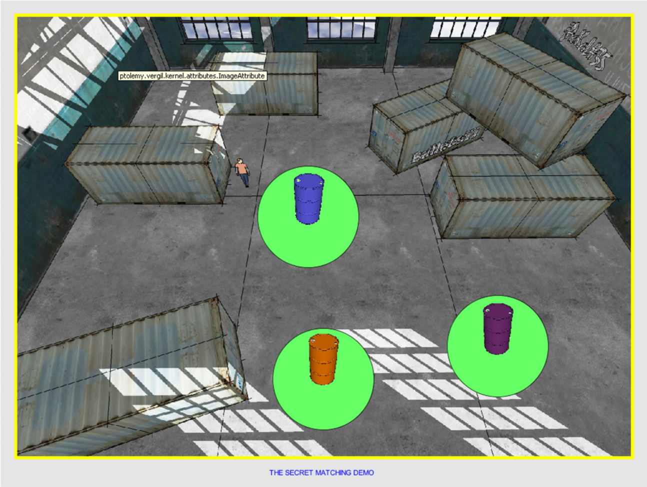


Figure 3. Incompatible chemicals in proximity scenario

social networks are often also exchanging personal and sensitive material; moreover, OSNs are more and more the theater of political, religious debate, often used as means to exchange confidential material that cannot go through official channels. Secret handshakes [5] are the main pillar to fulfill the operational and security requirements of a generic SIG framework.

Often it is desirable to support **revocation** for secret handshakes. In the SIG use case, revocation is required when either a SIG member got his membership token stolen or when he no longer qualifies for membership. Since SIG membership credentials can be used directly to authenticate to another SIG member a *reactive revocation* approach is required, that singles out revoked credentials based on a revocation list. In the scenario of assistance in international crimes, the credentials of the operating officers to authenticate should be automatically revoked when the criminal case was closed. *Proactive revocation* techniques are based on time-bound credentials, which have to be updated periodically, can be used in this case.

V. Solution Description

A *Secret Handshake*, first introduced in [5], is a mechanism devised for two users to simultaneously prove to each other possession of a property, for instance membership to a certain group. The ability to prove and to verify is strictly controlled by a certification authority that issues property credentials and matching values. Users are not able to perform a successful handshake without the appropriate credentials and matching values; in addition protocol exchanges have to be untraceable and anonymous.

We present a pattern for Untraceable Secret Handshakes with proof of group membership as described in [5] or [6]: users are required to possess credentials and matching values issued by a trusted certification authority in order to be able to prove and to verify possession of a given property. Therefore the certification authority retains the control over who can prove what and who can disclose which credentials. However verification is dynamic, in that it is not restricted to own properties.

The secret handshake requires an *initialization phase* followed by a *matching phase*, which can be repeated several times.

A. Initialization phase

A secret handshake is performed between two parties, in the following also called **users**. To carry out a secret handshake each user needs a property credential and matching values. A **property credential** is a certification of the user's property by a trusted entity. The entity responsible for the certification of properties is the **Certification Authority (CA)**. The CA is a trusted entity that after a successful verification of a property grants the user a credential. To verify a certain property the identity of the user and the related context are examined. This operation is described in the pattern *Property certification* and can be an offline step.

The CA can be a single person or organization, like Europol, the European Law Enforcement Organisation, in the Mutual Legal Assistance scenario. Europol assists the authorities in the EU Member States in preventing and combating terrorism, and other serious forms of international organized crime. Europol can certify the involvement of a justice force in a specific criminal case. In the third scenario, a non-empty group of so called SIG managers is responsible to verify the group membership in an offline process and to manage the property credentials, in this case called *membership credentials*.

The **matching value** allows a user to verify that the other user has a particular certified property. The user can get one or more matching values from the CA. The CA, according to a set of policies, delivers the matching values to a requesting user after verifying his identity and the context. The process of obtaining the matching values is described in the security pattern *Property Certification*. The policy with the relationships between property credentials and matching values has to be defined beforehand.

Figure 4 depicts an abstract model highlighting input, output and entities specific for this phase. The data flow between the CA and the user helps to understand the proposed solution. The security pattern is shown as rounded rectangle including the operations the applying entity has to perform. The exchanged data and documents (e.g., credentials and policies) are depicted together with the flow to and from the manipulating operation.

Like described in Section IV, revocation support can be required for a secret handshake scenario. Depending on the desired revocation technique (reactive or proactive) the properties credentials have to be generated differently. The cryptographic details for generating credentials with

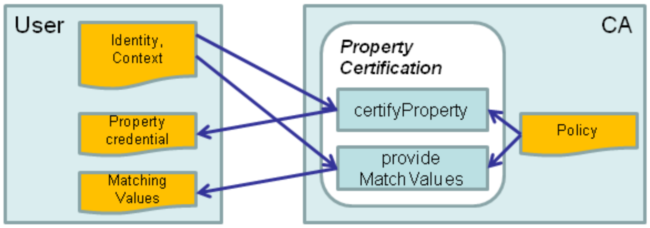


Figure 4. Conceptual Model - Initialization phase

identification handles for reactive revocation are described in [18]. The proposed scheme there, offers a solution that on the one hand assures that the protocol messages are untraceable, and on the other offers a credential tagging to single out the revoked ones.

B. Matching phase

The secret handshake itself is carried out between two users and can be repeated infinitely. It consists of two mandatory parts: the secure match of properties and the proof that both parties possess the same key after the matching. The revocation support can be added optionally when needed. The relationships between the security patterns and the data flow between the entities are pictured in Figure 5.

The secure match is initiated by one of the users, e.g., user A. User A sends an internal state (state A), e.g., a nonce, to user B. User B replies with another state (state B) and his hidden credential, computed from the received state and his property credential. When user A receives the hidden credential from user B, he is able to match it with his matching values (see e.g., [6] for details about the used cryptographic algorithms). To complete the matching protocol, user A computes his hidden credential and sends it to user B. This behaviour is described by the security pattern *Secure Match* and has to be applied to both parties.

When reactive revocation is supported by the property credentials (i.e., they contain an *identification handle*) and the CA maintains a list of revoked credentials (represented by so called **revocation handles**) the secure match can be extended. Before user A matches the received hidden credential from user B, user A checks if the received credential B is invalid. Invalid means that the revocation handle computed from the received credential matches one of the revocation handles from the publicly available revocation list. This behaviour is caught by the pattern *Credential Revocation*.

After a successful matching both parties, user A and B, own a secret, for instance a key (see key A and key B in Figure 5), that can be used to secure the further communication between the two. In order to prove that both parties have the knowledge of the same key the security pattern *Mutual Key Proof of Knowledge (Mutual Key PoK, PoK)* can be used. The parties exchange encrypted information to prove their knowledge without disclosing the key directly.[1] User A sends a challenge to user B. The challenge 1 contains an internal state, e.g., a random number, encrypted with the key obtained from the secure match (key A). User B replies with challenge 2: he uses his key B from the secure match to decrypt the challenge 1, modifies the result (e.g., he increments the number by 1) and sends this encrypted with his key B. User A can now verify that user B has obtained the same key while decrypting challenge 2 with his key and inversing the operation from user B (e.g., decrementing the result by 1). If the result is identical to the state used in challenge 1, user A has proved that B has an identical symmetric key. To complete the protocol, user A has to reply challenge 2. User A decrypts challenge 2, modifies the result in the agreed way and sends back the encrypted result. User B can now perform the verification on his side.

VI. Security Patterns and Integration Scheme

We identified the following four security patterns: *Property Certification*, *Secure Match*, *Credential Revocation* and *Mutual Key Proof of Knowledge*. The integration scheme that describes our security solution entirely is named *Untraceable Secret Matching*. In this section, each of them is defined in detail.

A. Property Certification Pattern

Problem and context: The secret handshake is based on the mutual verification of user's properties. In a first step, the possession of a given property has to be verified by a Trusted Third Party (TTP). This TTP certifies the property for an identified user by issuing a credential. In a second step, each user needs matching values. The matching values are given by a TTP according to a policy.

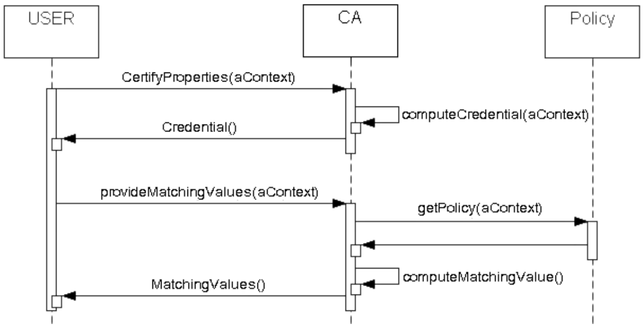

USER
CA
Policy
CertifyProperties(aContext)
computeCredential(aContext)
Credential()
provideMatchingValues(aContext)
getPolicy(aContext)
computeMatchingValue()
MatchingValues()

Figure 6. UML diagram - Property Certification

Solution: The pattern defines the following operations. In Figure 6, the interaction between the entities is shown.

- **certifyProperty**: The input of this operation is the application context of the handshake. The context contains

[1] Depending on the real setup, also other protocols could be used to prove the knowledge of an identical key. For example, both users could send their keys to a TTP that verifies the keys and returns the verification result.

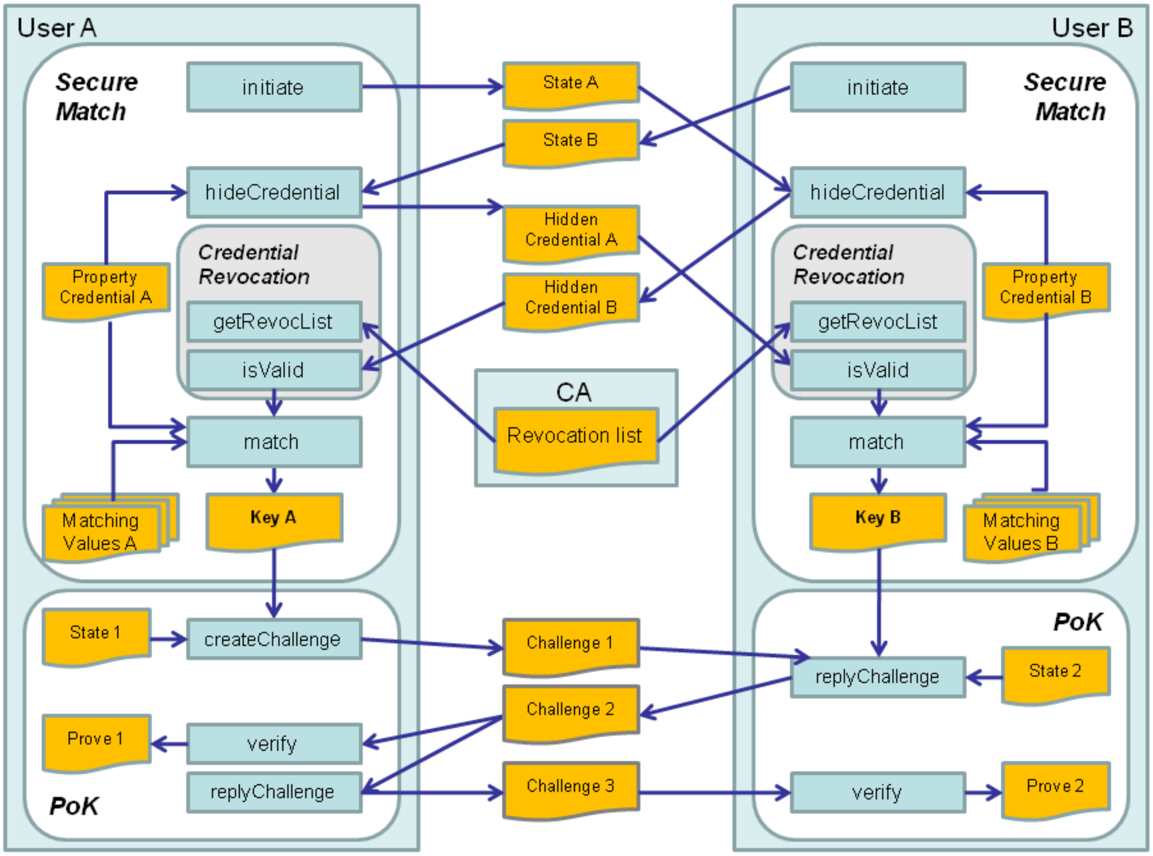


Figure 5. Conceptual Model - Matching phase

all information needed to decide about the possession of a predefined property, e.g., the user's identity and the process where he is involved in. The possession of the user's property is verified in the given context and if this was successful, a credential (resp. *property credential*) representing the property is returned.

- **provideMatchingValues**: The input of this operation is the application context (including user's identity) of the handshake. According to the policy, the operation returns a set of *Matching Values*.

Roles: The pattern is applied by a Certification Authority, which grants property credentials to users and provides them with matching values.
Pre-Conditions:

- The entity applying this pattern is a trusted party.
- A list of properties that can be certified and a policy how to match the different properties have been defined.
- Communication channels are secured.

Properties: Certification, Policy Enforcement
Features: Revocation Support (optional)
Consequences: The issued property credentials and matching values have been kept secret by the users and stored in a safe place.
Variants: According to requirements of revocation support, the created credentials have to carry an *identification handle* to support reactive revocation (see pattern *Credential Revocation* in Section VI-C) or to be time-bound for proactive revocation.
Relationships: The pattern realizes the initialization phase of a secret handshake (IS *Untraceable Secret Matching*). The second phase - the matching phase - of the secret handshake is described by the patterns *Secure Match* and *Mutual Key PoK* (see Section VI-B and VI-D).

B. Secure Match Pattern

Problem and context: A user wants to exchange secretly credentials with another user in order to verify that the other party possesses a matching property.
Solution: The pattern defines the following operations. In Figure 7, the protocol between the two parties is shown.

- **initiate**: An internal state, e.g., a nonce value, is sent to the other party to initiate the handshake.
- **hideCredential**: A hidden credential is generated from the received state and the property credential of the user and randomized. The result is sent to the other party.
- **match**: Input of the operation is the received hidden credential from the other party, the owned property cre-

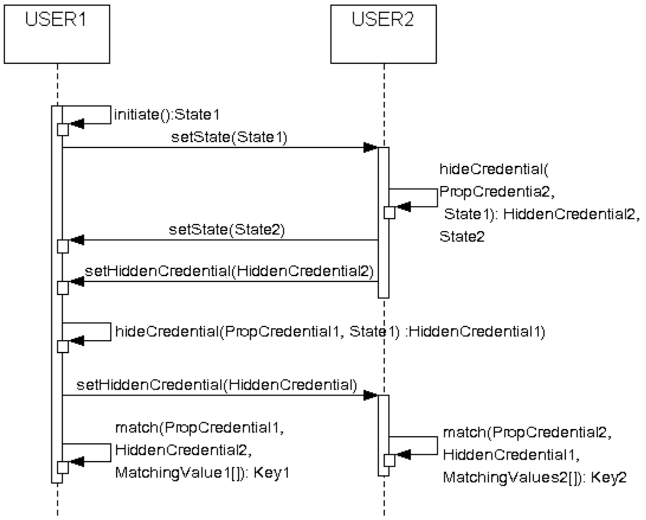


Figure 7. UML diagram - Secure Match

dential and the matching values. The operation checks, if the received credentials matches one of the matching values. The result of the match is a key.

Roles: The pattern is applied by 2 parties (users) who want to authenticate secretly.

Pre-Conditions:

- The entity applying this pattern possesses a property credential and a non-empty set of matching values.
- Both parties involved in the match must apply the same pattern implementation to ensure interoperability.

Properties: Authentication, Property Validation

Features: Untraceability, Key establishment, Fairness, Revocation Support (optional)

Consequences: None

Variants: The pattern *Credential Revocation* can be integrated to support reactive revocation.

Related patterns: The pattern realizes the matching phase of a secret handshake (IS *Untraceable Secret Matching*). The first phase - the initialization phase - is described by the pattern *Property Certification*. The pattern can be extended by pattern *Credential Revocation*.

C. Credential Revocation Pattern

Problem and context: It is possible that a (property) credential becomes invalid (e.g., the credential was compromised or a user has to leave a certain group). The user performing a secret handshake has to know, if the credential of the other party is valid in order to refuse any interactions with the concerned users.

Solution: The pattern defines the following operations.

- **getRevocationList**: Retrieves the list of revocation handles from the CA.
- **isValid**: The function computes first the revocation handle from the received hidden credential from the other party (details of the computation are in [18]). In a second step, it is verified that this handle is not part of the revocation list.

Roles: The pattern is applied by a user who wants to verify if a credential is valid and not revoked.

Pre-Conditions:

- The chosen secret handshake protocol supports revocation, that means the property credentials granted by the CA contain *identification handles*.
- The CA maintains a publicly available list of revocation handles.
- Compromised property credentials have to be announced to the CA.
- Dependent on the application context, the CA has to verify regularly that all group members are still qualified for group membership and otherwise to add the non-qualified members to the revocation list.

Properties: Revocation support for secret handshakes

Features: Anonymity and untraceability for valid credentials, traceability for revoked credentials

Consequences: Additional communication effort for the user to retrieve the revocation list from the CA before performing a secret handshake. Additional computation effort for the user to perform the operation isValid (computational cost $O(\frac{n}{2})$ in average, worst case $O(n)$).

Variants: None

Related patterns: The pattern is an optional part of the *Secure Match* pattern. The pattern is functional similar to the security pattern *Certificate revocation* described in [36]. The difference lays in the revoked objects (certificates vs. property credentials).

D. Mutual Key Proof of Knowledge Pattern

Problem and context: A user possesses a secret key. He wants to verify if another user possesses the same key without disclosing his own key.

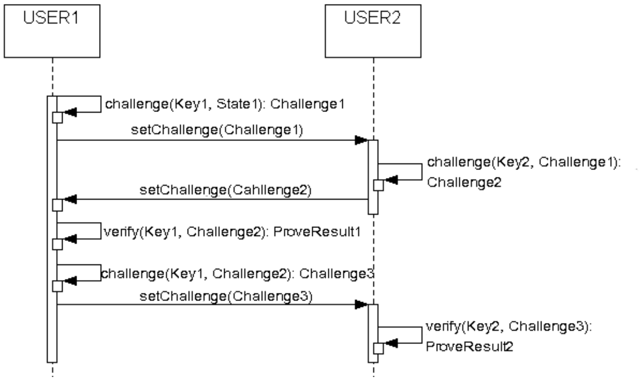

USER1
USER2
challenge(Key1, State1): Challenge1
setChallenge(Challenge1)
challenge(Key2, Challenge1):
Challenge2
setChallenge(Cahllenge2)
verify(Key1, Challenge2): ProveResult1
challenge(Key1, Challenge2): Challenge3
setChallenge(Challenge3)
verify(Key2, Challenge3):
ProveResult2

Figure 8. UML diagram - Mutual Key PoK

Solution: The pattern defines the following operations. In Figure 8, the protocol between the two parties is shown.

- **createChallenge**: A challenge is sent to the other party, containing e.g., an internal state (random number) encrypted with the owned key.
- **replyChallenge**: A received challenge is decrypted with the own key. An agreed operation (e.g., increment by 1) is applied to the decrypted value. The result is encrypted and sent back to the other party.
- **verify**: The answer to the challenge and the internal state for the current instance of the pattern are processed; this operation returns *true* in case the two users indeed share the same key.

Roles: The pattern is applied by 2 parties (users) who want to verify mutually if they possess an identical key.
Pre-Conditions:

- The entity applying this pattern possesses a key.
- A protocol that specifies how to reply to a challenge is agreed on both sides.

Properties: Key verification
Features: Non-disclosure of own key
Consequences: The mutual protocol is unfair since one user knows first if the other party possesses the same key and can therefore exploit his advantage by not replying his side of the challenge.
Variants: None
Related patterns: Can be used in the matching phase of a secret handshake (IS *Untraceable Secret Matching*).

E. Integration Scheme: Untraceable Secret Matching

This integration scheme describes the full solution made of a combination of the defined security patterns. It synchronizes the operations among the patterns in order to provide the desired security solution.

The sequence of the IS operations for the proposed solution are provided in Table I. We used the keyword $roles$ of the SERENITY security patterns' language [4] in order to separate between the two users of the protocol using the same security pattern.

The IS contains the security property **Mutual Authentication**. That means that the complex security solution provides a mechanism that allows two parties authenticating each other based on property credentials and corresponding matching references controlled by a certification authority.

Additional security features of the IS are:

- **Untraceability**: Consider an adversary with valid property credentials and matching references and able to perform a secret handshake with legitimate users. His goal is – given any two disguised credentials - to trace them to having been generated from the same credential, so as to prove possession of the same property and at the same time to the same group. The attacker cannot decide whether there is a property that both credentials can be matched to.
- **Impersonation Resistance**: It is computationally infeasible for an attacker with valid credentials and matching references, to impersonate a user owning a given property credential, which the attacker does not dispose of and did not steal.
- **Detector Resistance**: An adversary cannot verify the presence of a property of his choice without owning the corresponding matching references. That means that an adversary with valid credentials cannot find out while performing a secret handshake if the other party belongs to a group where he is not a member of.
- **Anonymity**: The identity of the users applying the secret handshake is not disclosed until the protocol was finished and both parties could match the received credentials with their matching references. The parties performing the secret handshake stay anonymous until the protocol was finished with a successful match on both sides. In cases of unfinished protocols or unsuccessful matches on one or both sides, both parties have no knowledge about the other parties' identity or properties.

Table I
IS OPERATIONS

```
Property Certification Pattern ← CA
Secure Match Pattern ← SMP
Credential Revocation Pattern ← CRP
Mutual Key PoK Pattern ← MKPP
SMP ← roles: User1 and User2
CRP ← roles: User1 and User2
MKPP ← roles: User1 and User2
—— Initialization phase ——
CA.certifyProperty(in:Identity1, in:Context1, out:PropCredential1);
CA.certifyProperty(in:Identity2, in:Context2, out:PropCredential2);
CA.provideMatchingValues(in:Identity1, in:Context1, out:MatchValue1[]);
CA.provideMatchingValues(in:Identity2, in:Context2, out:MatchValue2[]);

—— Matching phase ——
User1.SMP.initiate(out:State1);
User2.SMP.hideCredential(in:PropCredential2, in:State1,
    out:HiddenCredential2, out:State2);
User1.SMP.hideCredential(in:PropCredential1, in:State2,
    out:HiddenCredential1, out:State3);
User1.CRP.getRevocationList(out:RevocationList);
User1.CRP.isValid(in:HiddenCredential2, in:RevocationList,
    out:BooleanResult1);
if (BooleanResult1 == true) {
    User1.SMP.match(in:PropCredential1, in:HiddenCredential2,
        in:MatchValue1[], out:Key1);
}
User2.CRP.getRevocationList(out:RevocationList);
User2.CRP.isValid(in:HiddenCredential1, in:RevocationList,
    out:BooleanResult2);
if (BooleanResult2 == true) {
    User2.SMP.match(in:PropCredential2, in:HiddenCredential1,
        in:MatchValue2[], out:Key2);
}
if (BooleanResult1 == true && BooleanResult2 == true) {
    User1.MKPP.challenge(in:Key1, in:State1, out:Challenge1)
    User2.MKPP.challenge(in:Key2, in:Challenge1, out:Challenge2)
    User1.MKPP.verify(in:Key1, in:Challenge2, out:ProveResult1)
    User1.MKPP.challenge(in:Key1, in:Challenge2, out:Challenge3)
    User2.MKPP.verify(in:Key2, in:Challenge3, out:ProveResult2)
}
```

- **Resistance to replay attacks**: An adversary cannot successfully perform a secret handshake by repeating parts of the protocol that was eavesdropped from a successful handshake between two parties. Nor he can learn something about the identity or properties of the other party.
- **Revocation support** (optional): Property credentials can be *reactively* revoked when needed by adding them to a public available revocation list. *Proactive* revocation is possible with the help of time-bound credentials.

All listed features can be proved formally under the assumption that the Bilinear Decisional Diffie-Hellman (BDDH) problem is hard (details in [6]).

To provide the complex solution all pre-conditions defined in the embedded security patterns have to be considered before the IS can be applied.

VII. Analysis of SERENITY Security Patterns

While applying the SERENITY methodology to the new area of cryptographic protocols, some advantages but also some weaknesses of the pattern approach became evident.

The template used for the SERENITY security patterns allows to describe the security solution in an structured way, identifying the applying entities, their relationships and interactions. This goes far beyond the usual used natural language description, applied e.g., in [36] or [37]. Especially the description of the pattern solution as sequential flow of operations is more explicit and closer to a possible implementation. Therefore it is easier understandable for application developers used to read and interpret design patterns. The operations are the link to the other artefacts of the SERENITY methodology, like *Security Pattern Implementation* and *Executable Components*, which describe the detailed technical solution and provide the executable code.

In our security patterns, an operation consists often of two types of activities:

- **computation activities** like cryptographic computations or policy evaluations that are executed by the entity applying the pattern,
- **communication activities** where input (like parameters) is received from a partner entity or where the result of the computation is sent out.

A good example is the operation *certifyProperty* in the pattern *Property certification*. The CA applying the pattern receives the application context and identity from the user who wants to get a property certified. After successful validation of the user's context, the CA sends the property credential back to this user.

This shows an improvement possibility of the SERENITY pattern approach. In the pattern description, only the applying actor is described. This is not sufficient, when protocols, simple interactions or data exchanges are part of the security solution. In this paper, we decided to use UML sequence diagrams to correlate the pattern operations with the interactions of the different entities.

We could identify a related issue in the integration scheme, describing the security solution as composition of several security patterns. The IS operational flow (see Table I) is not descriptive enough to clearly describe the information exchange between the involved entity and needs more details than a simple sequence of operations. The operations define a number of input and output parameters, but there is no means to differentiate between a simple in-/output that is internal to the applying entities or an information exchange with another entity.

Standard modeling languages, like UML or UMLSec, can be used to describe the complex data flow and interactions in the protocols. This has also the advantage that these languages are well-known by application developers and the diagrams, e.g., sequence diagrams, are easy to understand. Information about activities that can be done in parallel, repeated, omitted or those relying on secure channels can be easily added.

We propose a *conceptional model* (see Figures 4 and 5) to visualize the data flow in the security solution. It is similar to an UML activity diagram and shows the involved entities applying the different security patterns and pattern operations as well as the exchanged information.

Another weakness of the security pattern approach is the pattern selection process. Can an application developer or architect responsible to implement one of the scenarios from Section IV identify the right security properties from the use case description or a specification? Security properties like 'Mutual Authentication' with 'Untraceability' have a meaning only for security experts. Here, we see the need to express the security requirements, security properties and features in a way non-security experts understand.

The problem of the definition of security properties and features is two-edged. The increasing number of security solutions as patterns demands a detailed classification of the security properties and features in order to be able to select the right pattern for a given problem. On the other hand, we have the non-security expert confronted with a huge set of terms related to security. In order to enable these people to use the security properties to select the right pattern, we need a clear, unique and understandable definition of each of them. There should be a catalogue of security properties, which allows with the help of a taxonomy to navigate in the property space.

VIII. Conclusion

In this paper, we described the Untraceable Secret Handshake protocol as SERENITY security pattern. To our knowledge, was this one of the first attempts to describe cryptographic schemes and protocols as formal security patterns

([1] provided the first definition of the Secret Handshake pattern).

In spite of the discovered weakness of the security pattern approach in general, the abstract description of the secret handshake protocols independent of the real algorithms helps to spread the knowledge in the security community and also among application developers and architects. Secret handshakes and secure matching are in general difficult to understand for non-cryptographic experts. The security pattern approach makes them available for people without deep cryptographic knowledge.

Following the SERENITY methodology, each security pattern has one or more possible implementations (*Security Pattern Implementations* and *Executable Components*), which give concrete examples of the security solution in different environments and show their feasibility.

In our case, we implemented two of the scenarios described in Section IV. A first solution is based on the simulation framework Ptolemy II [38] and implements the *Incompatible chemicals* scenario. The simulated sensor nodes are initialized with the properties credentials for the liquid in the attached drum and the matching value for all reactive liquids. When a sensor intersects the wireless range of a second sensor, the sensors perform the secret handshake and create an alert when necessary.

The second solution is a Java implementation of the *Secret Interest Groups* in the framework for Facebook motivated by its extreme popularity. The intended use is the following: a SIG member runs the java http proxy, which intercepts only requests toward Facebook servers. The proxy modifies requests and responses, running the secret handshake protocol upon membership invitation and chat events; notification of the success or failure of the protocol is provided to the user through modifications of the html that is displayed in the browser. Further details about the implementation are in [35].

Additionally, we developed in the context of the WASP project [39] a solution based on web services to mutually authenticate mobile Wireless Sensor Network gateways with different backend applications.

The three different implementations proved the feasibility of the developed security pattern for secret handshakes. Despite the different development environments, it was possible to follow the SERENITY approach. In each prototype, we could identify the described security patterns and their interactions.

The different developed solutions show also that mutual authentication with untraceability and attacker resistance is a recurring security problem that arises in different contexts. The security patterns captured in this paper represent a well-proven solution based on the cryptographic schemes of secret handshakes.

References

[1] A. Cuevas, P. El Khoury, L. Gomez, A. Laube, and A. Sorniotti, "A Security Pattern for Untraceable Secret Handshakes," in *Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE '09*, Mar. 2009.

[2] P. El Khoury, M.-S. Hacid, S. S. Kumar, and E. Coquery, *A Study on Recent Trends on Integration of Security Mechanisms*, Mar. 2009, ch. Advances in Data Management, special volume of Studies in Computational Intelligence.

[3] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns : Integrating Security and Systems Engineering (Wiley Software Patterns Series)*. John Wiley & Sons, March 2006.

[4] G. Spanoudakis, A. Mana Gomez, and K. Spyros, Eds., *Security and Dependability for Ambient Intelligence*. Series: Advances in Information Security , Vol. 55, Springer, April 2009.

[5] D. Balfanz, G. Durfee, N. Shankar, D. K. Smetters, J. Staddon, and H.-C. Wong, "Secret handshakes from pairing-based key agreements." in *IEEE Symposium on Security and Privacy*, 2003, pp. 180–196.

[6] A. Sorniotti and R. Molva, "A Provably Secure Secret Handshake with Dynamic Controlled Matching," *Proc. of 24th International Information Security Conference (IFIP SEC)*, 2009.

[7] A. Mana, C. Rodolph, G. Spanoudakis, V. Lotz, F. Massacci, M. Molideo, and J. S. Lopez-Cobo, *Security Engineering for Ambient Intelligence: A Manifesto*. IGI Publishing, 2007.

[8] D. Vergnaud, "RSA-Based Secret Handshakes," in *WCC*, 2005, pp. 252–274.

[9] C. Castelluccia, S. Jarecki, and G. Tsudik, "Secret handshakes from ca-oblivious encryption," in *ASIACRYPT*, 2004, pp. 293–307.

[10] C. Meadows, "A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party," *Security and Privacy, IEEE Symposium on*, p. 134, 1986.

[11] J.-H. Hoepman, "Private handshakes." in *ESAS*, ser. Lecture Notes in Computer Science, F. Stajano, C. Meadows, S. Capkun, and T. Moore, Eds., vol. 4572. Springer, 2007, pp. 31–42.

[12] W. Diffie and M. Helman, "New directions in cryptography," *IEEE Transactions on Information Society*, vol. 22, no. 6, pp. 644–654, november 1976.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[14] S. Xu and M. Yung, "k-anonymous secret handshakes with reusable credentials," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2004, pp. 158–167.

[15] S. Jarecki, J. Kim, and G. Tsudik, "Beyond secret handshakes: Affiliation-hiding authenticated key exchange," in *CT-RSA*, 2008, pp. 352–369.

[16] G. Ateniese, M. Blanton, and J. Kirsch, "Secret handshakes with dynamic and fuzzy matching," in *Network and Distributed System Security Symposuim*. The Internet Society, 02 2007, pp. 159–177, cERIAS TR 2007-24.

[17] S. Jarecki and X. Liu, "Unlinkable secret handshakes and key-private group key management schemes," in *ACNS'07*, 2007, pp. 270–287.

[18] A. Sorniotti and R. Molva, "Secret handshakes with revocation support," in *ICISC 2009, 12th International Conference on Information Security and Cryptology, Seoul, Korea*, 12 2009.

[19] N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey on security patterns," *Progress in Informatics*, vol. No. 5, pp. 35–47, 2008.

[20] J. Yoder and J. Barcalow, "Architectural Patterns for Enabling Application Security," in *In Proc. of PLoP'97*, 1997.

[21] S. Romanosky, Ed., *Security Design Patterns*, 2001.

[22] M. Schumacher, *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[23] E. Fernandez and R. Pan, "A Pattern Language for Security Models," in *In Proc. of PLoP'01*, 2001.

[24] E. B. Fernández, J. Jürjens, N. Yoshioka, and H. Washizaki, "Incorporating database systems into a secure software development methodology," *19th International Workshop on Database and Expert Systems Applications*, pp. 310–314, 1-5 September 2008, Turin, Italy.

[25] L. Compagna, P. El Khoury, F. Massacci, R. Thomas, and N. Zannone, "How to capture, model, and verify the knowledge of legal, security, and privacy experts: a pattern-based approach," in *ICAIL*, 2007, pp. 149–153.

[26] F. Sanchez-Cid, A. Munoz, P. El Khoury, and L. Compagna, "XACML as a Security and Dependability (S&D) pattern for Access Control in AmI environments," in *Ambient Intelligence Developments - AmI.d*, Sep. 2007.

[27] F. Sanchez-Cid and A. Mana, "Patterns for automated management of security and dependability solutions." *1st International Workshop on Secure systems methodologies using patterns (SPattern'07)*, 2007.

[28] A. Benameur, P. El Khoury, M. Seguran, and S. K. Sinha, "Serenity in e-business and smart items scenarios," *Security and Dependability for Ambient Intelligence, Series: Advances in Information Security*, vol. Vol. 55, 2009.

[29] A. Cuevas, P. El Khoury, L. Gomez, and A. Laube, "Security patterns for capturing encryption-based access control to sensor data," *The Second International Conference on Emerging Security*, 2008.

[30] P. Busnel, P. El Khoury, S. Giroux, and K. Li, "Achieving socio-technical confidentiality using security pattern in smart homes," *The Third International Symposium on Smart Home*, 2008.

[31] Europol, Eurojust, T. Van Cangh, and A. Boujraf, "Wp3-cs2: The Eurojust-Europol case study," *at http://www.r4egov.eu/resources*, 2007.

[32] [Online]. Available: http://www.europol.europa.eu/

[33] COBIS Consortium, "COBIS. FP STREP Project IST 004270," 2005. [Online]. Available: www.cobis-online.de

[34] [Online]. Available: http://chris.pirillo.com/pownce-social-networks-arent-identity-networks/

[35] A. Sorniotti and R. Molva, "Secret interest groups (SIGs) in social networks with an implementation on Facebook," in *SAC 2010, 25th ACM Symposium On Applied Computing, March 22-26, 2010, Sierre, Switzerland*, 2010.

[36] S. Lehtonen and J. Pärssinen, "A pattern language for cryptographic key management," in *EuroPLoP*, 2002.

[37] C. Steel, N. Ramesh, and L. Ray, *Core Security Patterns: Practices and Strategies for J2EE, Web Services, and Identity Management*. Upper Saddle River: Prentice Hall PTR, 2005.

[38] C. Brooks, E. Lee, X. Liu, S. Neuendorffer, H. Zheng, and Y. Zhao, "Introduction to Ptolemy II," in *UCB/ERL M05/21 Heterogeneous concurrent modeling and design in Java*. University of California at Berkeley, 2004, vol. 1.

[39] WASP, "WASP (Wirelessly Accessible Sensor Populations), IST 034963," 2006. [Online]. Available: www.wasp-project.org

www.iariajournals.org

International Journal On Advances in Intelligent Systems

➪ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

➪issn: 1942-2679

International Journal On Advances in Internet Technology

➪ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING

➪issn: 1942-2652

International Journal On Advances in Life Sciences

➪eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO

➪issn: 1942-2660

International Journal On Advances in Networks and Services

➪ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION

➪issn: 1942-2644

International Journal On Advances in Security

➪ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

➪issn: 1942-2636

International Journal On Advances in Software

➪ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

➪issn: 1942-2628

International Journal On Advances in Systems and Measurements

➪ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL

➪issn: 1942-261x

International Journal On Advances in Telecommunications

➪AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA

➪issn: 1942-2601