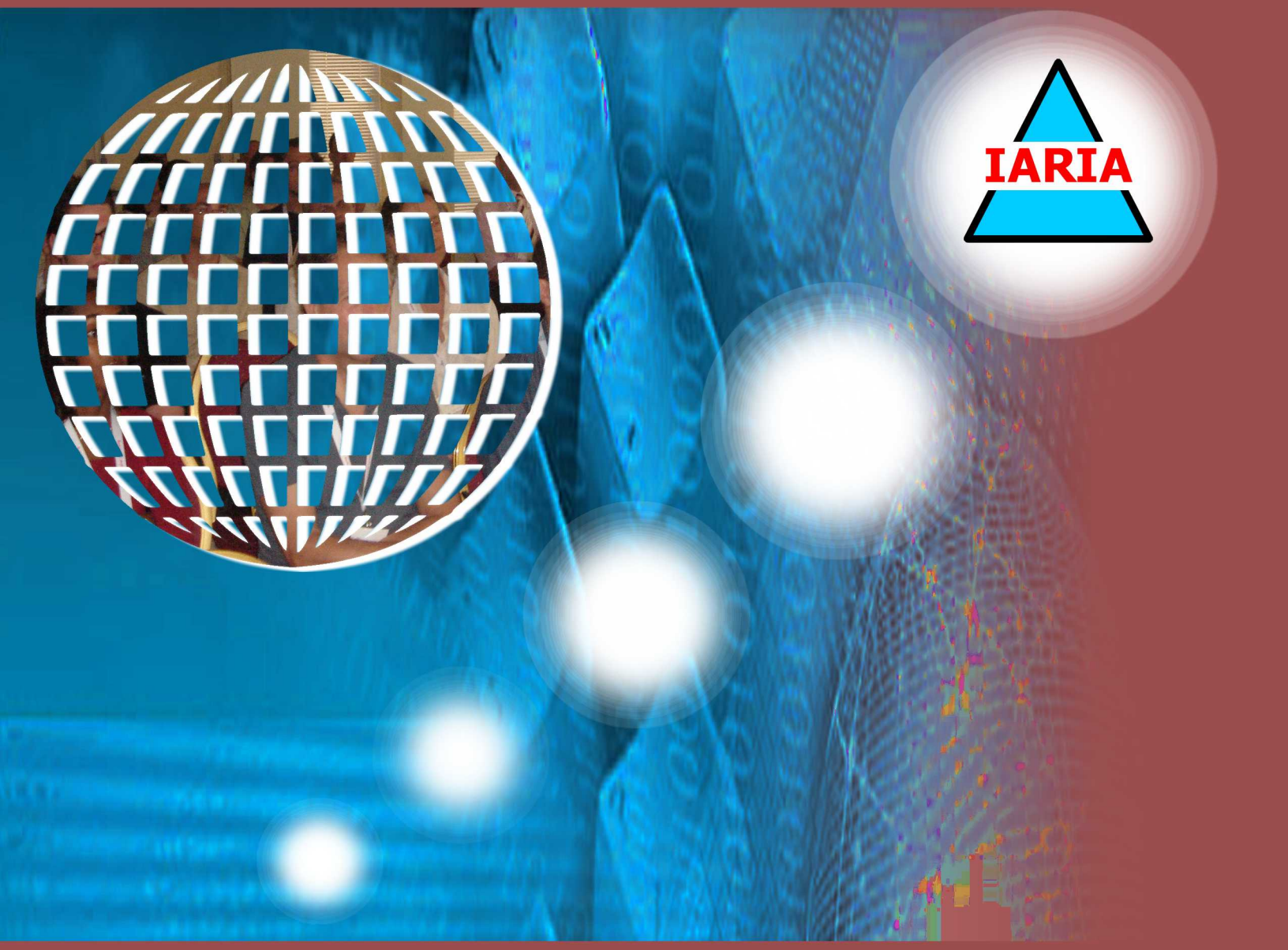
International Journal on Advances in Security



2011 vol. 4 nr. 3&4

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

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Security
Volume 4, Number 3 & 4, 2011

- Bjarne E. Helvik, The Norwegian University of Science and Technology (NTNU) – Trondheim, Norway
- 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 4, Numbers 3 & 4, 2011

CONTENTS

Universal Bluetooth™ Access Control and Security System

Francisco José Bellido-Outeiriño[(1)]
José Luis de la Cruz Fernández [(2)]
[(1)]Dept. of Computers Architecture, Electronics and Electronics Technology;
University of Cordoba
14071 Cordoba, Spain
fjbellido@uco.es; fa1crfej@uco.es;

Antonio Moreno-Muñoz[(1)]
Pedro M. Canales Aranda[(1)],Benito Pérez Jarauta[(3)]
[(2)]Dept. of Applied Physics;
[(3)] Stop Casa Segura S.L.
University of Cordoba
14071 Cordoba, Spain
amoreno@uco.es; p52caarp@gmail.com;
jarauta_seguridad@hotmail.com

Abstract— **In this paper, we describe the use of Bluetooth™ technology for the development of novel applications for home and intelligent buildings scenarios, as well as scenarios in which security and access control is necessary (e.g., garages). The design of the solution has been done following a hierarchical scheme, starting from a simple and functional module and providing new features that will give an added value to the final product. This methodology allows to generate a wide range of products with several features for the final consumer and end users. The developed system is low cost, autonomous, scalable (both from a physical point of view and functionality) and capable of interacting and being controlled by the end user and by other similar modules in range. Establishing communication with the system is quite simple as well as cost effective, since the universal key for these access points can be managed by user's mobile phone or PDAs. The solution has been implemented using Java. In addition, the developed software is offered in order to provide a friendly interface and to allow the system administrator to manage the control unit (e.g., edition of authenticated users, download record files of event, data sharing with other modules in range, etc.), giving an added value to Bluetooth™ system and to the mobile devices integrating this technology.**

Keywords - Bluetooth™, Home Automation & Control, WLANs e-Key, Security.

I. INTRODUCTION

Wireless communication has been a great quantitative and qualitative jump in information management, allowing access and interchange without a physical wire connection [1]. Wireless transmission of voice and data has a continuous evolution into new standards, like Bluetooth™ [2], Wibree™ [2] or Zigbee™ [3].

These newest wireless technologies are focused on communication systems of short-medium range and are optimized so as to be low cost and with low power consumption, to be integrated in mobile devices or embedded systems [4]. Hence, they have been established as the future technology for mesh nets or distributed acquisition and control systems.

The concepts of network embedded systems stem from hierarchically interconnected networks, which are based on tiered architectures that provide cost-effectiveness and scalability, and adapt straightforwardly to various application requirements [5], regarding security and users' aspects in the overall system design [6].

In the field of home automation, security or access control systems with wireless technologies, several interesting applications can be found. In figure [7] it is presented a Zigbee™'s based system for a digital door lock and in [8], [9], [10] the use of Bluetooth™ for home security and/or monitoring purposes is shown.

As to the applications that use Bluetooth™ (via the mobile phone or PDA) to perform control actions to third parties, we can find several software applications that allow the mobile phone to be used as a remote control for the PC (e.g., to manage Windows Media Player or Power Point presentations). Many of them are even freeware or shareware.

These tools, usually Windows-based, communicate with the application layer of Bluetooth™ systems using DLLs drive via USB dongles. Although the proposed solutions consider e-keys for user authentication when the e-key is being managed by mobile phone or PDA devices, the authentication process is carried out in each session by running an autonomous microcontroller unit, in which the management of the lower levels of the Bluetooth™ stack takes place, and not by back-end computers, which are LAN connected to the Bluetooth™ device and use predefined profiles for phone-metric recognition, some of them are similar to USB encrypted keys.

In this paper, we describe an interesting application of Bluetooth™ technology to access control systems, which is suitable for home and building applications and extendable to any security system or controlled access points such as parking garages. The proposed system, based on [1], has been designed following a hierarchical scheme, starting from

a simple but fully functional module and new features have been included in order to build a final product with an added value, as well as to offer a wide range of products with several features to the final customer or end user [5].

The developed system is low cost, autonomous, physically and functionally scalable and with control and interactive capabilities.

The main problem is that the system described in this paper has implemented an autonomous Bluetooth™ module by managing the stack and profiles at lower levels, with no computer or operating system requirements. The system could work, with or without additional software applications, for users and system administrators, in which higher security levels for access and control applications are implemented.

In addition to these or either to traditional RF devices for remote control access, we present a novel application with great potential, a step further due to Bluetooth™ use. These characteristics allow us to offer a very simple, powerful and cost effective means to manage the identification of users and to perform subsequent actions.

The paper is organized as follows: in the next section**,** an analysis of the challenges and requirements of the proposed system is presented and discussed and, in Section III, a full description of its structure and implementation is done. Finally, we discuss the advantages and applications of the developed system in some real scenarios.

II. CHALLENGES AND REQUIREMENTS

The main objective of our work has been to design and to build a universal access control and monitoring system, bearing in mind such requirements as power consumption, range, cost, network capabilities, and a standardized technology [4].

In the field of WPANs technologies, we find several possibilities: Bluetooth™ and Zigbee™. They are considered to be the optimal ones; RFID [11] could also be fit for this purpose. All of these technologies comply with the necessary features for the development of security control and identification systems, but they also show some disadvantages for this kind of applications.

For example, RFID tags work in different frequency bands; some of them are not allowed worldwide. There are also several standards for the Tags, like EPCGen 2 (Electronic Product Code consortium) [12] that are not adopted by all the companies. Therefore, a RFID based system could not guarantee a universal or worldwide functional system.

On the other hand, Zigbee™ and Bluetooth™ operate in the ISM band, and both standards have interoperability features. Then, according to our criteria, Zibgee and Bluetooth™ are the ones selected in this first analysis.

An analysis of both technologies in depth about the requirements of the application shows that Bluetooth™ is the optimal solution, since nowadays most of mobile devices (mobile phones, PDAs, laptops, etc.) support Bluetooth™, most people have at least one mobile phone, and it is a technology commonly used by mobile phone users. This means that the communication scheme between users and the access point, that is to say the user's terminal, is carried out without additional cost and without the need to carry new keys, ID-cards or remote control devices.

Since Bluetooth™ is a standard we could think that any digital device equipped with it can communicate with other devices. However, this is not completely true because Bluetooth™ Specifications [2] define several Profiles for different applications, and not all the Profiles are implemented or supported by all devices, e.g., in mobile phones it is usual to have only File Exchange and Audio capabilities, but it is not possible to manage the Serial Port with a Bluetooth™ adapter like in a desktop that supports the full stack run in a PC.

Therefore, it is necessary to design a *Control and Access Point Terminal* system that could be detected by any other device with Bluetooth™ capabilities. The solution proposed in this paper, as described in the next section, is to enable a minimum set of profiles to cover at least those profiles implemented in any electronic device equipped with Bluetooth™, thus all devices would be able to detect the Terminal by means of one of these profiles.

III. SYSTEM DESCRIPTION

In this section**,** we present the basis of the proposed system, we place special emphasis on some fundamental considerations about the Bluetooth™ profiles, and we propose the structure and architecture of the system and perform the system implementation.

First, we will establish the implementation scope of this system within the framework of access control. An access control could be defined as a system applied to an access that ensures the identification of users and authorizes them or not to enter, on the basis of a set of stored information.

A basic classification of access control systems could be as follows:

- According to connectivity
  - Off-line
  - On-line
- According to power source
  - Battery powered
  - Mains powered
  - Autonomous (power source located in the key)
- According to the kind of reading
  - Insertion (mechanical/classical key)
  - Contact (magnet key, chip, RFID)
  - Remote (remote control, mobile phone)
- According to security level
  - Level I. Something we have (e.g., a key)
  - Level II. Something we have plus something we know (e.g., a key+ PIN)
  - Level III. Something we have plus something we know plus something we only have (key + PIN+ biometric features).

The system proposed is off-line (on-line supported), mains powered, remote keyless (contact is made through RFID antennas) and it supports any security level: I, II or III (L-III interconnected to a biometric reader).

A. How the Bluetooth™ Access Point Terminal works

A way to achieve that any other device with Bluetooth™ capabilities could find the system is by enabling both Hands Free Profile and Serial Port Profile, so that any device could discover the Terminal.

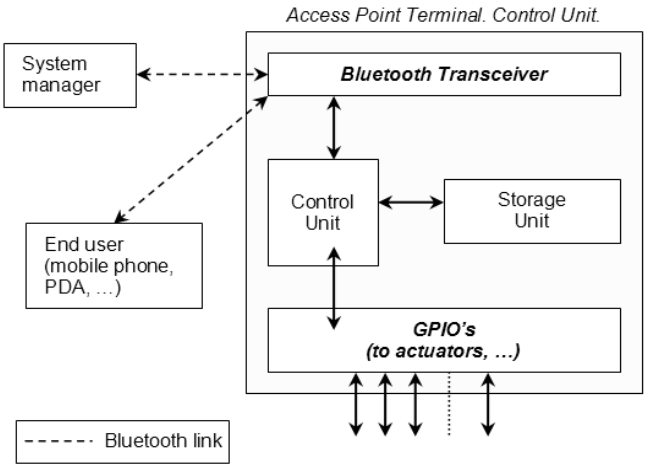


Figure 1. Main components of the system architecture.

The device appears to the end user as a hands free unit, and thus establishing communication with it (e.g., to try to be identified to open a door automatically) is very simple and inexpensive as *universal key* for these access points can be the user's own mobile phone or PDA. Thus, with his own personal device (i.e., the mobile phone) the user can gain entry to several places (parking door lock, clocking-in at work, enabling/disabling home alarm system, etc).

Figure 1 shows the main components of the system architecture.

The system core is the so-called "*Access Point Terminal*". It could be configured in three ways, depending on the application:

1. *Basis application*: it stores the data of authenticated users (preloaded). Unencrypted information and operation mode.
2. *Advanced application*: it stores encrypted data by means of which it recognizes authenticated users even when they have a different and unique encrypted key based on unique parameters of the own user.
3. *Net advanced application*: as the one above, except that it sends the information of events to other *Terminals* in range (Figure 2).

For security reasons, the *Terminal* is normally in a semi sleep mode, reading periodically the input channel for any request. If the Terminal receives information, both in operation modes (1) and (2) (mode 3 is formally like 2), it checks the user status and performs the operation and it carries out other general actions like storing the event information or sending the event information (if operation mode is 3).

For a better robustness, if the same MAC Bluetooth tries to connect several times and it proves to be an unauthorized user, this MAC is put into a filter table that is used in the first steps to avoid service denial due to the presence of sniffers.

The brain of the *Access Point Terminal* is the Control Unit that manages the communication and control through the following components:

- Bluetooth transceiver, in charge of sending/receiving the user's requests.
- Storage unit, which reads/writes data or updates firmware.
- Power subsystem, in charge of checking the auxiliary battery charging process, when needed.
- Inputs, like 'open door' relays or some others similar depending on the controlled item.
- Outputs: actuators, relays, etc.

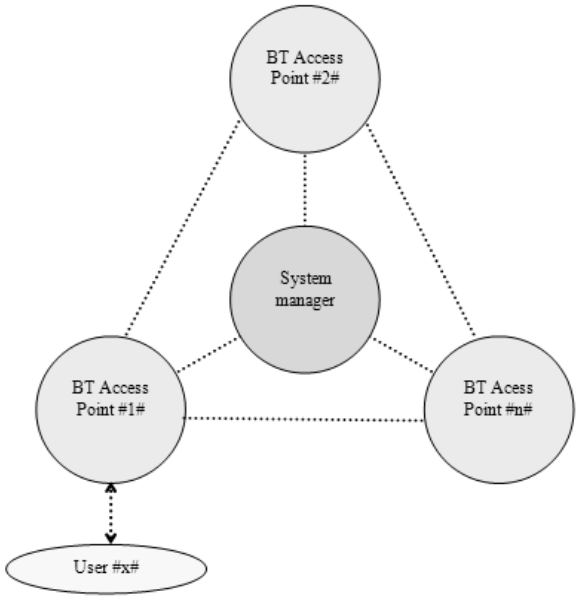


Figure 2. Networking scheme for data interchange.

The end user operation is quite simple. As to the basis application (see left side column, item "1"), it is only necessary to search for a Bluetooth device and once it is discovered, the user tries to connect to it (i.e. Garage#1). It will appear as a Headset or Hands Free device in the users mobile. If the user data is preloaded in the Application Point Terminal (see Figure 1) then the subsequent action will be done.

In respect to the advanced operation mode, which is explained in detail in Section F, the administrator must previously discharge the user from the system (only the first time), using the web server application developed for this purpose.

This "signing up" operation will download a simple MIDLet application which will generate the unique and encrypted key for this mobile phone in particular –taking some data out of it so as to generate the key, and only for a set of predetermined gates or terminals.

Once the key is installed in the mobile phone or PDA then the associated application is run and it transparently sends the users' encrypted key code to the Terminal.

The Net advanced application mode is similar to the Advanced application, except that in this mode the Terminals in range have the capacity for sending and receiving the events between each others. Thus all the Terminals connected could know if user "#x" has entered in the garage, and the "#door#" s/he used. This functionality allows to implement easily anti-passback strategies in multiple gate installations, or to filter users by their trajectory, etc.

As previously pointed out, the Access Point Terminal works normally off-line, and supports on-line connections as well.

Figure 3. Typical scenario of application

In the off-line operation, for downloading the event log or updating the firmware the user is required to connect to the system with administrator privileges; this is called System Manager in Figure 1 and Figure 2.

The System Manager consists of a Bluetooth connection to the Terminal which runs the manager software tool (which will be described in Section D.). For this purpose, only a SmartPhone, PDA or mobile phone with Windows Mobile and GPRS/3G connection is needed. Moreover, off-line updating done by the System Manager is carried out in three steps:

i. Download the file for uploading the Terminal from the Web Application Tool
ii. Connect to the Terminal, download the .log file and upload (if needed) the new configuration file.
iii. Reload, from the Web Application Tool, the log file obtained from the previous step and store it in the appropriate Terminal workspace.

B. Bluetooth™ profiles. Practical considerations

In the Bluetooth™ Specification [2], two types of links to support applications of voice and information are defined: an asynchronous link without connection (ACL, Asynchronous ConnectionLess) and a synchronous link orientated to connection (SCO, Synchronous Connection Oriented).

The ACL links support traffic of information without any guarantee of delivery; the transmitted information might be user information or control information.

The SCO links support voice in real time and multimedia traffic, using a reserved bandwidth. Both the voice and the information are transmitted in packages and the Bluetooth™ Specification allows implementing ACL and SCO links simultaneously. The asynchronous channel supports symmetrical and asymmetric communication. In the asymmetric communication 723.3 Kb/s can be sent from the server and 57.6 Kb/s towards the server, whereas in the symmetrical communication 433 Kb/s are sent in both directions.

In order for the system to include the expected functionality, we have implemented both selected profiles: Serial Port and Headset (ACL and SCO links). Thus, any electronic device with Bluetooth™ is able to discover the Bluetooth™ Access Control system.

The Bluetooth™ standard was created to be used by a great number of manufacturers and to be implemented in unlimited areas. To assure that all the devices that use Bluetooth™ would be compatible among them, standard schemes of communication are necessary. Therefore, different profiles have been defined considering several user communication models.

A profile defines a selection of messages and procedures of the Bluetooth™ Specification and it offers a clear description of the air interface for specific services. A profile can be described as a complete section of the stack of protocols, as shown in Figure 4.

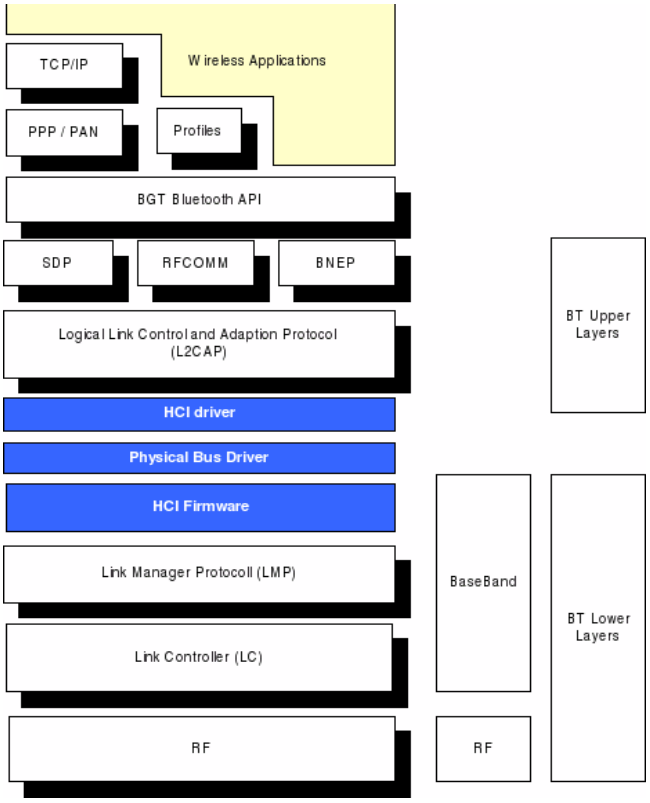


Figure 4. Bluetooth™ protocol stack [12]

There are four general definite profiles, on which some of the most important models of use and its profiles are directly based. These four models are Generic Access Profile (GAP), Serial Port Profile (SPP), Service Discovery and Application Profile (SDAP) and Generic Object Exchange Profile (GOEP), shown in Figure 5.

The Generic Access Profile (GAP) defines the general procedures for connection discovery and establishment between Bluetooth™ devices. The GAP handles the discovery and establishment between units that are not connected and ensures that any couple of Bluetooth units, no matter its manufacturer or application, could interchange information via Bluetooth ™ and discover the type of applications which support each unit. Furthermore, there are defined procedures related to the use of the different security levels.

The Serial Port Profile (SPP) defines the necessary requirements for Bluetooth™ devices to establish a connection of emulated serial cable between two similar devices using the RFCOMM protocol. This profile only requires support for one-slot packages. This means that information rates are the highest rates. RFCOMM is used to transport the user information, modem control signals and other configuration commands.

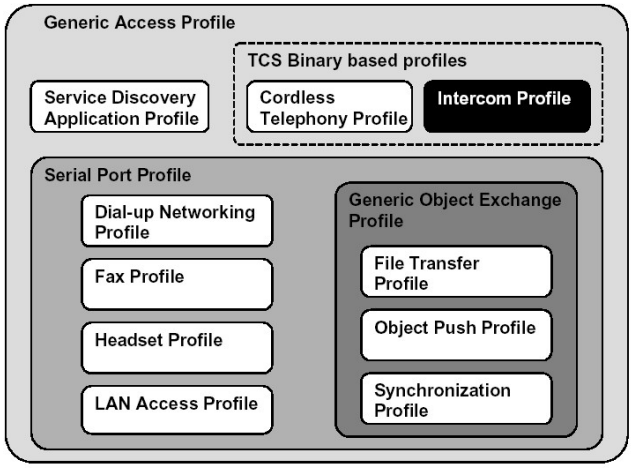


Figure 5. Bluetooth™ profiles –reduced set [2].

The Serial Port Profile is derived from the Generic Access Profile, therefore all the obligatory requirements of the Bluetooth specification applied to that profile are also applicable to the Serial Port Profile, and likewise those requirements defined as optional for the profile Generic Access Profile are also optional for the Serial Port Profile [2].

The Service Discovery Application Profile (SDAP) defines the protocols and procedures of an application in a device Bluetooth in which someone might want to discover and to recover information related to the services located in other devices. The SDAP is dependent on the GAP.

TABLE I. BLUETOOTH™ PROFILES RFCOMM UUIDS [2].

Profile Name	UUID
Serial Port	1101
LAN Access Using PPP	1102
Dialup Networking	1103
IrMC Sync	1104
OBEX Object Push	1105
OBEX File Transfer	1106
IrMCSyncCommand	1107
Headset	1108
Cordless Telephony	1109
Intercom	1110
Fax	1111
Audio Gateway	1112
WAP	1113
WAP_CLIENT	1114

The Bluetooth protocols stack contains the Service Discovery Profile (SDP), which is used to locate available services for devices that are found inside the environment or that are in range of some other devices. As soon as one has located the available services in one o more neighbouring devices, the user can choose one for his/her use. The selection, access and use of a service is the aim of this

profile. Though the SDP protocol is not directly involved in the procedure of access to a service, the information obtained through it facilitates the access to the above-mentioned service.

The Generic Object Exchange Profile (GOEP) defines the protocols and procedures used by applications to offer characteristics of exchange objects. The uses can be, for example, synchronization, transference of files or Object Push model. The most common devices that use this model are Personal Digital Assistants (PDAs) and mobile phones. The GOEP derives from the serial port profile.

We only have made a reduced list of them since our purpose was only to make an introduction and justification of the selected ones.

Table I shows the RFCOMM UUIDs codes of the Bluetooth™ Specifications considered.

Another issue is to set the device class code**,** which will be sent in subsequent inquiry responses. As observed in Table II, the device class code consists of a 6 digit hexadecimal derived number as defined in section "1.2 The Class of Device/Service Field" of the Bluetooth™ specification "Bluetooth™ Assigned Numbers" [2].

TABLE II. DEVICE CLASS CODE ASSIGNED TO HEADSET PROFILE [2].

Code (Hex)	Name	Major Service	Major Device	Minor Device
200404	Headset	Audio	Audio	Headset

Lower layers set up the SCO channel, and as soon as a SCO link is established, the following response is asynchronously sent to the host. It is very important to configure the three SCO channels correctly so as to support any BT audio device connection.

The user's security authentication and other related aspects need no further consideration since they are supported by all devices and are not a main source of problems. Final application in each case will force us to select the right security level among the possibilities that Bluetooth™ offers.

C. *Other practical considerations for the design*

Among the different alternatives of implementation, the adapters and the integrated modules, according to a two-processor model, are especially more adapted to the development of application systems with Bluetooth™ capabilities.

The main characteristics of this kind of systems will be exposed and the practical considerations will be taken into account before approaching their development. In Figure 6, it is shown a detailed graph of a model of a complete two-processor system based on the Bluetooth ™ technology [15].

In Figure 6, it can be observed that there are two main areas: the upper one or "Bluetooth Host" and the lower one or "Bluetooth Module". The top area or Host would contain the hardware/software support for the application that manages the communication with the Bluetooth ™ module, independently of the supported platform, and it also implements the higher layers of the Bluetooth protocol (upper central area with a white background). The lower area is the acquired Bluetooth™ module, that is, the one that physically will perform the wireless transmission tasks.

Therefore, we have to work on the development of the top layer, which includes the application, protocols of the top levels of the Bluetooth ™ Specification and the HCI driver (Host Controller Interface), by means of which Bluetooth™ manages the communication with the transceiver module (where the lower layers of the protocol are located).

In this area mentioned above, we also find the protocols (Figure 6, central area). The protocols are only a series of functions, procedures and commands with a given functionality established by the Bluetooth™ Specification. The manufacturers generally provide the protocols previously mentioned when the Bluetooth ™ module is acquired.

Thus it is necessary to develop an application that, by means of the use of the specification protocols [8], allows us to use the Bluetooth ™ technology like any other wireless communication as well as to implement the physical integration of the system previously mentioned with the Bluetooth™ transceiver module.

D. *Programming languages*

As to the application, it is necessary to develop the protocol profiles adapted to the functionality of the complete equipment. A profile defines a selection of messages and procedures of the Bluetooth ™ Specification and offers a clear description of specific services of the air interface.

A profile can simply be described as a complete section of the protocols stack.

When choosing the programming language, by means of which the application will be developed, it is very important to know the requirements of the environment on which it will be executed, and the main features are exposed in the following epigraph.

In relation to the real possibilities of implementation of both the protocols and the profiles, there is no restriction or limitation. Any programming language, of high and low level, is capable of supporting the collection of routines, functions and procedures that are established by the Bluetooth ™ Specification. This task will be more or less simple, obviously, depending on the chosen language.

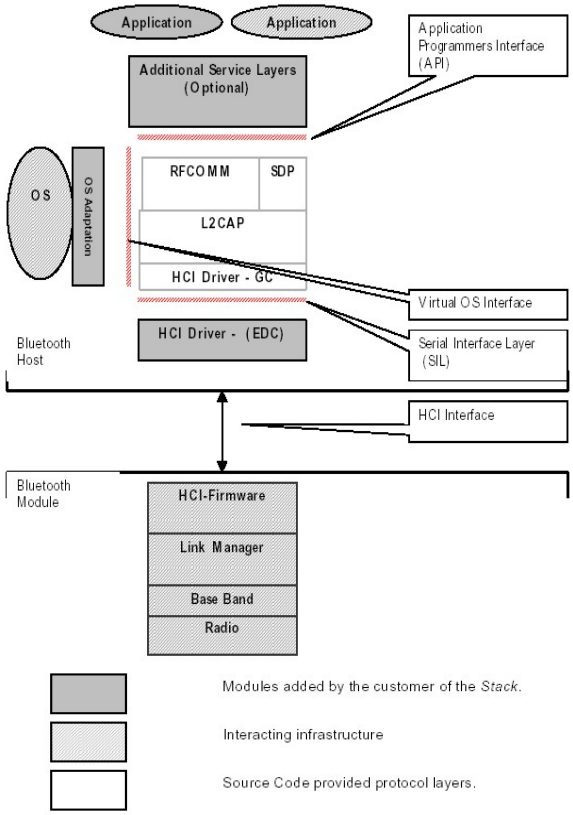


Figure 6. Diagram of a model of a complete two processor system [2]

At this point, it is important to remember that the Bluetooth™ Specification includes hardware, software and interoperability requirements. Therefore, the development of the process of protocols and profiles cannot be limited to a partial development of a running application, because they must comply with all the characteristics imposed by the norm, so in order to be qualified as Bluetooth™, the software must overcome the corresponding procedure of certification (Bluetooth Qualification Program).

Nevertheless, there are two languages that are especially better placed than the latter for this purpose: Java and C. With regard to its capabilities, power, existence of development and debugging tools, etc; for applications under multiple operating environments (in multiple operating systems) it is important to highlight two fundamental aspects that make these languages better than others for its use in applications with Bluetooth™ technology.

With regard to Java, we find great versatility for applications in embedded systems like PDAs, mobile telephones, etc., to the extent that a specific standard has been developed to support the Bluetooth ™ technology: JABWT (Java APIs for Bluetooth Wireless Technology) [14].

In the case of C language, we find also a similar versatility to Java since the code can be easily reusable and be integrated in an application developed by using high-level language or object oriented language, likewise it can be reusable for the programming of the systems based on such architectures as microprocessors or microcontrollers.

TABLE III. OPERATING SYSTEM REQUIREMENTS

Task	Comment
Process abstraction	Process can not be created, iniciated or stopped dinamically. Each process can only have one example.
Stack	Each task must have its own stack.
Message queue	Each task can only read from one message queue.
Dinamic memory management	The stack reads, assigns and frees memory dinamically.
Timers	Each task manage its own timer/counter.

E. The Virtual Operative system (VOS)

There is an element of great importance that might go unnoticed; it is the Virtual Operating System or VOS (Tables III and IV). The VOS is an abstraction of the services that an Operating System has to provide to the stack of protocols in order to be executed over a specific environment.

These services are a subset of the services that an O.S. would provide for multitasking and that are summarized in Table III; Table IV shows a classification of the most common operating systems, from a technical point of view.

TABLE IV. OPERATING SYSTEMS CLASIFICATION

OS	Description
Class A Windows 95/98/CE, Unix, Mac OS	Multithread (or multitask) with priority. Standar and advanced O.S. Provides a wide variety of services for multitasking. Adecuated for standard protocols and applications development.
Class B EPOC, OSE, PSOS+, VRTX,VxWork, MTOS	Multithread (or multitask) with priority. Multitask O.S. for embedded systems. Designed to be executed on proprietary hardware platforms. Limited set of services in relation to class A operating systems
Class C Windows 3.x	Multithread. Basic O.S. Lack of critical services. Main difference with class A and B operating systems.
Class D Palm OS, no operating system	Monothread or monotask. Doesn't provide multitask services in real time execution.

The aim of the design and development of the concept of the virtual operative system (*VOS*) is that the Bluetooth protocol stack could be directly exported towards environments of an A or B class. This fact does not imply that it could not be exported to more basic environments, it simply means that the management of the multiple processes, that internally provide an A or B class OS, must be re-implemented by the developer, using the methods of variable management and shared processes, such as semaphores or other multithread strategies.

F. System structure

The developed system has been designed following a hierarchical scheme, starting from a simple but fully functional module and adding new features in order to

improve the functionality and open connectivity with mobile devices. As Figure 1 shows, the user establishes a connection with the Terminal. Then, the Control Unit processes the information and generates a subsequent action. These actions could be of the kind of a simple activation of an output (to actuators), recording the timestamp and event, requesting or sending data to/from other similar units or managing the system using, for instance, air interfaces (see Figure 1 and Figure 2).

There are two models of the proposed system that correspond with two levels of security. As to the simple –and cheapest one, the basis is to carry out previously the recording of the MAC addresses of the authorized users in the control unit. Only by performing a connection to the system, that appears like a hands free unit, the Control & Access Point Terminal (Figure 1) knows MAC users, it checks if the user is authorized and performs some predefined actions (e.g., activates an output for door opening, activates/deactivates an alarm, etc.). This way, the end user does not need any additional software or human-machine interfaces.

Additional features are based on the processing of different actions depending on the value of the PIN requested or in other data, like recording the events in a file or sending a message to another control unit.

In addition, Java MIDP 2.0 and JSR82-based software [14] is thought to enable an end user friendly interface, present in most of the mobile phones in market, as an optional feature. This MIDlet is necessary for the so-called "Advanced App." mode of operation (see Section III – A), which performs higher security levels.

The core of this mode of operation (which includes mode .iii Network Advanced) is a Web Server Application that manages the unit control, as shown in Figure 7.

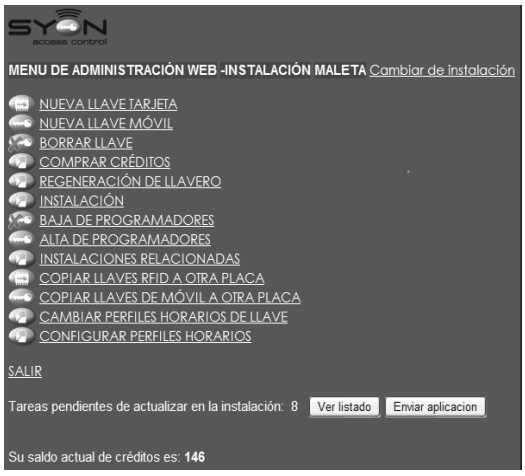


Figure 7. Web application for administrators.

When a user is registered (its phone number), s/he will receive a SMS with a link to this web application. This only takes place once. If the user runs the link, a MIDlet application will be installed in his mobile phone, generating the encrypted id-code for the user and for an Access Point Terminal in particular. Also, a little application stays in the mobile in order to manage the sending and receiving process while requesting to "open" a door or any other action associated to a module or terminal. Nowadays, the MIDlet is present in most of the mobiles in market, having only some limitations in functionality with the iPhone, which are supposed to be resolved in brief through AppleStore.

The main features of this web application are:

- Multilingual web application
- An Activate/deactivate key is fully functional worldwide.
- Management of event record file: downloading and sensing by email or even via SMS.
- Schedule users (e-keys) in real time.
- Valid for most mobile operators worldwide (SMS and GPRS connection must be supported).
- Management of several installations in the same application.

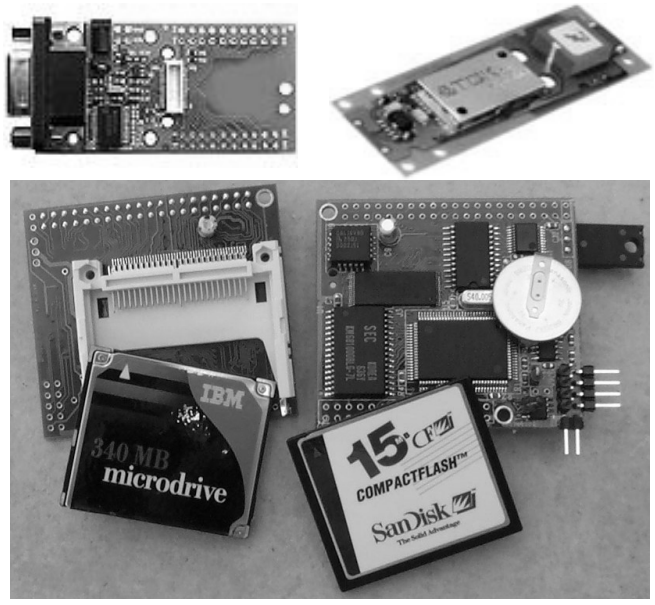


Figure 8. Bluetooth™ Class I modules (upper) and MCU (bottom) used.

Thus, by using this web-based application, the system administrator could easily manage any installation equipped with this kind of modules, e.g.:

- The edition of authenticated users in real time.
- Downloading event record files.
- Ordering the access profile for a key/user –by time and date.
- Association of multiple users and/or multiple installations (terminals).
- Administration of users' privileges.
- Generation of emergency keys, valid only temporarily.
- Other management utilities.

This web application allows to authorize new users in real time without the need to update online the main access module. It is important to point out that the modules work both off-line and on-line, in both cases they preserve the same features and functionality.

When registering new users, we find several gaps with additional information about the end user, like full name, surname, address, country, phone number (country code generates automatically, etc.). When we generate the authenticated user an SMS will be sent containing a link to this web application.

Then, the user connects to the link, if the user mobile phone is validated, then a simple MIDLet application will be downloaded and launched, taking out further information of some parameters of the mobile phone, and generating automatically an unique encrypted code for the selected key which will remain installed in the mobile phone. The security features make this code only valid for the user selected and for the gates predetermined in the signing up process carried out by the administrator.

It is only is necessary to connect to the Internet through the mobile phone the first time we install the MIDlet (encrypted e-key) in the mobile. The rest of the operation and use will be done using Bluetooth™, that is, free of any charge for the end user and independent from the degree of GPRS network coverage.

G. *Proposed implementation*

The developed system is based in a Class I Bluetooth™ module plus a microcontroller-based board acting out as Control Unit (Figure 8).

The Control Unit has as main features the followings: 20 GPIO, 16-bits ADC inputs, 12-bit output DAC, Real Time Clock, SPI ports, Timers and it supports Compact Flash Cards. The Bluetooth™ module is attached to one of the serial ports of the board.

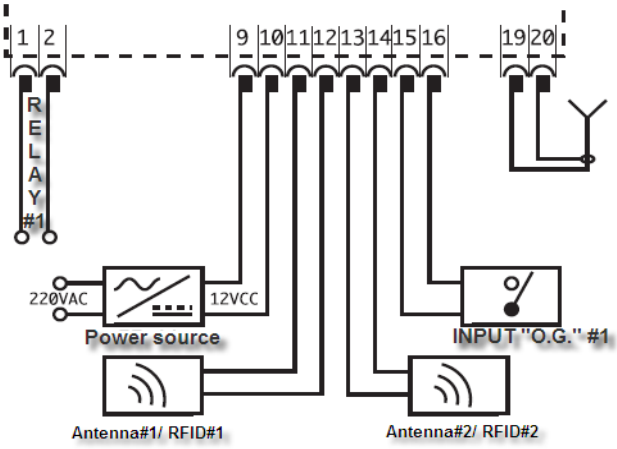


Figure 9. Wiring diagram of terminals for module developed.

This system is low cost and with low power consumption, so it complies perfectly with the system requirements for the controller role assigned (see Section III), which have been proposed and exposed in the present work. The complete wiring diagram is shown in Figure 9.

Additionally, two RFID antennas have been installed to enhance the behavior and market deployment and to provide an added value for the end user. So, the system supports simultaneously access control via Bluetooth and RFID, allowing it to be installed e.g., in a garage of a building and allowing the users to choose the kind of key they prefer. The antennas used in the modules are present in 125kHz tags, preferably Unique® brand.

Also, the Access Point Terminal has up two "open gates" inputs. Each input (12Vdc) must be returned to the system through a free potential conductor. These inputs are used as a feedback in the control unit.

Hardware relays up to four outputs. These outputs are connected to non-potential relays, with maximum power ratings of 1 A - 110Vac or 30 Vdc.

Finally, in Figure 10, a demo board set up in a suitcase is shown. On top of it there are several lights emulating the opening of a door, the activation/deactivation of an alarm, the opening of a garage and the lights switching on/off. Also, both RFID readers and the central switch represent journey endings or "Gate- opening" sensor as module inputs.

At the bottom, we can observe the module itself (bottom right corner) and some RFID tags, some mobile phones and the main DC power sources.

Figure 10. Prototype set up in demo suitcase

IV. CONCLUSION AND FUTURE WORKS

Wireless connectivity has become recognized as a flexible and reliable medium for data communication in a broad range of applications. This is due to wireless networks potential to operate in demanding environments providing clear advantages as far as the cost, size, power, flexibility, and distributed intelligence levels are concerned.

This paper presents a novel application of Bluetooth™ technology to access control systems suitable for home and building applications. The proposed system is low cost, autonomous, physically and functionally scalable and with control and interaction capacities. Moreover, using Bluetooth™ allows designing flexible networks for these kinds of applications. In the paper, some considerations about configuration, security, data management and other features have been presented and discussed.

The final system is divided in three modules: the web server based administrator tool, the end user application and the module itself, which is installed in the access point. Several software posibilities have been developed for an isolated usage or for more complex information exchange among multiple access points.

Further improvements can be done by using this Bluetooth™ based system for pervasive enviroments scenarios. Whilst the system by itself give us the neccesary network infrastructure for communicate, new applications can run to manage the gathered user information for other comfort issues. Of course, security and privacy of data must be assured.

ACKNOWLEDGEMENTS

R&D Project "Bluesensory". Local Government Funding for Research Projects. (Andalucia, Spain) & Stop Casa Segura S.L.

Spanish Patent Pending No.: 200501374/2264387. European Patent Pending No.: 06380014.8

REFERENCES

[1] Bellido Outeiriño F.J.; Canales Aranda, P.M.; de la Cruz Fernández, J.L., Pérez Jarauta, and B. "Universal Bluetooth™ Access Control and Security System for e-Keys Enviroment". Proc. of the 4th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2010). Published by CPS. ©IEEE. 2010, pp. 247-250, DOI 10.1109/SECURWARE.2010.47. ISBN 978-0-7695-4095-5. Best Paper Award.

[2] Bluetooth™ SIG. Bluetooth™ Specifications. Core and Profile. http://www.bluetooth.org, December 2011.

[3] Zigbee™ Alliance. http://www.zigbee.org, December 2011.

[4] Bellido Outeiriño, F.J., Flores Arias, J.M., Real Calvo, R., and Torres Roldán, M., "LR-WPAN Technologies. An approach to industrial applications". Proceedings of International Conference IT Revolutions 2008. pp. 1-4, Venice (Italy). IEEEXplore D.O.I 10.4108/icst.itrevolutions2008.5112

[5] Miroslav Sveda and Radimir Vrba, "Meta-Design with Safe and Secure Embedded System Networking" International Journal On Advances in Security, issn 1942-2636 , Vol. 2, no. 1, year 2009, pp. 8-15.

[6] Seppo Heikkinen, Kari Heikkien, and Sari Kinnari, "Security and User Aspects in the Design of the Future Trusted Ambient Networked Systems"International Journal On Advances in Security, issn 1942-2636, Vol. 2, no. 2&3 year 2009, pp. 156-170.

[7] Il-Kyu, H. and Jin-Wook. B., "Wireless Access Monitoring and Control System based on Digital Door Lock". IEEE Trans. On Consumer Electronics, Vol. 53, No 4, Nov. 2007, pp. 1724-1730.

[8] Soo-Hwan, C. Byung-Kug, K., Jinwoo, P., Chul-Hee, K., and Doo-Seop, E., "An Implementation of Wireless Sensor Network for Security System using Bluetooth™". IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, Feb. 2004, pp. 236-244.

[9] Tajika, Y., Saito, T., Teramoto, K., Osaka, N., and Isshiki, M., "Networked home appliance system using Bluetooth™ technology integrating appliance control/monitoring with Internet service", IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, Nov. 2003, pp. 1043-1048.

[10] Bellido Outeiriño, F.J., de la Cruz Fernández J.L., Torres Roldán M., and Moreno Muñoz, A., "Wireless technology applied to stimulation systems for auditory deficit children". Proc. of the 12th IEEE International Symposium On Consumer Electronics (ISCE 2008). Vilamoura-Portugal, pp. 1-3, D.O.I. 10.1109/ISCE.2008.4559501

[11] RFID. Association for Automatic Identification an Mobility. http://www.rfid.org, December 2011.

[12] EPC Global http://www.epcglobalinc.org, December 2011.

[13] Bluegiga http://www.bluegiga.com, December 2011.

[14] Kumar, C.B. Bluetooth™ Application Programming with the JAVA APIs, Elsevier Science & Technology Books, 2003.

[15] J. Haartsen, *The universal radio interface for hoc, wireless connectivity*, Ericsson Review Vol. 75 (1998):3, pp. 110-117.

Secure Communication Using Electronic Identity Cards for Voice over IP Communication, Home Energy Management, and eMobility

Rainer Falk, Steffen Fries, Hans Joachim Hof
Corporate Technology
Siemens AG
Munich, Germany
e-mail: [rainer.falk | steffen.fries | hans-joachim.hof]@siemens.com

***Abstract*—Using communication services is a common part of everyday life in a personal or business context. Communication services include Internet services like voice services, chat service, and web 2.0 technologies (wikis, blogs, etc), but other usage areas like home energy management and eMobility are will be increasingly tackled. Such communication services typically authenticate participants. For this identities of some kind are used to identify the communication peer to the user of a service or to the service itself. Calling line identification used in the Session Initiation Protocol (SIP) used for Voice over IP (VoIP) is just one example. Authentication and identification of eCar users for accounting during charging of the eCar is another example. Also, further mechanisms rely on identities, e.g., whitelists defining allowed communication peers. Trusted identities prevent identity spoofing, hence are a basic building block for the protection of communication. However, providing trusted identities in a practical way is still a difficult problem and too often application specific identities are used, making identity handling a hassle. Nowadays, many countries introduced electronic identity cards, e.g., the German "Elektronischer Personalausweis" (ePA). As many German citizens will possess an ePA soon, it can be used as security token to provide trusted identities. Especially new usage areas (like eMobility) should from the start be based on the ubiquitous availability of trusted identities. This paper describes how identity cards can be integrated within three domains: home energy management, vehicle-2-grid communication, and SIP-based voice over IP telephony. In all three domains, identity cards are used to reliably identify users and authenticate participants. As an example for an electronic identity card, this paper focuses on the German ePA.**

Keywords - eMobility security; home energy management security; VoIP security; Smart Grid security; electronic identity card; elektronischer Personalausweis; authentication; identification

I. INTRODUCTION

Communication services use identifiers to indicate the intended recipient of a message or to setup a call. In the old telephone system, a telephone number according to ITU-T E.164 was used. But further identifiers are used nowadays for communication, e.g., email addresses, URLs of personal Web page, SIP URIs, Network Access Identifiers (NAI), customer numbers, or instant messaging screen names.

The SIP protocol is used as signaling protocol for Voice over IP (VoIP) communication and is also the base for different instant messaging applications. It uses a SIP URI to identify a user, both the originating and destination party.

The increasing usage of VoIP leads also to the fact that annoyance by unsolicited communication is not restricted to email SPAM, but also to voice calls. Such unsolicited communication is called SPIT, SPAM over Internet Telephony. Some ad hoc countermeasures include the filtering of incoming calls using a whitelist of permitted callers, a blacklist of denied callers, or to make a decision based on the caller's reputation that may be obtained from a reputation system. Also security-critical usages of voice communication take place commonly, as e.g., giving a bank order over a telephone line, or requesting information from a public administration office about the own case.

However, in all these cases trustworthy information about the identity of the communication partner is required as well as the verification of the authenticity of the identity information by either the communication peer or an intermediate component, vouching for the identity of the caller/calleé. Otherwise, identities can be spoofed e.g., to circumvent whitelists and blacklists. Another issue is that obtaining identities must involve some costs; otherwise a malicious party can simply change its identifier whenever it has bad reputation or is blacklisted.

The SIP standard defining VoIP signaling supports various authentication options that apply certain identifiers. Beyond them are direct authentication options but also assertions issued by a trusted third party. However, broad deployment of such a security solution requires a common security infrastructure for identity attestation. As the deployment of a new security infrastructure is costly, it is attractive to reuse an already existing security infrastructure for VoIP/SIP security. With the appearance of electronic identity cards as, e.g., the "Elektronischer Personalausweis (ePA)" in Germany that provides functionality for authentication towards an online service and trusted identities, such a security infrastructure is readily available. Moreover, the security architecture introduced by an official identity card is likely to be considered trustworthy by many people as the identity card is issued by the government and follows defined rules from the initial user authentication till the final identity card emission. Integration of ePA into VoIP communication has already been described in short in [1]. This paper presents the integration of identity cards in

mature applications like VoIP communication as well as in upcoming applications like home energy management, and eMobility. The German "Elektronischer Personalausweis" ePA is taken as an example for how an identity card can be used for authentication of communication partners and how trusted identities of the identity card can be applied in the communication.

The remainder of this paper is structured as follows: Section II introduces ePA user authentication and ePA web authentication as examples of the authentication mechanisms of an identity card that is issued by a trusted instance (the German government). Section III gives an overview of authentication and identification in VoIP communication. Section IV describes identity handling in SIP. Section V describes different VoIP use cases that would profit from an integration of the authentication mechanisms of the "Elektronischer Personalausweis" into the SIP protocol. Section VI and VII describes the use of the "Elektronischer Personalausweis" in the usage area home energy management and eMobility respectively. Technical approaches and practically viable options for integration ePA-based user authentication within SIP/VoIP communication, home energy management, and eMobility are described. Section VIII provides an outlook, while Section IX concludes this paper.

II. AUTHENTICATION USING THE ELECTRONIC IDENTITY CARD EPA

The authentication function of the ePA allows secure transmission of attributes from the ePA to a third party. Attributes may be related to a person (name) or to a property or characteristic (e.g., age). Even relative attributes like "holder is older than 18" are possible. Attributes related to a person may be used as identities in VoIP communication. The corresponding authentication mechanism hence allows identification of communicating parties. The ePA offers various forms of authentication. In the following, the Extended Access Control is described:

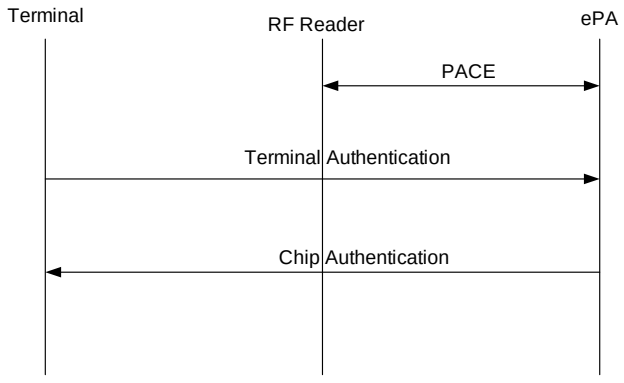


Figure 1. PACE Message Flow

Extended Access Control between a Terminal and an ePA involves three phases, see Figure 1: In the first phase, the PACE protocol is executed that is necessary to access the ePA. In the second phase, the terminal authenticates against the ePA, and in the third phase the ePA authenticates against the Terminal.

- PACE: PACE is a password authenticated Diffie-Hellman key exchange. A session key for protection of the communication is set up between the Radio Frequency (RF) Reader and the ePA.
- Terminal authentication: Each terminal has a terminal certificate for identification. The certificate is signed by a root CA. All root CA certificates are based on an international public key infrastructure (PKI). The German root CA is hosted by the BSI (Bundesamt für Sicherheit in der Informationstechnologie). Terminal certificates have a very short lifetime (24 hours). However, the ePA does neither have a certificate revocation list nor a physical clock. Instead, it stores the time of the last successful verification and does only accept timestamps that lie in the future. The terminal certificate includes information encoding which attributes of the user may be provided to that terminal.
- Chip authentication: A Diffie-Hellman authentication using a static chip key is performed to authenticate the ePA towards the Terminal.

Another authentication function of the ePA is web authentication (cf. Figure 2).

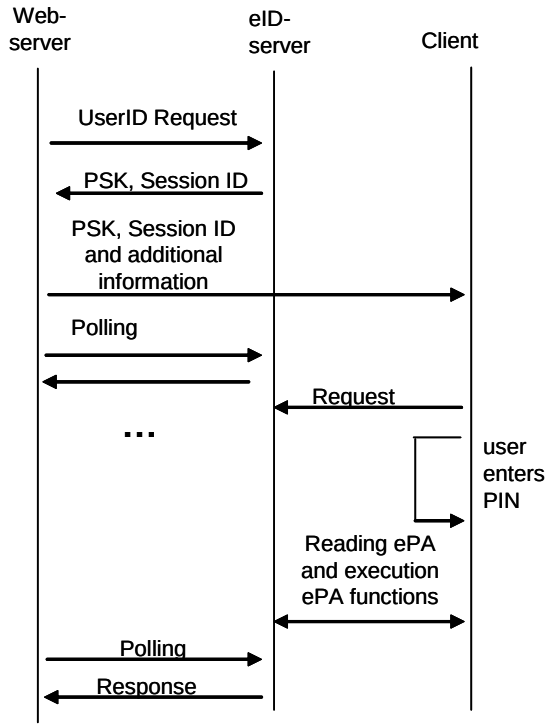


Figure 2. Web Authentication

A web server requests authentication of a user by sending a UserID Request to the eID server. The eID server provides a symmetric key (PSK) as well as a Session ID to the web

server. The web server sends the PSK, the Session ID, and additional information related to the authentication request to the client of the user. After sending this message, the web server starts polling the eID server for a response of the authentication request. Upon reception of information from the web server, the client of the user sends an authentication request to the eID server. This starts the authentication between the eID server and the web client of the user. In the process of authentication, the user enters its PIN to activate the ePA. The eID server reads the ePA and executes ePA function. When this process finished, the eID server sends a response upon reception of a polling message of the web server. The response includes the result of the authentication request. The provider of the web server must ensure that the confidentiality of the PSK is protected during transfer, e.g. by using TLS/SSL, see [10] for details. Nowadays, eID services are offered by various providers, e.g. Bremen Online Services GmbH [11], Bundesdruckerei GmbH [12], and Deutsche Post Com GmbH [13]. Those providers offer integration modules for use in web servers.

This authentication approach can be integrated in existing multimedia applications as well as be used for upcoming usage areas as shown in the next sections. Keying material established by the authentication mechanism (some authentication methods support inband key agreement) may be used for further protection of, the established communication channel (e.g., for protection of integrity and confidentiality of VoIP communication, which is out of scope of this paper).

III. Authentication and Identification in VoIP Communication

Main security objectives with relevance for VoIP communication concern the authentication and identification of participants, and the protection (integrity, confidentiality) of signaling and media data. Authentication and identification are the basis for the protection of VoIP communication; hence the focus of this paper is on authentication and identification. This section highlights some basic concepts of authentication and identification of participants in VoIP communication.

One may distinguish device authentication and user authentication: device authentication authenticates devices, even if they are shared between different users. Standard voice communication often uses device authentication, e.g., a telephone that is available to more than one person but only has one telephone number (used as identifier) associated. User authentication in contrast authenticates users, hence distinguishes between different users even if they use the same device. Appropriate identifiers for the intended authentication target need to be used. Device authentication and user authentication may be used together, or between different parties, e.g., device authentication to the service provider and user authentication to the calleé.

Another important point is who authenticates to whom. One may distinguish here between unilateral authentication and mutual authentication. For example, in unilateral authentication, a caller authenticates to a service provider. Mutual authentication is used if the service provider also authenticates to the caller. Mutual authentication may be necessary to avoid certain types of attacks. One example may be Man-in-the-Middle attacks, were the attacker manages to be an intermediate node in the communication path between the user and the server acting as the server towards the user and misusing the user credentials towards the server.

The endpoints of authentication are also of interest: the caller may authentication to a service provider or to the calleé. In the case of mobile phones, a user authenticates to the service provider and no end-to-end authentication between caller and calleé takes place. This approach is based on the trust users have in their service providers. The service provider may or may not assert the callers identity to the calleé. Asserted identities by a service provider may be trustworthy if only one service provider is involved (e.g., if a mobile phone connects to another mobile phone in his own network). However, if a call involves several service providers, the trustworthiness of the asserted identity may be much lower. Even worse, if the service provider use different technologies like one uses VoIP and the other one PSTN, it is more likely that such information is not provided end-to-end.

IV. Identity Handling in SIP

The SIP protocol is the major signaling protocol for VoIP communication. It can be used for instant messaging (SIMPLE) as well. The SIP protocol defined in RFC3261 [2] supports two basic options for providing identity information in the SIP header to peers and also several options for authenticating users. For the provisioning of identity information the *From* header field defined as part of RFC3261 can be used as well as the *P-Asserted-Identity* header field, which is defined in RFC3325 [7]. While the first identity field is provided by the client itself, the latter one is typically used between trusted intermediaries (proxies). The following authentication options utilizing at least one of the fields named, but may not always provide true end-to-end protection of these fields. This is due to the fact that intermediate entities may alter some of the fields. This can be done for instance through Back-to-Back User Agents, which terminate a session setup in both directions. These entities are neither explicitly defined by SIP nor forbidden.

A. SIP with HTTP Digest

SIP digest authentication is based on HTTP digest authentication used by HTTP communication with Web servers. It authenticates a client using a username (identity) and a password or secret key in a simple challenge-response authentication, applying cryptographic hash functions. Thus, the password is never sent in the clear. Nevertheless, there are some deficiencies in the usage of the HTTP Digest scheme, as it does not provide complete message integrity and cannot be applied to all messages. Here Transport Layer Security (TLS) kicks in, which can be used for signaling protection.

B. SIP with TLS

The TLS protocol is the successor of the well-known Secure Socket Layer (SSL) protocol. It protects all communication on the transport layer for TCP connections against loss of integrity, confidentiality and against replay attacks. RFC3261 mandates the support of TLS for SIP proxies, redirect servers, and registrars to protect SIP signaling. Using TLS for User Agents (UAs), the SIP clients, is recommended. It provides integrated key-management with mutual authentication and secure key distribution. TLS is applicable hop-by-hop between UAs and proxies or between proxies. The SIP-Secure (SIPS) scheme defined in RFC3261 requires the usage of TLS to protect the signaling until the last proxy in the call flow. According to RFC3261 the last hop (from the proxy to the client) remains unprotected as a separate signaling connection is used for sending and receiving. This deficiency is being handled as part of RFC5626, describing an option to use client initiated connections also for the return signaling. RFC5923 is discussing a similar solution for the connection of two communicating proxies (cf. [6]).

C. S/MIME to protect SIP message body data

The S/MIME standard is commonly used for encrypting and signing emails. RFC3261 recommends S/MIME to be used for end-to-end protection of SIP signaling message payloads following a similar approach as email. S/MIME within SIP supports authentication, integrity protection and confidentiality of signaling data. However, S/MIME is not widely used in current SIP deployments. One reason may be the overhead produced by S/MIME in terms of message size and complexity of parsing S/MIME protected content, which may be not acceptable in synchronous communication.

As new scenarios arise, the defined security measures within SIP do not always provide a solution. SIP is flexible with regard to extending the protocol. Therefore several security enhancements have already been standardized. Two approaches supporting identity management services needed in the different security levels are described in the following as prominent examples for extensions.

D. Authenticated Identity Management

RFC4474 defines enhancements for SIP providing assertions for the user identity (Address of Record) valid in the domain the authentication server is responsible for to securely identify originators of SIP messages [3]. New header fields include a signature used for validating the identity and a reference to the certificate of the signer. As RFC4474 only addresses the request direction, a further standard has been defined: RFC4916 [4] describes the application of an authenticated identity service for the response direction.

Unfortunately, both documents do not provide a solution suitable for all relevant scenarios. They preferably work in adjacent domains. If, however, multiple administrative domains are to be traversed, the likelihood increases that a Session Border Controller SBC, i.e. a SIP proxy running on the border of a network, or a SIP proxy working as Back-to-Back user agent (B2B UA) is located on the signaling path. A B2B UA is a proxy that terminates the signaling traffic in both directions. This intermediate node may alter the SIP header or even the SIP body, hence destroying the signature of the authentication proxy. As the problem of a real end-to-end identity is not solved with the available solutions, further approaches are expected to be proposed.

E. SIP/SAML

The Security Assertion Markup Language SAML is an XML extension for exchanging security information that has been developed by OASIS. SAML is a XML-based framework for creating and exchanging security information. When SIP requests are received by a server, there may be authorization requirements that are orthogonal to ascertaining the identity of the User Agent Client (UAC). Supplemental authorization information might allow the UAC to implement non-identity-based policies that depend on further attributes of the principal that originated a SIP request.

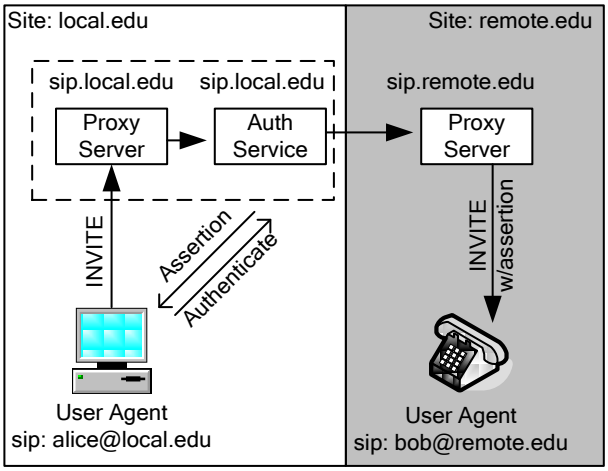


Figure 3. SAML Usage in SIP

An Internet draft [5] proposes a method for using SAML in collaboration with SIP to accommodate richer authorization mechanisms and enable trait-based authorization based on roles or traits instead of identity (see Figure 3). Moreover, it also defines how SAML assertions can be carried within SIP, which can be used end-to-end to vouch for a certain identity.

As stated at the beginning of this section, up to now SIP does not provide a universal identity management solution, which can be applied in every scenario. Therefore, the discussion within the IETF proceeds defining dedicated elements in the SIP messages, which may be altered by intermediate components, without scarifying the complete message identity. There already exist drafts describing potential solutions.

V. USAGE AREA VOICE OVER IP COMMUNICATION: PROVIDING SIP IDENTITIES THROUGH EPA

The ePA user authentication can be applied in SIP-based VoIP telephony, either directly integrated into or adjacent to the SIP protocol

Different use cases for ePA-based user authentication pose different requirements on the technical solution as described in the following. The authentication can be performed towards the SIP service provider or toward the SIP communication peer (calleé), see Figure 4. Therefore, also different technical options for integrating ePA-based user authentication within SIP are needed.

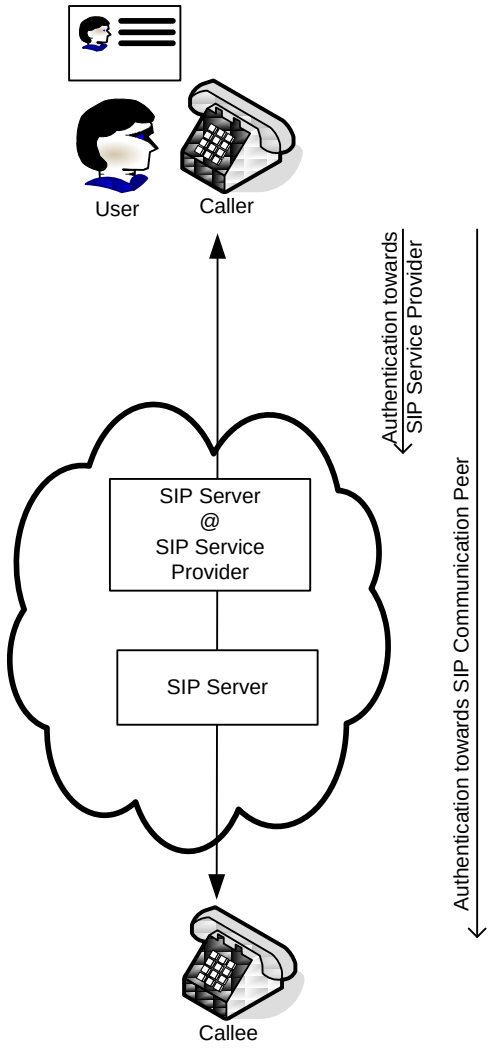


Figure 4. ePA-based VoIP/SIP User Authentication Scenarios

A. Use Cases

User authentication within SIP protocol supported by the ePA functionality can be used to realize different application scenarios:

- Authentication of the user towards the SIP service provider for authorizing the use of SIP-based communication services. An ePA-based user authentication is performed when using SIP-based communication services.
- Authentication of the user towards the SIP service provider for bootstrapping security credentials needed for using SIP-based communication services. An ePA-based user authentication is performed as part of the establishment of a valid SIP-device configuration.
- Authentication of a user as part of registering with a communication service provider using a self-subscription Web portal. Here, the ePA user authentication is not used for technical authentication within the SIP application, but to establish the business relationship (contract) with a service provider. This may happen independently of SIP communication services.
- Authentication of the user towards an Identity Provider independently of the SIP signaling. The Identity provider issues an assertion (e.g., SAML assertion) that can be used with various communication services. Assertions may even be part of SIP.
- Authentication of the user towards the communication partner. Assured user identification or attributes of the user (as e.g., the user's age or even a pseudonym) can be provided to the communication peer. This information can be used by the communication peer for providing personalized communication services. For example, when calling a service hotline (merchant, public institution as a finance office), the calling user can be automatically identified resp. the user's attributes can be verified. When e.g., premium communication services are used, attributes as, e.g., the user's age can be verified before providing services.
- Media encryption: Session keys derived from ePA authentication may be used to encrypt the media stream towards the SIP service provider or the SIP communication peer. However, protection of VoIP communication besides authentication is out of scope of this paper.

Using ePA-based authentication towards the communication peer cannot be used in all of the described use cases. In particular in public SIP service offerings, SIP signaling is often terminated by the service provider's SIP server, so that specific SIP signaling may not reach the communication peer. This specifically applies to SIP headers, which may be added or removed by intermediaries.

The stated use cases vary concerning the following main requirements:

- SIP Protocol Integration: The ePA user authentication can be integrated within the SIP protocol itself, or it can be performed outside the SIP protocol for establishing a security context that may be used within SIP later on.
- Authentication Peer: The ePA authentication information can be performed towards a generic identity provider, a SIP service provider, or the communication peer.

- Frequency: An ePA-based user authentication can be performed only once to bootstrap SIP security configuration, or with each SIP session.

Therefore, different technical solutions for integration of ePA-based user authentication within VoIP are outlined in the next subsection.

B. Integration of ePA-based Authentication within SIP

Different possibilities exist for integrating ePA user authentication into SIP applications:

- Manual integration using an online Web authentication service: if a voice services requires identification, the caller connects to an associated website and uses the ePA Web Authentication mechanism. If authentication succeeds, a one time password is given to the caller that can be used for authentication to voice services, e.g., by entering the password using DTMF tones. This obviously requires a connection of the voice service provider with the ePA authentication infrastructure.

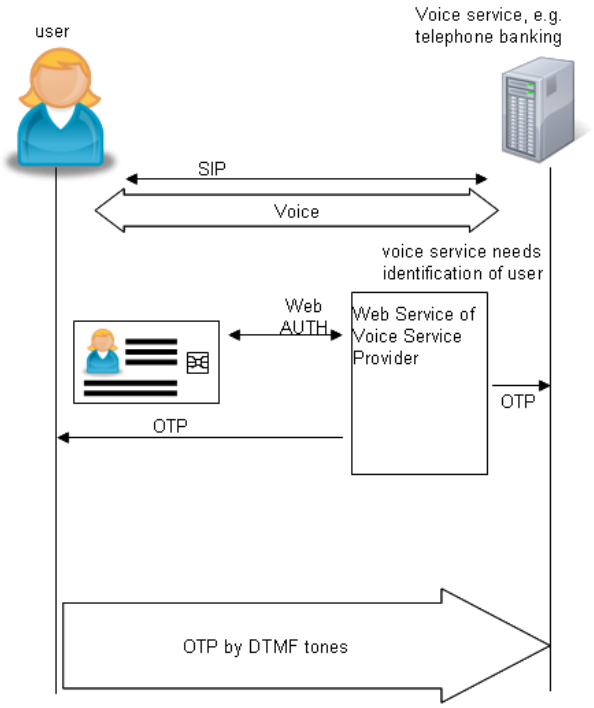


Figure 5. Manual ePA Integration

- HTTP Digest within SIP: The ePA authentication requires a specific server certificate to activate the ePA client authentication. Hence an ePA Web authentication is performed first to obtain credentials that are used as HTTP Digest parameters within SIP. The same requirement as above applies to this example.

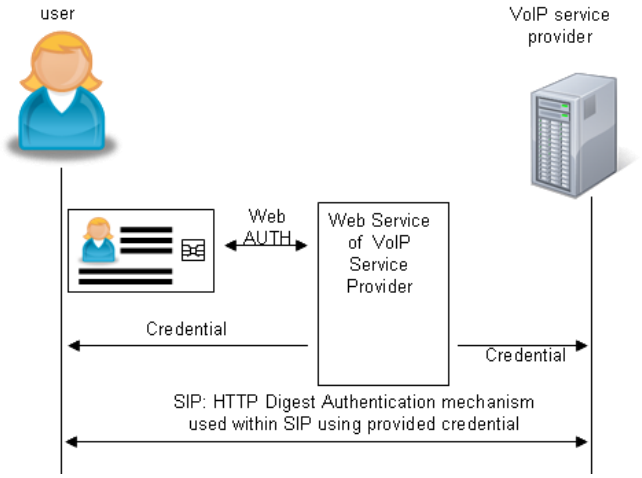


Figure 6. ePA integration using HTTP digest

- Perform ePA client authentication within TLS security protocol protecting the SIP signaling channel towards the service provider, assumed the service provider possess the specific ePA server certificate. This would be transparent to the SIP protocol itself and completely relies on the usage of SIPS (SIP over TLS). It allows for authentication towards the SIP service provider.
- Authentication towards a separate identity provider issuing an identity assertion: A separate identity provider asserting identities is a flexible solution that allows fine-grain user control of revealed data. However, this approach requires both, the calleé and caller, to trust the identity provider. Involving a third party also poses privacy risks as the identity provider can collect caller data. This approach may be performed inband the SIP signaling as described for the SAML application in SIP.

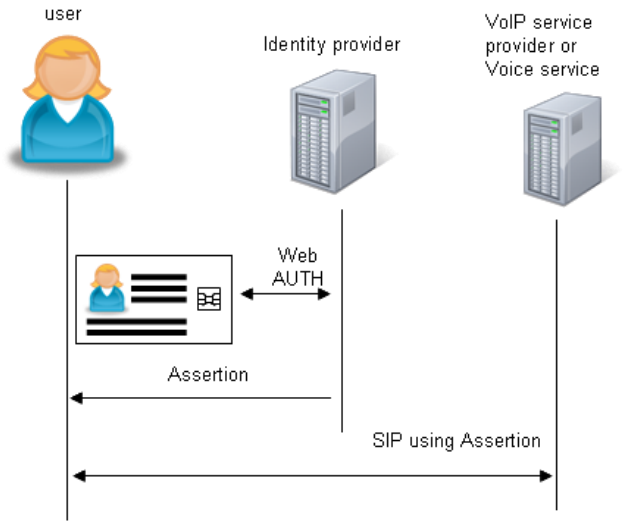


Figure 7. ePA integration using identity provider

- SIP with S/MIME: For end-to-end authentication, the integration of the ePA in SIP may use an additional part of the SIP message body. This part can be protected with S/MIME hence allows for integrity protection of the caller information through signing the S/MIME part with the callers private key. Moreover, if pseudonyms are used in the outer SIP header, the S/MIME part of the message may be used to transport the real identity of the caller in a privacy protected manner. This can be achieved by encrypting the S/MIME part using the calleé's public key (certificate). Note that this requires the availability of the calleé's certificate before establishing the call.

C. *Reality Check*

From a real-world perspective, the following use cases seem to be appropriate:

- ePA Authentication towards a Web Portal: The ePA-based Web authentication mechanisms is already available. Extensions to the SIP protocol are not necessary, as HTTP Digest authentication can be used as defined. Necessary is the integration of a preceding step binding the ePA Web Authentication to the HTTP Digest Authentication.
- ePA Authentication using an Identity Provider (SIP/SAML): The ePA-based authentication mechanisms is readily available. Extensions to the SIP protocol are already defined but it has to be considered, that they are still in draft status.
- ePA Authentication towards SIP Service provider: The easiest approach to integrate ePA authentication within SIP is to map the Web-based authentication on SIP, i.e. to use TLS for server authentication and to transport ePA authentication messages within SIP in the same way as over HTTP. Here, the SIP service provider needs an ePA terminal certificate to verify the user's identity. The SIP service provider may include verified user attributes in the SIP signaling towards other SIP service providers in case of multiple provider spanning connections, e.g., using the P-asserted identity extension (cf. [7]).
- ePA Authentication towards SIP Communication Peer: A direct integration within SIP signaling is difficult, as a SIP communication partner will usually not possess an ePA terminal certificate. The ePA user authentication follows a strict client/server principle, not a symmetric peer-to-peer like model as VoIP telephony. Also SIP signaling will often be terminated by the SIP service provider. Therefore a realistic approach seems to run an ePA-based user authentication towards a service/identity provider. The caller and the calleé may authenticate independently. Then known SIP security mechanisms can be used for end-to-end security (asserted identity, media encryption). The security association is, however, not established directly between the communicating entities, but by infrastructure nodes that have to be trusted.

VI. USAGE AREA HOME ENERGY MANAGEMENT

Allowing consumers to monitor their current energy usage level, even from remote, is one of the visions of the smart grid. The expectation is that energy usage monitoring results in a wiser use of energy in everyday life. Figure 8 shows a typical smart grid architecture.

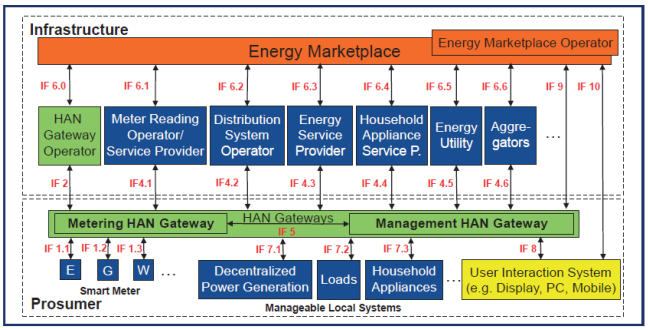


Figure 8. Smart Grid Reference Architecture from [9]

The architecture includes an "intelligent home" at the prosumer side (prosumer=energy PROducer + energy conSUMER), especially a HAN Gateway (Home Area Network Gateway) that exposes the available information on the energy consumption from the home to the outside, e.g., the Internet. To protect the smart home against attacks from the Internet, it is required to authenticate the communication (establishment) at the HAN Gateway. In the smart grid system, most communication partners of the HAN Gateway are already known and rather static. Secure identities are likely to already exist, for instance for certain control operations of or pricing information from the smart grid. However, secure identities for users accessing their HAN Gateway to monitor energy usage

via Internet do not exist. Electronic identity cards can be used here to provide the secure identities needed.

A. *Use Cases*

The user can access information about its smart home (e.g. energy usage in the home) using a web browser. Several ways exist to implement this energy consumption web page:

1. The HAN Gateway runs a web server and provides the web page. The user authenticates towards the HAN Gateway using the ePA. However, this would require the HAN Gateway to implement an eID server. This means that the HAN gateway must comply to the very strict requirements defined in [10], which for example requires a hardware security module.
2. The utility provides the web page and polls the HAN Gateway for information. In this case, the user authenticates towards the utility using the ePA. The utility uses existing security associations that are independent of the ePA authentication to secure communication with the HAN Gateway. The prosumer could also use the ePA authentication towards the utility to establish or change contractual agreements as e.g. the selected tariff.

B. *Options for Integration ePA-based User Authentication*

Different possibilities exist for integrating ePA user authentication with HTTP access to the web page over a public network (Internet) providing information about the HAN Gateway:

- Direct use of web authentication (see section II)
- HTTP Digest: The HTTP protocol is used to establish a session between the HAN Gateway and the user. The ePA authentication requires a specific server certificate to activate the ePA client authentication. Hence an ePA Web authentication is performed first to obtain credentials that are used as HTTP Digest parameters within HTTP access towards the Web server (see Figure 9.).

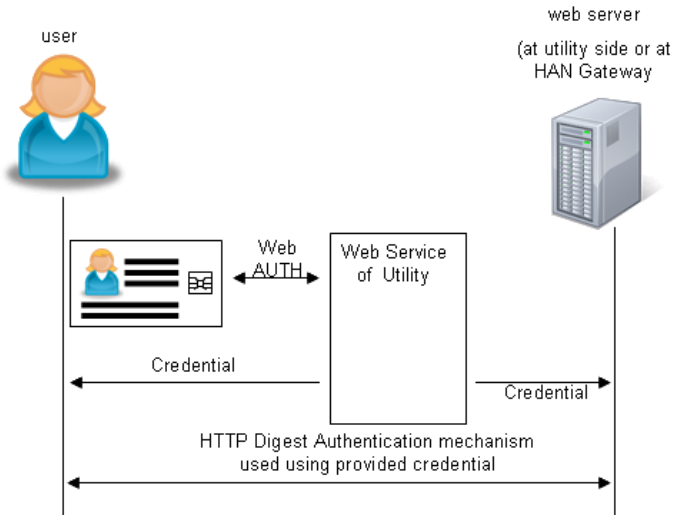


Figure 9. ePA integration using intermediate credential provider

- Perform ePA client authentication within the TLS security protocol protecting the communication towards the web server, assuming the HAN Gateway possesses the specific ePA server certificate.
- Authentication towards a separate identity provider issuing an identity assertion (see Figure 10.): A separate identity provider asserting identities is a flexible solution that allows fine-grained user control of revealed data. However, this approach requires both communication parties to trust the identity provider. In the home energy monitoring use case, the utility may for example be such a trusted party. Involving a third party also poses privacy risks as the identity provider can collect service usage information. Here, it may be possible to use SAML or Kerberos to issue a security token asserting the successful authentication towards the identity provider

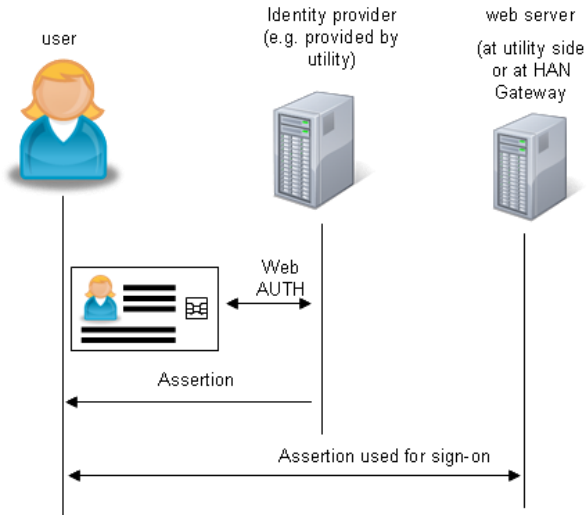


Figure 10. ePA integration using identity provider

C. *Reality Check*

From a real-world perspective, the following use cases could be realized:

- Direct use of web authentication: This integration approach seems to be only realistic if the utility provides the web site that the user accesses. In particular, a special server certificate is needed for the web server to interact with the ePA. Expecting each HAN Gateway to obtain such a server certificate seems unrealistic. Moreover, often HAN Gateways in a residential area will be connected via a Digital Subscriber Line (DSL) or power line communication and may therefore not be accessible by the prosumer directly (they do not have a public, static IP address, requiring services like MobileIP or DynDNS, to make them accessible from the public Internet).
- HTTP Digest: As with the approach above, this approach seems to be only realistic if the web site that the user accesses, is provided by the utility. However, this kind of session establishment involves an

unnecessary indirection. This indirection may be justified if the HAN Gateway does not possess a fixed IP address, e.g., through connectivity via DSL. Thus address resolution for the HAN gateway is necessary. This may be achieved by having the HAN gateway permanently registered with a utility server, detecting IP address changes via the registration messages. This registration should be done using a secured connection between the HAN Gateway and the utility server. This utility server may then also perform the HTTP digest authentication mechanism and act as a proxy for remote access to the HAN Gateway.

- ePA Authentication towards a Web Portal or an Identity Provider (SAML): The utility can provide identities as it has a customer relationship with the user.

As the usage are home energy management is not yet mature, there is still the opportunity to embed ePA authentication into emerging protocols and standards.

VII. USAGE AREA CHARGING SPOT ACCESS

Electric vehicles are becoming more and more important in the national and international strategies, e.g., to reduce CO_2 emissions. The integration of e-cars into the Smart Grid is still an evolving issue. Architectures like seen in Figure 11. are currently emerging.

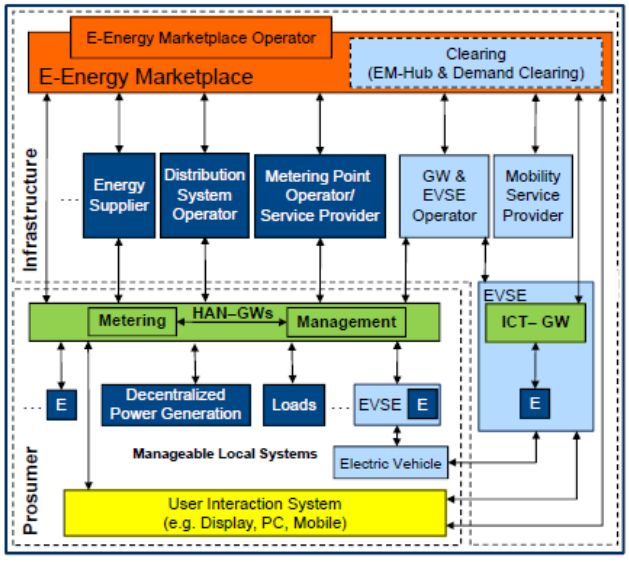


Figure 11. eMobility Extension of Smart Grid Reference Architecture from [9]

Authentication using the ePA can be useful for usage authorization at public charging spots. An authentication using a secure identity is important as the identity is used for accounting and billing. A user lock-in by different solutions of different vendors can be avoided if the already available ePA infrastructure is used for authentication. A charging provider could re-use an existing authentication infrastructure without having to issue separate authentication tokens. This has to be evaluated regarding potential privacy issues when re-using an authentication service revealing the user identity.

A. Use Cases

The user authenticates towards a charging spot before the user's electric car is charged.

B. Options for Integration ePA-based User Authentication

Different possibilities exist for integrating ePA user authentication as the user may authenticate directly at the charging spot or from inside the car. This requires the appropriate ePA interfaces on either the car or the charging spot. In the following, the focus is ePA authentication at the charging spot directly:

- Direct terminal authentication (see Section II): Each charging spot is acting as terminal for ePA authentication.
- HTTP Digest (see Figure 12.): The charging spot uses HTTP to establish a session with the charging provider in the backend system. The ePA authentication requires a specific server certificate to activate the ePA client authentication. Hence an ePA Web authentication is performed first to obtain credentials that are used as HTTP Digest parameters.

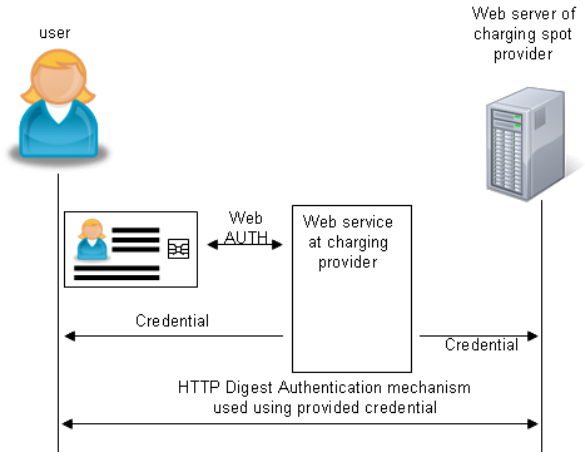


Figure 12. ePA integration using HTTP digest

- Perform ePA client authentication within the TLS security protocol protecting communication towards the charging service provider, where the charging provider possesses the specific ePA server certificate.
- Authentication towards a separate identity provider issuing an identity assertion (see Figure 13.): A separate identity provider asserting identities is a flexible solution that allows fine-grained user control of revealed data. This approach allows easy integration with existing authentication solutions. However, this approach requires both the charging spot provider and the user to trust the identity provider. Involving a third party also poses privacy risks as the identity provider can collect mobility data. This approach may use for example SAML Assertions.or Kerberos tickets.

The identity used for authentication is the name of the user, potentially together with the user's address information. However, the current design of charging points often requires a contract ID instead of a real user name. The contract ID is linked with a name (resp. an associated database entry) in the backend system.

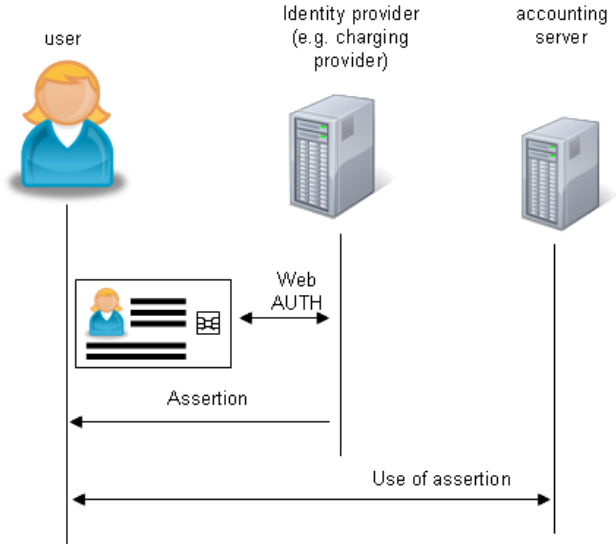


Figure 13. ePA integration using identity provider

C. *Reality Check*

From a real-world perspective, the following use cases could be realized, assuming that the charging spot provides the required interfaces for ePA interaction:

- Direct terminal authentication requires ePA server certificate for acting as a terminal as well as additional hardware [14], which may result in higher costs for charging spots.
- HTTP Digest: This approach could be implemented using available technology.
- ePA Authentication towards an Identity Provider (SAML): The ePA-based authentication mechanisms is available and the charging provider may serve as identity provider.

VIII. OUTLOOK

Based on the use cases and potential solution options stated for the usage areas voice over IP communication, home energy management, and vehicle-to-grid integration the evolvement of identity card (e.g., ePA) based authentication in well established applications as well as in upcoming applications is highly expected. This expectation is supported by the fact that there is an available central authentication infrastructure, which can easily be used within a variety of services requiring authentication or some form of assertions of characteristics like the age of a person.

Moreover, from a technical point of view, it is also possible to load further applications onto the identity card, which enables further authentication schemes to be executed directly on the identity card supporting even more scenarios and usage areas by using the identity a universal secure transmission device of authentication credentials and applications. This would even enable further usage of the identity card in the curse of key management protocols, e.g., to agree on a session secret used to provide integrity or confidentiality or to simply serve as hardware based key generator for the key management protocol. Note, that the legal ramifications having multiple applications residing on one identity card as base for this option need to be further elaborated.

Regarding the direct utilization within SIP, further investigation into the enhancement of currently defined assertion or claim based service support to better utilize ePA functionality is recommended. Thanks to the SIP extensibility, new authentication services may be added without changing the base protocol. Some of the stated use cases in Section V apply the ePA for the initial security bootstrapping of SIP user environments. This option should be investigated more deeply, e.g., in the context of already established frameworks like the GBA (Generic Bootstrapping Architecture) or SACRED, (Securely Available Credentials, RFC 3760), as it provides a more generic bootstrapping option.

Regarding home energy management and vehicle-2-grid, embedding the use of identity cards into upcoming protocols and standards is an important issue. Moreover, as already stated the possibility to load own applications onto an identity card may provide even further solution spaces.

IX. CONCLUSION

This paper gives on overview on different aspects of authentication and identity management for voice communication, home energy management as well as eMobility. For authentication and identity management, identity cards like the German "Elektronischer Personalausweis" (ePA) can be used. In Germany, a security infrastructure for identity management using the ePA is provided that is highly trusted and can be beneficial for protecting mature applications like voice communication as well as upcoming applications like home energy management and vehicle-2-grid. This paper presented a number of options for integrating an ePA-based user authentication in protocols typically used in voice over IP communication (SIP - Session Initiation Protocol), home energy management, and vehicle-2-grid. Identification and authentication are important building blocks in the protection of communication in all three usage areas. They may serve as a base for the exchange of further (session based) keying material which may be used for further protection of the session, e.g., for protection of integrity and confidentiality.

REFERENCES

[1] R. Falk, S. Fries, and H.-J. Hof, "Protecting Voice Communication Using Electronic Identity Cards", in Proceedings of The Third International Conference on Advances in Human-oriented and

Personalized Mechanisms, Technologies, and Services (CENTRIC 2010), Nice, France, August 2010

[2] J. Rosenberg, H. Schulzrinne, et al.: "SIP: Session Initiation Protocol", RFC3261, IETF, 2002.

[3] J. Peterson and C. Jennings: "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC4474, IETF, 2006.

[4] J.Elwell: "Connected Identity in the Session Initiation Protocol", RFC4916, IETF, 2007

[5] H. Tschofenig, J.Hodges, et al.: "SIP SAML Profile and Binding", draft-ietf-sip-saml, http://tools.ietf.org/html/draft-ietf-sip-saml-08, Work in Progress, accessed 04.01.2011.

[6] V. Gurbani, R. Mahy, and B. Tate, "Connection Reuse in SIP", RFC5923, IETF,.2010

[7] C. Jennings, J. Peterson, and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC3325, IETF, 2002

[8] C. Jennings, R. Mahy, and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC5626, IETF, 2009

[9] C. Müller, J. Schmutzler, C. Wietfeld, S. Fries, A. Heidenreich, and H.-J. Hof, "ICT Reference Architecture Design based on Requirements for Future Energy Grids", First International Conference on Smart Grid Communications (IEEE SmartGridComm 2010), Gaithersburg, Maryland, USA, Oktober 2010

[10] BSI, "Technische Richtlinie eID-Server", BSI TR-03130 Version 1.4.1, 2010

[11] http://www.bos-bremen.de/de/governikus_autent/1854605/, accessed 13.01.2011

[12] http://www.bundesdruckerei.de/de/produkte/produkte_dokument/dok_personala/dok_eiDs/index.html, accessed 13.01.2011

[13] http://www.deutschepost.de/dpag?tab=1&skin=hi&check=yes&lang=de_DE&xmlFile=link1022943_1022939, accessed 13.01.2011

[14] BSI, "EAC-Box Architecture and Interfaces", Technical Guideline TR-03131, Version 1.1, 2010

TREMA: A Tree-based Reputation Management Solution for P2P Systems

Quang Hieu Vu
ETISALAT BT Innovation Center (EBTIC)
Khalifa University, UAE
quang.vu@kustar.ac.ae

***Abstract*—Trust is an important aspect of Peer-to-Peer (P2P) systems, because in such systems, peers are usually anonymous. A popular method for evaluating trust in P2P systems is to use reputation, where the reputation of a peer is determined based on its prior transactions with other peers. Since no peer has easy access to global knowledge in a decentralized system, the main challenge of this reputation-based method is how to collect and distribute reputation scores of peers efficiently. While several solutions have been proposed to address this challenge, most of them rely on a gossiping algorithm, which is costly and communication-intensive. In this paper, we propose TREMA, a tree-based reputation management solution in which we present a trust model between nodes in the tree, and explain how trust is established and maintained between pairs of nodes. We show that, compared to existing solutions, TREMA allows for scalability and efficient algorithms with low overhead. We present two possible implementations of TREMA, and explain how they could be made stable and robust to network dynamism, thus addressing the greatest weakness of a tree structure. We also analyze each implementation for its security against various adversarial scenarios, and suggest further improvements that are possible for general tree-based systems.**

***Keywords*-Peer-to-Peer; Security; Trust Evaluation; Reputation Management; Tree Structure.**

I. INTRODUCTION

Over the last decade, Peer-to-Peer (P2P) systems have received more and more attention from both computer users and researchers. They have become the first choice for large-scale distributed systems due to their scalability. Nevertheless, there are still problems that need to be solved before P2P systems can be truly ubiquitous, one of which is security. Since peers are usually anonymous, security is a problem of greatest concern among people using P2P systems. A popular method for evaluating trust in distributed systems as well as in P2P systems is to base on reputation, where the reputation of a peer is summarized from opinions of all peers who have participated in previous transactions with that peer. For examples, users of eBay [1] and Amazon Auctions [2] are provided a separate channel for feedback. After each transaction, both sellers and buyers can rate each other and the score is kept for a later reference. In this way, from the reputation score of a person, others can decide easily if they can trust that person or not (e.g., a high reputation score is an indicator of having had many successful and trustworthy transactions).

The main challenge of the reputation-based method for trust evaluation in P2P systems is how to collect opinions of all peers in the system about a particular peer, and to provide access to the reputation score to all who request it. In existing reputation-based systems like eBay and Amazon Auctions, the solution to both challenges is to use servers. However, this solution suffers from problems of server-based systems such as network bottlenecks and a single point of failure. An alternative solution is to employ a gossiping algorithm [3], [4], [5], [6] for exchanging knowledge among peers in the system. In this way, after a sufficient number of knowledge exchange steps, every peer should have a global knowledge about reputations of all peers in the system.

The gossiping algorithm can be implemented in two ways. In the first way, each peer itself has to maintain global state and knowledge of the whole system. After each transaction or after some interval time, peers report the score of their partners in new transactions to all other peers in the system. Based on this report, peers update their global state. This method requires that peers keep and maintain reputation scores for all peers, which is inefficient. The second way avoids this problem by letting each peer keep track of the reputation of peers that it has been in transactions with previously. Whenever a peer wants to retrieve the reputation of another peer, it can apply the gossiping algorithm to ask for that peer's reputation from its neighbors, the neighbors of its neighbors, and so on. Combining the feedback with its local knowledge, it can determine the trust value of that peer. Even though these two ways are different, they share the same drawback of the gossiping algorithm: both are expensive in terms of computation and communication costs.

Instead of using gossiping, in this paper, we present TREMA, a Tree-based REputation MAnagement solution. In TREMA, we organize nodes at different positions in a tree-based on their reputation, with peers of higher reputation at higher levels. In this tree structure, reputation of a peer is maintained at its parent. A peer always trusts its ancestors while it is answerable for its descendants. When two peers execute a transaction, a trust route is formed between them. If the transaction succeeds, a reward is given to all nodes in the route. On the other hand, if the transaction fails, all nodes in the route are penalized. The main advantage of TREMA is that it does not incur a high cost in reputation management compared to methods that use the gossiping algorithm for

reputation distribution. Furthermore, the flexible design of TREMA allows us to develop a complete system for trust management to use in any existing decentralized P2P system. To sum up, our paper makes the following contributions in the area of P2P security:

- We present TREMA, a general solution for trust management in P2P systems based on a tree structure. Besides, we expose a set of APIs that allows P2P applications to work on top of TREMA.
- We show how to augment a tree with extra links to create robustness and to allow nodes to exchange queries without overwhelming the root. These extra links help to eliminate the problems of bottlenecks and single points of failures in the tree structure.
- We present two possible implementations of TREMA. One is an extension of BATON [7], an existing tree structure. The other is HICON, a novel tree structure. We compare these two systems and show how to improve the weaknesses of both systems.
- We conduct an experimental study on the implementation of TREMA in BATON to evaluate the effectiveness and efficiency of our proposed solution.

This paper is an extended version of a previous paper [8]. In the previous paper, we introduced a secure protocol for trust management in P2P systems based on BATON [7], a tree structure. In this paper, we extend the idea to support all tree structures and also provide further design discussion. The rest of the report is organized as follows. In Section II, we introduce related work in the area of trust management in P2P systems. In Section III, we explain the basic design of TREMA in terms of the trust and security models. In Section IV, we discuss some issues of our basic design, and suggest possible solutions to improve it. In Section V, we describe the general APIs that we are proposing. In Section VI, we first use our design to extend an existing tree structure (BATON) to support reputation management. After that, we present our own tree structure design (HICON). We suggest potential applications of TREMA in Section VII. Section VIII describes our experimental study and its results. Finally, in Section IX, we summarize the important contributions of our design and its potential.

II. Related Work

Trust management in P2P systems can be classified into two main categories: credential-based and reputation-based management. Credential-based management systems employ the classical method where a peer trusts another peer after examining the other peer's credentials. If the credentials satisfy the peer's policy, that peer can be trusted in a transaction. Otherwise, the peer would refuse to be in a transaction with the other peer. The weakness of this method is that it has to rely on servers for keeping every peer's credentials, which is not entirely a scalable method. Moreover, since credentials are usually generated once and stored, past transactions of peers, both good and bad, are not considered. As a result, this method is only suitable for specific kinds of systems with fixed credentials, like access control systems. Examples of systems that apply this trust model include X.509 [9], PGP [10], PolicyMaker [11] and its successors, REFEREE [12] and KeyNote [13].

On the other hand, reputation-based management systems rely on reputation to evaluate the trustworthiness of a node. In general, the reputation of a node is computed based on its previous transactions with other nodes in the system and how they rated these transactions. Reputation-based management systems can be further classified into two subcategories. One type of system considers only the reputation of an individual, like those in [3], [4], [5], [6], [14], [15], [16], while the other takes into account social relationships between nodes in addition to individual reputation, such as [17], [18]. Since no nodes know of all nodes in the system, reputation of nodes have to either be collected and stored on servers for reference or distributed to all nodes in the network by the gossiping algorithm. Both of these methods are not viable for large networks because the first method is not scalable while the second method is expensive.

In the field of data structures, the structure of a tree has a very important role. NICE [19] can be used to do scalable application layer multicast [20] by using the idea of overlay trees for efficient content distribution. However, very few networks proposed so far uses the topology of a tree. In this kind of structure, if the standard query processing algorithm is used, nodes near the root will be accessed many times more compared to nodes near the leaves, and hence congestion at the root or nodes near the root may happen. This is not acceptable in P2P systems. To avoid this problem, P-Tree [21] suggests a use of partial tree structure. In this method, each leaf node in the tree is represented by a P2P node while internal nodes are all virtual. Each P2P node maintains a path from the index root to the leaf node. As a result, queries can be processed at any node without pushing all queries to a special node. Note that, however, if a node has to maintain the whole tree structure, the maintenance cost is very expensive and not suitable for P2P systems. Alternatively, BATON [7] creates links between nodes at the same level in the form of routing tables. Consequently, queries can be processed at any node in the tree without going through the root. Nevertheless, these systems focus only on range query processing, and not trust management.

III. Basic Design

A. Trust Model

TREMA consists of peers arranged by their reputation. Peers of higher reputation occupy positions at higher levels in the tree, with each parent having a higher reputation score than its children, and so the root node is the peer with the highest reputation. Peers of higher reputation are accorded higher privileges of some kind, to provide incentive

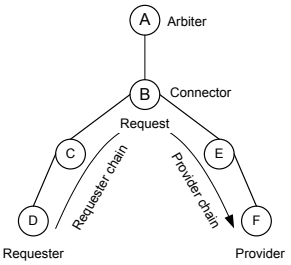


Figure 1. Trust relationships in a trust route

for nodes to increase their own reputation. We develop the following terminology and use it to present the model of trust relationships between nodes in the tree.

- *Trust link*. A link exists between a peer and its child, and this denotes a link of trust. We say that (1) the child peer in this link trusts its parent because the parent has a higher reputation than itself, and (2) the parent is answerable for the child. The latter point means that any misbehaved action on the part of the child reflects poorly on the parent as well, and the parent is also held accountable for any misbehavior of the child. This is desirable because it is every peer's responsibility to minimize the presence of malicious peers entering the network as children. Trust links are inherently transitive, because a child that trusts its parent would also trust its parent's parent of higher reputation, while a parent is accountable for its children and thus its children's children as well.
- *Trust chain*. A chain of trust is formed by consecutive trust links. In such a chain, we say that the lowest peer trusts the highest peer, based on transitivity of trust in our model.
- *Trust route*. A trust route is the path between any two peers in the tree. It is composed of one or two trust chains that meet at a common ancestor of the two nodes. We call that ancestor the *connector* of the route. The trust route also includes the connector's parent, which we label as the *arbiter* of the route. A trust route is formed when a peer requests content from another peer in the system. The former peer is known as the *requester*, and the latter is the *provider*. The trust chain from the requester to the connector is called the requester chain, and that from the provider is called the provider chain. Figure 1 illustrates the relationships mentioned here.
- *Transaction*. A transaction is initiated by a requester, by sending a request through the tree to a chosen provider. The provider responds with the appropriate content to the requester. Transactions occur over a trust route in our tree, and they have a *transaction outcome* in the form of a report sent out by the requester. A positive outcome indicates a successful transaction when the requester is satisfied with the received information. Conversely, a negative outcome indicates a failed transaction when the requester is not satisfied with some part of the received information.
- *Rewards and punishments*. To give a reward means a peer increases the reputation score of a child peer, and a punishment is the converse, a decrease in the reputation score of the child peer. Rewards and punishments are managed based on the transaction outcomes reported by requesters.

B. Trust Management

This subsection describes how trust in our model can be managed. There are two possible outcomes of transactions each of which is dealt with in a separate way.

- *Successful transactions*: if a successful transaction occurs between two nodes via a trust route, parent nodes would reward the child nodes. The rationale is that rewarding a child would allow it to be trusted by more nodes, and hence to increase its potential for bringing in more transactions for itself. This would lead to more opportunities for the parent node to earn its own rewards. In general, after a successful transaction, the arbiter rewards the connector, the connector rewards both the children in the requester and the provider chains, and so on, downward both trust chains. The only exception is the requester, which does not get any reward for initiating a request, since it adds no value to the network.
- *Failed transactions*: for a failed transaction, the converse happens. The arbiter punishes the connector, which in turn pushes the blame downward the tree from parents to children in both chains. The requester again is unaffected by the punishments because it has nothing to gain or lose for accurately reporting the outcome of the transaction. A truthful report would, however, increase the effectiveness of the whole network. To prevent the malicious scenario of a node deliberately reporting multiple failed transactions, a parent might keep track of node failure reports, and identify any nodes that are misbehaving in this way. The parent could then terminate trust links with any evil node, deeming it to be deliberately causing trouble by either requesting content from reputably bad nodes, or inaccurately reporting many failed transactions.

This design leads to two main implications. On the one hand, nodes will try to maximize the number of successful transactions and minimize the number of failed ones, in order to optimally increase their reputation. This selfish and self-centered behavior, however, allows for optimal gains for the system as a whole, because each node selfishly seeks to maximize its own rewards and to do so, it has to shrewdly monitor its children and their behavior in transactions. On the other hand, a node would quickly break off links with children that result in many failed transactions and refuse to forward transactions from such nodes, because it is being

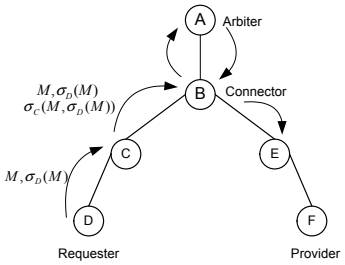


Figure 2. Message forwarding over a trust route. For brevity, we only show the first two of a chain of signed messages.

held accountable for the behavior of its children. At the same time, a node would be willing to forward requests and content from reputable nodes or new nodes because doing so would give it the potential to increase its reputation.

C. Security Management

We claim that two nodes in TREMA can exchange messages over a trust route in an accountable and authenticated manner as follows. When the requester sends a request, the request is forwarded via the requester chain of trust. Forwarding a message means signing the message and sending it over a trust link. Here, a child node always trust its parent to forward the request properly, and a parent node is willing to be accountable for its children because it stands to gain in reputation if the transaction is successful. Note that if the transaction fails, the parent could always terminate the link to its children.

In case of the arbiter, when it receives a message that is signed by one of its children, which is the connector of the trust route. The arbiter signs that message, indicating it has seen the request and is willing to trust that the connector child will do a proper job forwarding that request. The signed request is then sent back down to the connector, who continues forwarding it down the provider chain to the provider. At the provider chain, each node is willing to forward the message down the tree because the message has been signed by its parent, whom it trusts. The provider upon receiving the request can obtain the entire trust route by observing the signatures on the request. When the provider responds with content, it sends both the original multiply-signed request and the content signed by itself. This message is forwarded back via the same trust route, and the reverse happens, with each node signing the message and forwarding it, including the arbiter again. The incentive for forwarding remains the same: a node could stand to gain from aiding a successful transaction, and is able to exert control on any misbehaving children.

Finally, once the requester receives the requested content, it first evaluates the transaction and assigns the transaction either a positive or negative score, and then sends this score in an encrypted message to the arbiter, via a direct connection. Based on the received score, the arbiter will give the appropriate reward or punishment for the transaction.

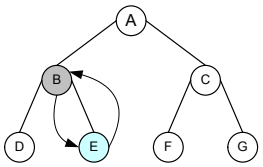


Figure 3. Swapping positions between nodes

Figure 2 shows an example of message forwarding over a trust route. Note that requests and complaints are also communicated securely and authentically by encrypting then signing the messages. This design ensures that only the correct nodes would be able to read the messages.

D. Node Ranking Management

If we want to know reputation score of a node, we have to ask its parent, since the parent in our tree is of higher reputation and is thus more trustworthy. If an internal node cannot accomplish its task or turns malicious, we should replace it with a better node. Since a node may never want to step down from its position, we have to exert control over that node through its parent.

Additionally, reputation scores of a node is not only stored at the parent but also at the grandparent. Consider the situation where a node now has a reputation lower than that of its child, implying that the tree is currently not well-formed. The solution to this situation is to swap the positions of these two nodes through a swap operation, and that can only be done from the position of the parent of the ill-placed node. By changing positions, these nodes also exchange knowledge information of their children and reputation of these children they are keeping.

An example of node swapping is shown in Figure 3 in which node B has to swap its position with its child E because E has a higher reputation. Actually, since A knows reputation of all B, C, D, E, F, G, it can also swap positions between B and G if G has a better reputation than both B and E. This sort of swapping can be done if A wants to assign a node that is known to be trustworthy from another subtree to be the parent of a subtree that could possibly contain colluding malicious nodes.

E. Concerns of a Tree Structure

Tree structures are widely used in computer systems. They are especially good for storing and retrieving data. Among various data structures, the tree is a dominant structure used in the area of databases. However, there are usually some concerns when employing tree structures in distributed systems. Below are the three greatest concerns.

- Bottlenecks and single points of failure could exist at the root or at nodes near the root in a tree. If every query has to reach the root node before it is processed, the root will become overloaded by queries. As a result, this would be a problem in large networks.

- Usually, the cost of query processing (in terms of the number of search steps) is bounded by the height of the tree. Therefore, if the tree is skewed and unbalanced, the cost of searches might be high.
- In a weakly-connected tree structure, failure of a node may partition the tree completely. As a result, in distributed systems, trees are often augmented with extra links to avoid this problem.

Keeping in mind these concerns, we will design a system without such problems in Section VI.

IV. An Improved Model

The above basic model works well under an assumption that the information given by a node to another node about its children is always correct. In other words, all internal nodes can be trusted in giving information. This is because if an internal node is bad, it can return wrong reputation results about its children to other nodes. For example, a malicious node could return a good reputation score about a bad node or a bad reputation score for a good node. To avoid the problem of the basic model, we introduce a new type of score called a *reference score* for internal nodes. The reference score is used to reflect exactness of information a node gives to others. Now, the trust value of a node is based on not only its reputation score but also the reference score of its parent. In other words, if a node always gives correct information about its children to others, we should trust its information. However, if a node often makes mistakes or gives incorrect information deliberately, our trust in information provided by that node is reduced. Similar to reputation score, a reference score of a node is stored at its parent. So now, as illustrated in Figure 4, before each transaction, a node y should find not only x's reputation score, which is stored at z, the parent of x, but also z's reference score, which is stored at t, the parent of z and after each transaction, y updates scores for both x and z.

The problem now is how to evaluate correctness of information received from z to give feedback of a score after a transaction. Here, we propose a simple solution as follows. When a node is asked about reputation of its children, in addition to giving the total reputation score, it also gives the standard variation of the scores calculated from previous transactions. As a result, the correctness of received information is evaluated by both the reputation score and the standard variation. For example, if the result of the transaction falls far away outside the standard variation, the node giving information should be rated with a bad reference score.

That is not all. Assume that in the worst case, x, z, and t are all malicious peers and they cooperate with each other. If t gives a wrong reference score for z while z gives a wrong reputation score for x, y would still be cheated. To further enhance security, y can also ask reference score of t from its parent. In general y asks for reference scores of a chain

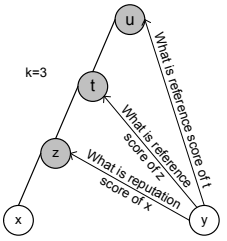


Figure 4. A k=3 reference chain

of k ancestors of x in which k is a configurable parameter of the system. Note that since these k nodes form a chain, the cost of lookup algorithm and update algorithm is just $logN + k$. By setting k with a large number, the system becomes strong against collaborative malicious peers. A worry is that k may have to be large, and hence it may be costly. However, since nodes in the system cannot determine the location of them in the tree structure, they have to follow the join algorithm, which scatters nodes along the system to make the tree balanced. As a result, forming a long chain of malicious peers connected by parent-child links is not easy. An example of a $k = 3$ reference chain is shown in Figure 4 in which y asks z for reputation score of x, t for reference score of z and u for reference score of t.

Another technique which can be used by a group of malicious nodes to trick other nodes is to create fake transactions and report good results to their parent to increase their reputation score. To avoid this problem, we just use a simple technique in score calculation as follows. First, we do not simply consider the number of successful transactions as the score. Instead, we limit the score at a maximum value, and the score of a node can only reach that maximum score. Second, we calculate not only the number of successful transactions but also the number of *different* successful transactions of nodes. By "*different*", we mean that transactions of the node that are done with different nodes. As a result, even though a node may have many good transactions with a specific node, it still has a low score if it has many other bad transactions with other nodes.

V. APIs

Based on the design of TREMA as described in the previous sections, the general APIs we need to provide can be divided into three categories. The first category contains APIs for trust management, which perform the behaviors discussed in our earlier sections. The second category is a set of indirection APIs, which are function calls that applications might need to call. These APIs also provide indirections to the lower P2P network layer. This design allows us to keep the underlying layer completely hidden from the applications that use our framework. Furthermore, with this separation, it is less likely that applications would make incorrect calls that have not been secured by the framework. The third set includes APIs of the underlying P2P

network that are augmented with additional functionality to support TREMA. We respectively discuss APIs in these three categories in the rest of this section.

A. Trust Management APIs

- *Reputation-Query*: this API takes in a target node and returns the reputation of that node by querying its parent over a trust route. This reputation request is considered as a transaction and will have an accompanying transaction outcome report from the requester.
- *Reputation-Update*: this API is called by a node that wishes to update the reputation of its child. The update is done by sending the request up to the k reference chain, and upon getting the approval proceeding to change the reputation of the child.
- *Reputation-Complaint*: this API is called by a node to lodge an official complaint up to the k reference chain when the node feels that its reputation has been unfairly modified.
- *Transaction-Report*: this API reports the outcome of a particular transaction to the arbiter of the trust route. Given the report, the arbiter decides whether to direct a cascading series of rewards or punishments.

B. Indirection APIs

- *Node-Find*: in general, when a new node joins the system, it needs to perform bootstrapping. In most decentralized P2P systems, this action involves finding and connecting the node to an existing node in the system. While this function is usually provided at the P2P network layer, we use an indirection API to securely expose it to upper layer applications.

C. Augmented APIs

- *Node-Join*: this API is called by a node that wishes to join the network. The API allows the new node to find its position in the tree structure, and is only triggered after calling *Node-Find* described above. Basically, upon receiving the contact node from *Node-Find*, the new node sends a "Join" message to the contact node. If that node is the correct parent, the new node is accepted as its child. Otherwise, the contact node forwards the request to either its parent or a child that is more suitable. In this manner, the request can be forwarded through the tree until the correct parent of the new node is found within $O(\log N)$ steps. Note that for correct forwarding, the tree needs to have positional determinism, where the position of a node can be determined given the state of the network. Both our proposed tree implementations support this.
- *Node-Depart*. this API is called by a node when it wishes to leave the network. This allows the system to establish new tree links and close down old ones where applicable. In a best-case scenario without failures, calling this API allows efficient updating of any routing table and link, and minimizes disruption to the network due to nodes leaving the system.
- *Node-Failure-Discovered*: this API is called by a node that discovers one of its neighbor nodes is not responding, presumably because that neighbor has failed. Calling this API would set off the appropriate measures to confirm that failure and establish new links as if that failed node had called *Node-Depart*.

VI. System Design

At this point, with the APIs firmly laid out, we are able to describe in greater detail how to extend TREMA to use in two tree implementations: BATON, an existing tree-based framework and HICON, a novel scheme we propose in this paper. In essence, we try to place TREMA on top of existing networking frameworks that provide the topology of a tree. The challenge here is to ensure that we can effectively and efficiently implement our proposed trust management APIs above, and also use the desirable properties of these frameworks to address the weaknesses of the basic tree.

A. BATON

In this section, we first describe the structure of BATON. After that, we introduce the way to deploy TREMA on it.

1) Basic Structure

In BATON, each peer participating in the network is responsible for a node in the tree structure. The position of a node in the tree is determined by a pair of a *level* and a *number*. The level specifies the distance from the node to the root while the number specifies the position of the node within the level. BATON uses three kinds of links to make connections between nodes: parent-child links are used to connect children and parents; adjacent links are used to connect adjacent nodes; and neighbor links are used to connect neighbor nodes at the same level having a distance 2^i from each other. Neighbor links are kept in two special sideways routing tables: left routing table and right routing table. An example of a BATON tree is shown in Figure 5. Note that in this figure, only neighbor links of the grey node are shown.

BATON controls the balance of the tree by forcing that if a node has a child, it has to have a maximum number of possible neighbor links within its level, or in other words, have full routing tables. As a result, when a node receives a join request from a new node, it can only accept the new node as its child if it has full routing tables. Otherwise, depending on the condition, the request is forwarded to either its parent, its neighbor or its adjacent node. In case of node departure, if a node is a leaf node and none of its neighbors has children, it can leave the network. Otherwise, if it is either an internal node or a leaf node, whose neighbors have children, it has to find a replacement node, which is

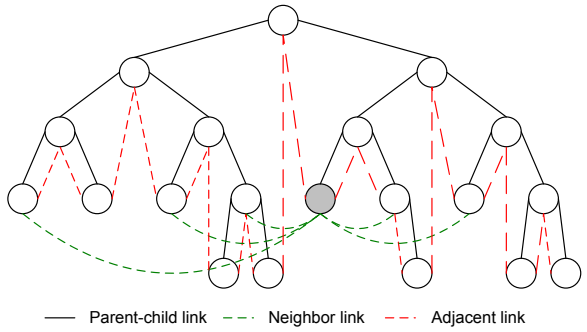


Figure 5. BATON structure

a node in the previous case to replace its position in the tree structure. In particular, by employing sideways links, BATON does not suffer any of the tree issues discussed in Section III-E.

2) TREMA Deployment

Since the most important issues in deploying TREMA are how reputation of a node is looked up and how transaction results are reported to responsible nodes, we focus our discussion of these issues.

- *Reputation lookup*: before each transaction, nodes exchange information about their location in the tree to each other. Knowing the location of a node x, its partner y can infer the location of x's parent, which is z as below:

$$z_{level} = x_{level} - 1$$

$$z_{num} = \begin{cases} x_{num}/2 & \text{if } x_{num} \text{ is even} \\ (x_{num}+1)/2 & \text{if } x_{num} \text{ is odd} \end{cases}$$

  Note that in the tree structure, the level is setup increasingly from the root to the leaf starting at 0 while the number is assigned from the left to the right of each level starting at 1. Now, knowing the location of z, y can issue a query to lookup x's reputation towards z. The algorithm of sending a query towards a node knowing its location is represented as in Algorithm 1. Since at each step, this algorithm makes the search space reduce by half, it is guaranteed that after maximum $O(logN)$ steps, the query should reach the destination node z. When z receives the query, it returns the reputation score of x to y. Note that if x does not tell a truth about its location, and hence when y issues the query either z can not be found or z is not the parent of x. As a result, x can be considered as a bad node.
- *Transaction result report*: after each transaction, a similar process is done to report the result of the transaction between partners to their parent. In particular, each peer rates the transaction by giving its partner a score in a range of [-1.0, 1.0]. Depending on the level of satisfaction or dissatisfaction, a value is given in which a positive score is used to indicate a good transaction while a negative score indicates a bad transaction.

Algorithm 1 :Query (level l, number n, node z)

```
l_node = level of the current node
n_node = number of the current node
if l_node = l then
  t = the nearest node to z
  t.Query(l, n, z)
else
  if l_node > l then
    t = a child of the current node
    t.Query(l, n, z)
  else {l_node < l}
    t = parent of the current node
    t.Query(l, n, z)
  end if
end if
```

B. HICON

In this section, we first introduce the basic design of HICON. After that, we show how to deploy TREMA on HICON.

1) Basic Structure

HICON stands for Hierarchical IP Clustering Overlay Network. Nodes in a HICON tree are arranged such that we have the following condition hold for every pair of nodes x and y in the network:

- x is an ancestor of y if and only if, for their common IP prefix P, x has the highest reputation among all nodes with the same IP prefix P.

This condition ensures that for a given state of all nodes in the network, there is a fixed and logical tree structure based on the IP addresses and the reputation scores of nodes, and each node will have its own fixed position in the tree. We call this hierarchical IP clustering because we can build this tree by first clustering all nodes into groups with the same prefix, and then appointing a parent node with the highest reputation as the leader in each cluster. This process can be repeated for clusters of other prefix lengths until we get a complete tree.

By varying the lengths of the prefixes considered, we can change the expected height of the tree and the maximum number of children a node might have. As an example, Figure 6 shows nodes in a HICON tree where we have considered prefixes of 1, 2, 3 and 4 bytes, i.e., 8-bit increments. We see that the node A with IP address 18.4.6.5 has reputation 1000 and is the ancestor of all nodes with IP prefix "18.*". Node B is a child under node A but is the parent of the cluster of nodes with IP prefix "18.4.*", which includes nodes E and F.

In general, in IPv4, if we consider prefixes of B-bit increments, the expected height of the tree will be $32/B$

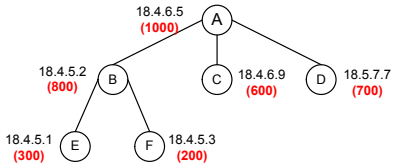


Figure 6. Nodes in a HICON tree

and the maximum fan-out would be 2^{B+1}. This is useful because the height and fan-out of this tree is configurable based on the parameter B. In IPv4, having $B = 8$ gives a very short tree of height 4, but in IPv6, this would give a reasonable height of 16.

The basic structure in HICON gives us the properties of scalability, and some efficiency that a tree inherently has. However, in order to achieve robustness to network dynamism, we introduce the idea of successor nodes and links. In the improved design, in addition to links a node x has to its parent and children, x has a *successor link* to the child node that has the highest reputation among its children. This child y is the *successor node* of x. When x departs or fails, y will take over the position of x. Besides, *successor links* are also maintained among y and its siblings (other children of x), and between y and its potential parent z, which is the parent of x (z will be the parent of y if y comes to replace the position of x). Note that within the trust model, all these successor links are also trust links over which communication can happen, and are recognized and acknowledged by nodes on both sides of the links.

3) TREMA Deployment

For TREMA to work with HICON, we need to implement the augmented network APIs in the following way.

- *Node-Join*: to implement the *Node-Join* procedure, we first ignore the reputation score of x and determine the position of a new node x simply based on its IP address. In particular, when a node y receives a join request from x, if x has some common IP prefix P with y and P is within the cluster that y is the leader of, x is a descendant of y. In this case, y searches its children to find the one whose IP has the longest matching prefix with the IP of x, and forwards the join request of x to that node. If no such child exists, y accepts x as its child. In the other case, if x does not have common IP prefix with y or x does not belong to the cluster of y, y forwards the join request of x to its parent node, who has a broader view, to process the request. Finally, once x is accepted as a child of a node z in the tree, the reputation score of x is used to compared to the scores of other children of z so that x is put it at the correct position.
- *Node-Depart*: when a node x is leaving, x informs its neighbor nodes of its intended action, so that they can update their link information. The successor child of x at this point proceeds to take over the position of x, and uses successor links to establish connections with its new parent and children. Links from x are then removed from the network.
- *Node-Failure-Discovered*: when the failure of a node x is discovered, the discovering node confirms this with the parent of x, who can then proceed to find the successor child of x to take over the position of x. This reduces to the situation of a node departure, with the successor child taking over the position of x.

Additionally, we need to implement these methods for trust management.

- *Reputation-Query*: given a target node x, the requester formulates its query and forwards the query to the parent of x. This is done in a method similar to *Node-Join* where we try to find the position of a node. The same procedure is used here to determine who is the parent of x and how to forward the query to that node. In particular, when a node y receives a query for the reputation of a node x, if x is a descendant of y (i.e., x has a common prefix P that is within the cluster of y), the parent of x must be in the subtree of y. In this case, y chooses the appropriate child node to forward the query to. Otherwise, y forwards the query to its parent, who can continue to forward the query to the correct parent of x. Finally, once the query reaches the parent of x, a trust route is established, and that route can be traversed in reverse to get the response back to the requester.
- *Reputation-Update*: there is no special implementation needed for this tree, we just need to follow the general APIs' specification.

C. Comparison of BATON and HICON

In this section, we make a comparison about the two tree structures presented above. In general, there are three main differences between these structures, as described below.

- HICON is based on a multi-way tree structure whose fanout is 32 for prefixes of 4-bit increments, while BATON is based on a binary tree structure whose fanout is fixed at 2. As a result, in terms of reputation lookup boundary, HICON is better.
- In HICON, reputation is looked up through a chain from child to parent to child. As a result, if nodes are in different branches of the tree, the query has a high potential to be forwarded to the root or nodes near the root, and hence bottlenecks as well as single points of failure may still be problems (even though it is not as severe as in centralized servers based systems). On the other hand, BATON employs sideways routing tables in the process of reputation lookup, which can avoid forwarding the request to higher level nodes.
- In HICON, nodes are grouped by their IP address while in BATON, nodes are distributed randomly to guarantee the balance of the tree.

D. Further Improvements for Tree Structures

From the above comparison between BATON and HICON, we can see the advantages and disadvantages of each tree structure compared to the other. As a result, in this section, we propose further techniques to improve the two tree structures.

- BATON can employ a higher fanout tree structure as that in HICON. The reason why BATON is based on a binary tree structure is because a higher fanout tree structure makes it more difficult to manage tree nodes to support range queries, which is the main motivation of BATON. In our system, since our target is to support reputation management, it is possible to extend BATON to support a higher fanout tree structure. On the other hand, HICON could leverage the idea of sideways links (routing tables) in BATON to provide sideways travel of reputation messages, and hence it can totally eliminate the problem of bottleneck and single point of failure as BATON does.
- Another technique, which can be used for both systems is to allow a node to have multiple parents. This technique makes the system less susceptible to being partitioned in case of a massive failure.
- An interesting feature of HICON is that nodes are grouped by their IP address. Based on this feature, we can employ a special kind of reputation called social reputation in HICON. As pointed out in [18], [17], the reputation of a node is affected by the reputation of the society it belongs to and vice versa. The design of HICON is very suitable to employ this kind of reputation scheme since all nodes in the same group have the same prefix IP address, and hence they usually come from the same organization or region.

VII. Potential Applications

A. General P2P Applications and File-sharing Systems

Currently, file-sharing applications are dominant in P2P applications. All well-known P2P applications such as Gnutella [22], Napster [23], BitTorrent [24] are file-sharing systems. A security challenge in these systems is that when a peer wants to download a file, it issues a query and may receive a lot of answers from other peers. What is a good peer that the peer issuing the query wants to download the file from? How can a peer be guaranteed that it does not download a wrong file or a file containing virus from another peer? A reputation management system like TREMA can be used in this case to answer these two questions. Furthermore, reputation can also be used to evaluate the quality of services. As an example, a node with high reputation is not only a trustee node but also a node which can provide high bandwidth for downloading.

B. Spam Reporting Systems

If anyone wants to build a distributed spam reporting system that is similar to Blue Frog [25], developing from TREMA will be a good start. In general, this system will consists of two basic operations: spam reporting and spam lists forwarding. In particular, when a node receives a spam, it will report the spam and the contact information to a subset of nodes in the network. The recipient nodes will then decide for themselves if the given thing is really a spam and if the contact information is matched, given both the message and the reputation of the claimant. If the report is correct, they will add the spam to their spam lists and forward these lists to other nodes, together with the identifier of the one who claimed it. Finally, they will update the reputation of the sender. Usually, these spam lists could be sent around, and some way of compounding reputations will be worked out. In this way, every node has a list of email addresses it thinks spams, and the contact information thereof. It is interesting to note that without a careful management of reputations, this system may devolve into a way to DDoS any website a malicious node wants to.

VIII. Experimental Study

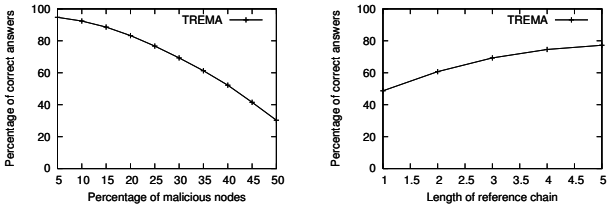
To evaluate the performance of our proposal, we have implemented an extension of BATON [7] to support our security protocol. We tested our system in a network of 1,000 nodes, where exists two kinds of nodes: good nodes and malicious nodes. We just make a simple assumption that that good nodes always do good transactions and give correct answers if they are asked for reputation of their children. On the other hand, malicious nodes always do bad transactions and give incorrect answers about reputation of their children.

A. Effect of Varying Number of Malicious Nodes

We first evaluate the effect of varying number of malicious nodes on the strength of the system. The result is displayed in Figure 7(a) in which the x-axis presents the percentage of bad nodes in the system while the y-axis presents the percentage of correct answers about reputation of nodes. The length of reference chain in this experiment is fixed at 3. The result shows that our system can suffer up to 20% of malicious nodes while still provide good answers for a reputation of nodes: more than 80% of answers is correct. It is because in order to fully cheat other nodes, malicious nodes have to form a subtree height greater than 3. However, it is difficult to do that since nodes are distributed equally in the leaf level to keep the tree balance.

B. Effect of Varying Length of Reference Chain

In this section, we vary length of reference chain from 1 to 5 while keeping the percentage of malicious nodes at 30%. The result is displayed in Figure 7(b). The result confirms that the system increases its strength with the increasing length of reference chain.



(a) Effect of varying number of malicious nodes

(b) Effect of varying length of reference chain

Figure 7. Experimental results

IX. Conclusion

In conclusion, in this paper, we proposed TREMA, a general solution for reputation management in Peer-to-Peer systems based on a tree structure. By using a tree structure, TREMA can avoid the high cost of broadcasting messages that is seen in gossiping-based solutions. At the same time, TREMA does not suffer the problem of bottlenecks and single points of failure as seen in server-based solutions through the employment of extra links in the tree structure. We came up with two specific tree structures to implement TREMA on. One is extended from BATON [7]. The other, HICON, is a novel design. We made a comparison between these tree structures, showing their advantages and disadvantages. From there, we suggested further improvements to both. Finally, we conducted experiments on the implementation of TREMA on BATON to show the effectiveness and efficiency of our proposed solution.

References

[1] eBay, "http://www.ebay.com/ (last accessed: Jan 09, 2012)."

[2] Amazon Auctions, "http://auctions.amazon.com/ (last accessed: Jan 09, 2012)."

[3] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "EigenRep: Reputation Management in P2P Networks," in *Proceedings of the 12th WWW Conference*, 2003, pp. 123–134.

[4] S. Lee, R. Sherwood, and B. Bhattacharjee, "Cooperative peer groups in nice," in *Proceedings of the 22nd INFOCOM Conference*, 2003, pp. 1272–1282.

[5] B. Dragovic, B. Kotsovinos, S. Hand, and P. R. Pietzuch, "Xenotrust: Event-based distributed trust management," in *Proceedings of the 2nd International Workshop on Trust and Privacy in Digital Business*, 2003, pp. 410–414.

[6] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, no. 7, pp. 843–857, 2004.

[7] H. V. Jagadish, B. C. Ooi, and Q. H. Vu, "Baton: A balanced tree structure for peer-to-peer networks," in *Proceedings of the 31st VLDB Conference*, 2005, pp. 661–672.

[8] Q. H. Vu, "SPP: A Secure Protocol for Peer-to-Peer Systems," in *Proceedings of the 2nd International Conference on Advances in P2P Systems (AP2PS)*, 2010, pp. 1–6.

[9] International Telegraph and Telephone Consultative Committee (CCITT), *The Directory - Authentication Framework, Recommendation X. 509*, 1993 update.

[10] P. Zimmermann, *PGP Users Guide*. MIT Press, 1994.

[11] M. Blaze and J.Feigenbaum, "Decentralized Trust Management," in *IEEE Symposium on Security and Privacy*, 1996, pp. 164–173.

[12] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss, "REFEREE: Trust management for Web applications," *Computer Networks and ISDN Systems*, vol. 29, no. 8–13, pp. 953–964, 1997.

[13] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, *The KeyNote Trust Management System, Version 2*. RFC-2704. IETF, 1999.

[14] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," in *Proceedings of the 9th International Conference on Information and Knowledge Management*, 2001, pp. 310–317.

[15] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," in *Proceedings of the 11th WWW Conference*, 2002, pp. 376–386.

[16] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 2002 ACM Conference on Computer and Communication Security*, 2002, pp. 207–216.

[17] J. Pujol and R. Sanguesa, "Extracting reputation in multi agent systems by means of social network topology," in *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2002, pp. 467–474.

[18] J. Sabater and C. Sierra, "REGRET: A Reputation Model for Gregarious Societies," in *Proceedings of the 4th Workshop on Deception, Fraud and Trust in Agent Societies*, 2001, pp. 61–69.

[19] NICE, "http://www.cs.umd.edu/projects/nice/ (last accessed: Jan 09, 2012)."

[20] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 205–217, 2002.

[21] A. Crainiceanu, P. Linga, J. Gehrke, and J. Shanmugasundaram, "Querying peer-to-peer networks using P-Trees," in *Proceedings of the 7th WebDB*, 2004, pp. 25–30.

[22] Gnutella, "http://www.gnutella.com/ (last accessed: Jan 09, 2012)."

[23] Napster, "http://www.napster.com/ (last accessed: Jan 09, 2012)."

[24] BitTorrent, "http://www.bittorrent.com/ (last accessed: Jan 09, 2012)."

[25] Blue Frog, "http://sourceforge.net/projects/bluefrog/ (last accessed: April 12, 2006)."

A New Pattern Template to Support the Design of Security Architectures: A Case Study

Santiago Moral-García[1], Roberto Ortiz[2], Santiago Moral-Rubio[2], Javier Garzás[1, 4] and Eduardo Fernández-Medina[3]

(1) Kybele Group. Dep. of Computer Languages and Systems II.
University Rey Juan Carlos, Madrid, Spain.
{santiago.moral, javier.garzas} @urjc.es
(2) Dep. Information Security. BBVA, Madrid, Spain.
r.ortizpl@gmail.com, santiago.moral@bbva.com
(3) GSyA Research Group. Dep. of Information Technologies and Systems.
University of Castilla-La Mancha, Ciudad Real, Spain.
eduardo.fdezmedina@uclm.es
(4) Kybele Consulting, Madrid, Spain.
javier.garzas@kybeleconsulting.com

***Abstract*—New work paradigms are emerging in the information technology sector, which are causing changes in the technological infrastructures of organizations' information systems. Organizations should adapt to all these changes in order to guarantee the confidentiality, integrity and availability of their information assets. Organizations should therefore seek support from security architectures. A good means to design security architectures is through the use of security patterns. After carrying out a systematic review of security patterns, we observed that the vast majority of current security patterns are oriented towards the production of security mechanisms, such as secure access systems or secure authentication systems. This type of patterns may be extremely useful to those security engineers who work on the production of this type of mechanisms, but they cannot be applied by a wide sector of security engineers who work in the development of security architectures. In a previous work, we proposed a new pattern template in order to complement security patterns and make them more applicable to security architecture design environments. In this paper, which is an evolution of the work mentioned above, we have validated the proposed template with a case study. This case study also provides a new security solution to ensure external accesses to organizations' production environments.**

Keywords-information security engineering; security architectures; security technologies; security patterns; real environments.

I. INTRODUCTION

The globalization of the Information Technology (IT) sector has encouraged the appearance of new paradigms in the traditional role of software development companies. This situation has been motivated by the advancement in communication systems and the need for organizations to reduce costs. These new paradigms are based on outsourcing the role of software developers in areas or regions in which labor costs are cheaper, thereby optimizing the profit margins of those organizations that hire these services.

Although the paradigms within the IT industry are changing and organizations have decided to outsource some of their services, they must ensure the confidentiality, integrity and availability of their information assets [13]. In view of the fact that the realization of this task should consider the constant evolution of the organization's setting [27], we should specifically consider the variation between people, technologies, risks, processes, volumes of information, business strategies, etc. The need therefore exists to adapt the organization to all these changes in order to guarantee the fundamental security properties for their assets [21]. It is not easy for an organization to evaluate its level of risk and adapt itself to permanent changes. It is therefore vital for it to seek support from a security architecture [3] in order to mitigate the impact of these changes and thus minimize the risks associated with each of them.

The concept of security architecture can be defined as complete, structured, coordinated and rigorous designs of information systems that support business processes in order to reduce the risk of confidentiality, integrity and availability when managing its information assets [19]. Security architectures are installed with the intention of minimizing the risks associated with the use of information technologies and optimizing an organization's business process and strategies. If this objective is to be achieved, it is necessary to establish a set of technological infrastructure controls with which to identify the security mechanisms that are needed to define the system's security.

The concept of security mechanism can be defined as artifacts designed to prevent, detect and respond to information security incidents, in order to manage and reduce the confidentiality, integrity and availability of business processes' information risks [26]. A security mechanism cannot be used in isolation to protect a business process, but a

wide set of security mechanisms can reduce security risks when managing information assets in a business process. A security architecture consists of a wide set of security mechanisms, which is complete, structured, coordinated and rigorous.

Security patterns are a good way to design security architectures because they describe a recurring problem, providing a documented and validated solution that can be used multiple times, and they combine experience and good practices in the design of information systems. After carrying out a systematic review of the literature related to security patterns, we discovered that the vast majority of current patterns are focused on supporting the construction of new security mechanisms [12, 28]. These patterns are a useful support for those engineers who work developing security mechanisms, which are the basic elements of an architecture [8, 17, 23]. However, it is difficult to apply most of them to those work environments that are focused on the analysis and design of security architecture, since they do not consider that several security mechanisms must be used to solve a security problem and they do not consider the details of installing the solution in real complex systems [15]. We understand a real complex system to be all those elements that are involved in an organization, i.e., human resources, business processes, technologies, etc.

The lacks detected in current security patterns led us to believe that it was necessary to define a new description template of security pattern with which to resolve these limitations. In order to complement current security patterns, in a previous work [16], we defined a new pattern template with which to define security patterns, characterized by the fact that it includes all the aspects that are necessary for a simple and reusable definition of security architecture designs.

In this paper, which is an evolution of the work mentioned previously [16], we have developed a case study with the aim of checking the pattern template's validity. This case study has also provided us with a security solution to ensure external accesses to organizations' production environments. Finally, the security solution obtained has been deployed in a financial organization, thus helping us to ensure the security of the organization's information assets consisting of outsourced personnel who work outside the organization's security perimeters.

The remainder of this paper is organized as follows. Section II provides a description of the goodness of the security patterns and shows related works in order to represent these patterns. Section III presents a new description template of security patterns. Section IV introduces a case study with which to validate the description template and guarantee the security of external accesses to organizations' production environments. Section V presents the lessons that we have learned after carrying out the case study. Section VI shows our general conclusions with regard to the approach, and presents our future work.

II. SECURITY PATTERNS

A security pattern describes a recurrent security problem, which arises in a specific context, and provides a well tested generic scheme as a solution to that problem [23]. One of the main advantages of patterns is that they combine experience in the design of information system [8], thus making them more efficient. Patterns are a literary format with which to capture the knowledge and experience of security experts, resulting in a structured document in the form of a template to which the security experts' knowledge is transferred [22].

The first authors to propose security patterns were Yoder and Barcalow in 1997 [29]. The number of security patterns which have been published has increased considerably since then [12, 23, 30].

A great heterogeneity exists between the different descriptions in each of the security patterns published [2, 9, 11, 22]. This is because the authors who describe the security patterns that have been discovered have historically used different description templates to represent them. The most frequently used templates are those proposed by the Gang of Four [10], which have been adapted to describe security patterns, the template proposed by Buschmann et al. [4], the template proposed in the SERENITY project [24], and that proposed by Alexander [1]. Apart from these, other templates for the description of patterns have also been published, but their use is not yet massively widespread. One example of these is that proposed in [25], in which the security patterns are represented as calculation events. Recent years have seen proposals of other types of more specific security patterns, such as attack patterns [7] or misuse patterns [9].

Although the various authors who describe security patterns do not use a standardized description template, the majority of the description templates of these patterns have the following trio of elements in common: the context in which the pattern has been discovered; the security problem that it attempts to resolve within the context put forward; and the forces that affect the solution. The solution is conditioned by the associated forces, and these are expressed through UML diagrams which model this solution [9].

In order to resolve the lacks detected in current security patterns and to thus support information security engineers when analyzing and designing organizations' security architectures, we propose a new description template of security patterns. The template proposed below is intended to be an easy-to-use guideline, which will allow both experts and non-experts in security to access a structured and methodical document with which to resolve security problems in the real complex systems of the organizations in which they work.

III. A NEW DESCRIPTION TEMPLATE OF SECURITY PATTERNS

In this section, we shall set out the new description template of security pattern, explaining its characteristics and the contribution that it will make to the scientific community in the field of security. We shall then go on to enumerate and detail

each of the description elements of the proposed template.

A security pattern focused on the development of security architectures describes a valid generic path that assists security engineers to make analysis and design decisions when confronting the development of a secure architecture, which will resolve a real security deficiency in an information system. In order to obtain the maximum applicability within an organization, the proposed solution is oriented towards the architecture and technology that must be used in that organization in order to guarantee the security of the information assets associated with the deficiencies that they intend to resolve.

The new template will be specified with the description elements from the description template proposed by Buschmann et al. [4] and the template proposed in the SERENITY project, used in [5], together with the new description elements that are necessary to provide security experts and non-experts with a template to support the design of security architectures.

One of the principal contributions of this approach is that the proposed solution provides the security engineer with three complementary levels or *viewpoints:* the platform independent level, the platform specific level and the product dependant level. This solution model manages to separate the implementation of the system's functionality specification over a platform in a specific technology. This allows us to differentiate the functionality that the system must satisfy and the technologies that could be implemented to develop the solution. The security engineer can also visualize the evolution of the solution from abstract models to real implementations in the complete system.

Figure 1 shows a graphic representation of the solution levels.

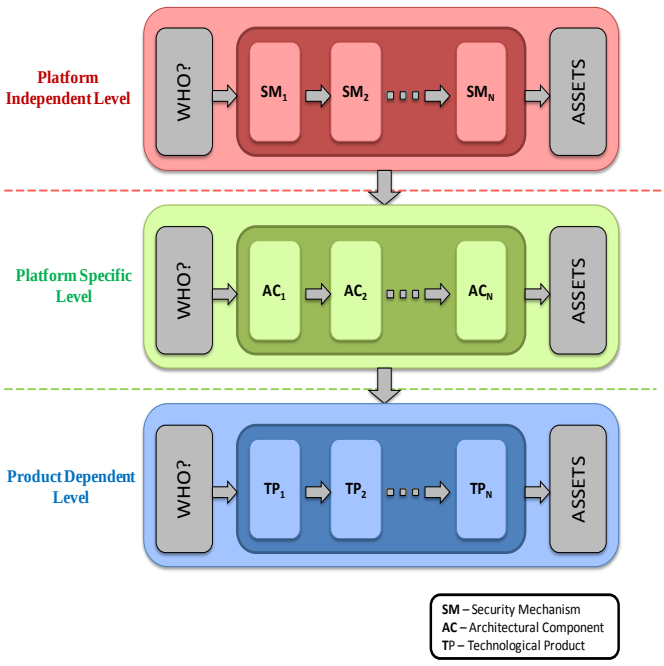


Figure 1. Abstraction levels of the solution.

As the figure above shows, all security systems must consider which information assets they intend to protect and who will have access to them.

We shall now provide a short description of each of the abstraction levels shown in Figure 1, and how the transformations needed to move from one level to the following should be carried out, illustrating which new elements should be incorporated or considered.

Platform Independent Level: this level provides a description of the security functionalities that the system should have, independently of its technological characteristics and implementation details. More specifically, a conceptual description of the security mechanisms that should be incorporated into the system is provided, along with the type of relationships that exist among them. The elements that should appear at this level are security patterns which are oriented towards the development of security mechanisms. A good guideline which can be used as a basis for discovering the type of patterns that are necessary is the guideline developed by Schumacher et al. in [23].

Platform Specific Level: the solution should be defined in this level, detailing the architecture or platform to which it will be applied. It is also necessary to set out how the necessary security mechanisms should be situated, through the presentation of an optimum security architecture with which to resolve the problem, independently of the technology used to protect the organization's systems. Given that security problems have repercussions on specific technological architectures, the same platform independent model can be instantiated N times, since it corresponds with different technological architectures. The security mechanisms described in the independent level become architectural components in this level.

Product Dependent Level: it is necessary to install the platform specific model in a specific architecture in this level, in order to implement it with technological products that are already available. Each of the architectural components can, therefore, be transformed into N technological products. The technological products must be valid products made by known manufacturers in the security industry. The final solution may vary significantly depending on the technologies used. This level should be independent of the information system's technological conditions. This view of the solution is very practical since it shows the user the different technologies that already exist on the market and that are oriented towards resolving the given problem.

This manner of structuring the solution provides a clear example of the steps that must be followed to implement the pattern, signifying that both experts and non-experts can understand the solution and know how to deploy it in a real system.

A further implicit property of this description template is its associated *decision path*. This element is of great assistance when selecting the most appropriate pattern with which to resolve a determined problem. The following five levels have

been proposed in the decision path in order to classify the patterns that are associated with a discovered security deficiency:

1) What is the state of the information, programs or configurations that need to be protected? The possible states are the following:

a) Stored: These are found in a data base.

b) Transit: Through a transfer to another company or service. There is a movement of information.

c) Accessed: The information is being accessed.

2) Who accesses the information that we wish to protect? The people who can access the information are:

a) The organization's internal users.

b) External users or customers.

c) Computing personnel during their work. This type of user is special since s/he can access data, applications and systems without using the security mechanisms which have been designed in the applications utilized by the final users.

3) How is the information accessed? or *What is the means of access?* In short, the information can be accessed in the following manners:

a) Directly: By accessing the data directly without any limitations to the use that is made of them.

b) Through an application: By applying business logic to the use, through which the information is shown.

4) Where is the information accessed from? It is basically accessed from two places:

a) From within the organization, i.e., all the technological spheres that are governed by the same security policies.

b) Outside the organization: where it is not possible to ensure the fulfillment of the same security policies that appear in the organization in which the assets are located.

5) Who manages the means used to access the information that needs to be protected?

a) The person responsible for security, who will use the pattern and will be legally authorized to manage the systems' security.

b) Any other person who does not belong to the organization or does not have legal authorization to manage the system's security.

This *decision path* can be used to verify what type of problem, in general terms, will be resolved with the pattern discovered, i.e., two security patterns that respond identically to the same path resolve problems of the same nature, and could thus be alternatives to the same problem.

With regard to the elements described in the template, it is also necessary to emphasize that they do not describe the security vulnerabilities that may affect the information system in which the solution is installed. This is owing to the fact that new vulnerabilities frequently appear and the pattern must constantly be modified. We consider that it is the technologies themselves that should be updated each time a new vulnerability is encountered, and that in this case it should be the manufacturer who updates them, or the security administrator who incorporates new rules into the security technologies used, if the impact of these vulnerabilities is to be minimized. This new template of security patterns therefore considers that vulnerabilities appear in all technologies on a permanent basis, and this concept forms a part of the pattern's considerations. The greater a technology's exposure to public networks, the higher its level of weakness. All security architectures will therefore be designed by bearing in mind that critical vulnerabilities repeatedly appear in all technologies.

The template proposed for the description of security patterns focused on the design of security architectures will be shown as follows. We must emphasize that this template is used to evolve existing security patterns, since it maintains the same base structure as their description, and it is only necessary to add the new elements that are proposed. The template that is proposed consists of the following elements:

A. Name

The pattern's name should represent the problem that it is attempting to resolve. This name must also be unique within the sphere of this type of patterns.

B. Context

The context provides a generic description of the setting, at both user level and system level, and includes the conditions in which the described pattern should be applied.

C. Problem

This describes the situation which has led to the necessity to apply a series of security mechanisms in order to obtain an optimum solution, and basically describes the reasons for the problem. It should also indicate the following questions:

- Which assets need to be protected? Information, programs and/or configurations.
- What are we protecting ourselves from? Information leaks, massive attacks, etc.
- Which security properties do we intend to conserve? Confidentiality, integrity, availability, auditability and/or non-repudiation.

D. Known incidents

A description of real cases of known security incidents, in relation to the problem posed that the implementation of the pattern intends to resolve. These incidents can be easily located on the Internet on specialized sites [6], which collect this type of events and specify when they occurred, how they occurred and what their impact was.

E. Decision Path

This element should describe all the general levels of the state of the assets that need to be protected (described previously). This will make it possible to determine which pattern should be used to resolve a specific security problem. The objective of this descriptive element is to be able to develop a methodology based on security patterns, on the basis that the definition of the pattern in itself develops its own path in the decision tree.

F. Solution

This element describes the solution in accordance with the scenario and the problem being considered. This solution must be expressed in three different abstraction levels, as has been shown previously. It is first necessary to set out the solution for a platform independent level, showing the security mechanisms that must be used and the relationships that exist among them. This first level is then transformed into a second level, called a platform specific level, which refers to the technological architecture proposed to resolve the given problem. The second level is finally transformed into a third level, called the product dependant level, which shows a proposal for the technologies that can be used to implement the solution proposed by the pattern described. The Security Engineering sector must consider these technologies to be trustworthy.

G. Considerations

It is necessary to carry out a qualitative analysis of the solution in relation to the critical parameters found in the real complex system: a) storage; b) memory consumed; c) frequency with which the systems, technologies and applications are patched up; d) process capacity; e) complexity for final user; f) complexity for security/system administrator; g) complexity of log management; h) broadband consumed; i) complexity for massive use of the solution; j) cost of installing solution; and k) solution fulfillment guarantees. It is necessary to decide whether each of these aspects is qualitatively altered in a Null (0), Low (1), Medium (2) or High (3) manner when deploying the solution in a real information system.

These decisions will assist when evaluating whether or not the implantation of the solution is appropriate with regard to the organization's current situation. This is particularly true when considering the cost parameters and fulfillment conditions, since excessive costs and an inability to ensure the fulfillment of the solution might be the principal cause of a solution being rejected.

H. Rules and Regulations

If the adoption of a predefined solution in the form of a pattern in a real environment is desired, it is necessary to consider the regulations of the country in which the solution is intended to be installed, with regard to the information activities that need to be protected. We must also bear in mind the rules associated with these regulations which must be fulfilled in the proposed solution in order for them to be correct from both a juridical and legal standpoint. For example, Argentina does not permit the movement of information related to people who reside in that country, and a solution which does not fulfill this regulation could not, therefore, be installed.

I. Benefits

A short description of a solution's goodness with regard to the sphere and specific context in which the pattern is developed.

J. Consequences

This element describes the consequences of adopting a pattern as a solution in a real information system. An analysis of the risks that the organization will run if it does not adopt this solution must also be carried out. To do this, it is necessary to describe the following consequences:

- Negative consequences of tackling the solution.
- Consequences of not tackling the solution.

K. Alternatives

The majority of security deficiencies can be resolved in different ways, and this section should therefore describe other solutions that can be used to resolve the problem considered. These alternatives may differ from the pattern described in the technological level, in the architectural level or even in the security mechanisms used to guarantee the information assets that are at risk.

IV. A CASE STUDY

In this section, we present a summary of a case study that was carried out in an organization in the banking sector with the objective of validating the template proposed in the previous section. To do this, we have followed all the elements included in the pattern template. The other objective of this case study is to help us to resolve a security lack related to the accesses of personnel who work outside the organization's security perimeter in production environments, such as external personnel dedicated to software maintenance.

Within the sphere of Information Technologies, an organization's production area is of maximum criticality owing to the fact that it is here where the information and data directly used by the customers and end users is kept. The extraction of this information or the malicious modification of the programs that access it may cause great losses in the organization. We have therefore carried out research to discover a security pattern that will resolve this lack, such that any security engineer who confronts this problem will be able to use this solution as a basis to guarantee the security of the information assets of the organization to which s/he belongs. The elements included in this security pattern are described as follows:

A. Name

Security Pattern for External Access to Productive Environments.

B. Context

The evolution of technology, and in order to reduce costs in infrastructures and installations, both on the part of an organization and on that of the suppliers in charge of maintaining the applications, signify that it is possible to locate the work carried out by these maintenance companies outside organizations' internal networks. This work is, in some cases, currently developed in the organization's installations with the objective of locating it in infrastructures belonging to the suppliers so that the maintenance work is carried out outside

the organization's installations. A supplier's design and development personnel who work for the organization therefore need to access the production environments from outside the internal security network.

C. Problem

The people who access the production information are situated in locations which are not controlled by the same person, from the point of view of security, i.e., they are not under the same security management as the systems in which the asset to be protected is located. It is also necessary to be able to download the information onto the maintenance personnel's computers to allow them to deal with it and reestablish normality in the system, signifying that these people can download information and remove it from the organization's internal network. It is therefore vital to provide any accesses that occur with security as regards the organization's assets in order to avoid undesirable situations such as misuse, illegitimate copying or any manipulation of these assets that may affect output and the organization's image.

- *Which assets need to be protected?* The principal asset to be protected is the organization's information to which the company personnel outside the organization's security limits have access.
- *What are we protecting ourselves from?* Asset leaks, i.e., leaks of the information to which the company's personnel have access.
- *Which security properties do we intend to conserve?* Confidentiality and auditability.

D. Known Incidents

Two known incidents of the theft of information or money from large organizations will be shown as follows. These thefts took place in companies that were carrying out an external service for other organizations which were the real victims of the theft.

The first incident is related to the theft of a large amount of money ($2 million) from Citibank [20]. Russian hackers accessed critical customer information from Citibank via an SQL injection vulnerability that they found on the Website of the American chain store 7-Eleven. At the time of the information theft there were 5,500 Citibank-branded ATMs at 7-Eleven. This means that for two weeks in September 2007, anyone who entered their PIN number in one of these ATMs was exposed to this fraud. As soon as the hackers had obtained duplicate bank cards and their associated PIN numbers, they began to withdraw money and to pay by stolen credit card.

The second incident is related to the theft of information from Epsilon [14], the World's Largest Permission Based Email Marketing Services Company. Epsilon sends over 40 billion emails annually and has over 2,500 clients, including 7 of the Fortune 10 to build and host their customer databases. Security Week has been able to confirm that the customer names and email addresses, and in a few cases other pieces of information, were compromised at several major companies, including the following: Kroger, TiVo, US Bank, JPMorgan Chase, Capital One, Citibank, Ameriprise Financial, Lacoste, Hilton Honors Program and Marks & Spencer, among many others.

This type of harvested data can be categorized as a minor threat, but having access to customer lists opens the opportunity for targeted phishing attacks against customers who expect communications from these companies. Attackers can use this type of data to send a targeted phishing message to a bank customer and personally address them by name. This type of attack will certainly result in a much higher success rate than a typical spamming campaign. Having access to this information will therefore simply help phishing attacks to achieve a higher success rate.

E. Decision Path

The following questions will assist in the classification of this pattern in the general context of solutions that can be found within this type of security patterns.

1) What is the state of the information, programs or configurations that needs to be protected? The state of assets is *accessed.*

2) Who accesses the information that we wish to protect? The people who access the information that we wish to protect are computing personnel during their work.

3) How is the information accessed? or *What is the means of access?* The information is accessed *directly*.

4) Where is the information accessed from? The maintenance team will access it directly from outside the organization's security perimeter.

5) Who manages the means used to access the information that needs to be protected? The security of the installations in which the maintenance work on the applications is carried out is not under the security management of the organization that requires this work.

F. Solution

The solution to the problem proposed will be set out at different abstraction levels, from the platform independent level to the product dependent level. We shall first show the platform independent level model, and shall then go on to transform this level in order to develop a model that is specific to the platform. Finally, we shall transform the previous level in order to show the solution from a product independent level.

Platform Independent Level: As is shown in Figure 2, the people in charge of the organization's software maintenance gain access via the organization's Internet in order to modify defective data and/or programs.

The security mechanisms that must be implanted in the organization's internal network to develop the desired solution are extracted from the following security patterns, which are described in greater detail in [23].

Each of the security mechanisms used to develop the solution in the platform independent level is described as follows:

- *Identification & Authentication:* Security patterns such as "I&A Requirements", "Automated I&A Design Alternatives" and "Password Design and Use" can be

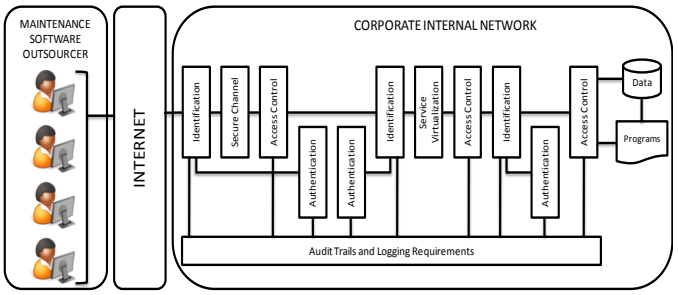


Figure 2. Platform Independent Level

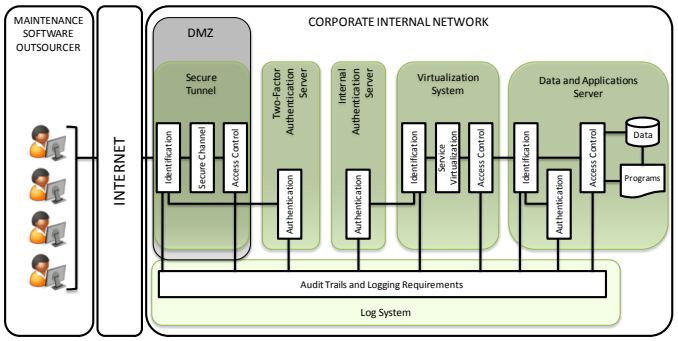


Figure 3. Platform Specific Level

used and combined for this solution. These security patterns can be used when an actor (person, process or other entity) intends to interact with an organization's system, and it must recognize that the actor is interacting with the system. Although these mechanisms appear separately in the previous figure, they are in fact complementary systems, since the whole system identification must validate the credentials via an authentication system.

- *Secure Channel:* This security mechanism is incorporated to avoid the situation of an attacker intercepting messages exchanged between the maintenance personnel and the organization on the Internet. This mechanism is able to code the channel that the information, which is in many cases sensitive, travels along.
- *Access Control:* A multitude of security patterns that expose various access control mechanisms currently exist. Those such as: "Authorization", "RBAC", "Multilevel Security" and "Role Rights Definition" can be used and combined, if considered necessary, in this solution. This type of security patterns defines security restrictions, i.e., they define the rules that permit access to the zones, resources and other aspects that strengthen the organization's security.
- *Audit Trails and Logging Requirements:* This security mechanism allows the registers that are carried out with regard to events and activities to be captured and audited, such as the identification and authorization of a resource in the organization.
- *Service Virtualization:* This security mechanism is in charge of creating a virtual version of a device or resource: a server, a storage device, a network, an operative system etc. This mechanism also permits the handling, management and provision of a computer's four main resources (CPU, memory, network and storage), thus allowing the dynamic sharing of these resources among the virtual machines defined in the central computer.

Platform Specific Level: As has already been explained in the previous section, this level is a transformation from the platform independent level. This transformation includes the architectonic components that are necessary to provide the solution. As Figure 3 shows, each of the architectonic components is composed of one or more of the security mechanisms detailed in the previous level.

Each of the architectonic components used to develop the solution in the platform specific level is described as follows:

- *Demilitarized zone (DMZ):* The intention of this security zone is to isolate the organization from potential attackers by separating the access to the various applications and services (the organization's public zone) from the different servers (the organization's private zone).
- *Secure Tunnel:* This is situated in the organization's internal network DMZ and is in charge of establishing the first filter between the Internet and the users who attempt to access it via the Internet. This security measure establishes a secure communication tunnel via the Internet in order to ensure that access to the organization's systems is exclusive to the personnel who are registered in the organization's two-factor authentication server. This tunnel consists of the following security mechanisms: (i) an identifier to allow the users to introduce their credentials. These credentials will be passed to the two-factor authentication server to verify whether they are correct. If the user has access permission then (ii) a secure tunnel (SSL) will be established from the software provider's installations which are located outside the organization's perimeter and installations. In parallel to this, (iii) the access control will manage the permissions concerning the user's resources in order to ensure that the user can only access the resources facilitated.
- *Two-Factor Authentication Server:* The external credentials of the users who wish to access the systems of the organization's internal network are checked in this server, which controls the login, the password and a two-factor to make the user access mechanism more robust.
- *Internal Authentication Server:* This authentication server will check the internal credentials which can be used to access the organization's systems. In this case they will be different credentials to those requested by the two-factor authentication server.
- *Virtualization System:* This system is in charge of creating a virtual version of the organization's systems. It must also contain (i) an identifier which is in charge of being the interface into which the users introduce their credentials. Once the credentials have been checked against the internal authentication server, (ii) a virtualized version of the system is

created which (iii) maintains an access control to establish the permissions needed with regard to the resources available.

- *Data and Applications Server:* This system is in charge of storing both the organization's data and the applications. In order to make the access to this system, which contains the resources, more robust it is necessary to install a series of security mechanisms which carry out the tasks of: (i) identification, (ii) authentication, and (iii) access control, as in the other previously explained areas.
- *Log System:* This architectonic element is in charge of collecting all the activities that are relevant to identification, control and access control of the various mechanisms. The information is gathered in the form of a log.

Product Dependent Level: As is shown in Figure 4, in this level the architectonic elements detailed in the previous level are transformed into specific technological products of concrete manufacturers. The technological products to be installed in the organization's internal network must be products which have been validated in the company's security environment.

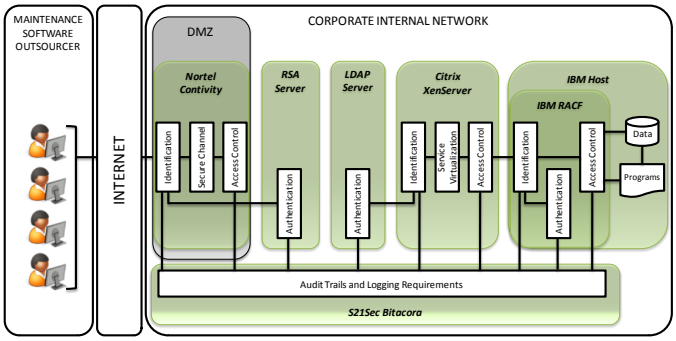


Figure 4. Product Dependent Level

As is shown in the previous figure, the specific technological products correspond with one or more of the architectonic components defined in the platform independent level. Table 1 shows the correspondence between the architectonic components and the specific technological products of this solution:

TABLE 1. TECHNOLOGICAL PRODUCTS

Architectonic Components	Technological Products
Security Tunnel	Nortel Contivity
Two- Factor Authentication Server	RSA Server
Internal Authentication Server	LDAP Server
Virtualization System	Citrix XenServer
Data and Applications Server	IBM Host
	IBM RACF
Log System	S21Sec Bitacora

G. Considerations

It must be possible to answer whether each of the technological aspects related to the system in which the solution is integrated are qualitatively altered in any of the following manners: null (0), low (1), medium (2), or high (3). The first column in Table 2 shows the considerations that must be taken into account, while the second column provides a brief description of the means used to analyze each consideration. Finally, the third column shows the results of the qualitative analysis of this solution.

TABLE 2. CONSIDERATIONS

Aspects to consider	Description	Analysis
Storage	It is necessary to identify specific causes resulting from an elevated consumption of storage in comparison to other existing solutions. It is principally necessary to estimate the economic impact that the adoption of this solution will have on the organization.	1
Memory Consumed	It is necessary to identify specific causes resulting from high memory consumption in comparison with other existing solutions.	3
Frequency of Patching	It is necessary to estimate specific causes resulting from a rise in the frequency of patching up the solution proposed. This evaluation must be carried out in both the economic plan and the risk plan associated with vulnerabilities.	0
Process Consumption	It is necessary to identify specific causes resulting from the elevated consumption of the process capacity in comparison with other existing solutions.	3
Broadband	It is necessary to estimate the technological aspects needed to identify specific causes resulting from a high consumption of broadband in comparison to other existing solutions.	2
Installation Cost	It is necessary to evaluate the global cost of the installation of the solution in the organization's real environment in comparison with other existing solutions.	3

Complexity for Security Administrator	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the person in charge of adapting it and maintaining in the organization's installations.	0
Complexity of Log Management	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the personnel who manage and research the logs collected.	2
Complexity of Use for End User	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the end user when using the systems in which the solution has been applied.	0
Complexity of Massive Expansion	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by those in charge of the massive installation of the solution in various points of the organization.	0
Complexity for System Administrator	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the administrators of the other systems, and the impact on their daily work.	0
Residual Risk	It is necessary to evaluate whether the pattern, once it has been installed and is functioning correctly, needs complementary measures to attain its initial objective.	0
Capacity to Ensure Fulfillment	It is necessary to evaluate whether it is possible to display the necessary measures which allow us to verify the correct functioning of the pattern, and whether all the participants will be sufficiently motivated to fulfill it.	3

H. Rules and Regulations:

In order to adopt the proposed pattern it is necessary to bear certain considerations in mind with regard to the location of the organization, the location of the software maintenance factories and the flow of information between the organization and the factories. It is therefore necessary to bear the following considerations in mind:

- Privacy laws are different in the different countries involved in adopting the solution.
- Make an inventory of the data that will be part of the normal flow of the solution in order to adopt the necessary measures relative to each country's treatment of personal data.
- The local restrictions of each country with regard to the treatment, quality and robustness of the passwords used, and the minimum security measures of the organizations residing in each country.
- Consult those responsible for security in the various headquarters of the organization so that they can evaluate the risks involved in not complying with the laws associated with the country in which they are.

I. Benefits:

If the pattern shown in this paper is adopted to provide security in productive environments outside the perimeter of the organization, then the following benefits will be obtained:

- The simplification of all externalization processes, in which externalization signifies the location of personnel outside the organization's perimeter, since the technological complexity associated with these processes is reduced.
- The architecture is reusable in similar situations such as the remote administration of systems, remote access to different environments to carry out tele-maintenance tasks, etc.
- The structural solution for an organization, since it is a robust solution that will last.
- Great savings in the organization's infrastructures, since the maintenance work on the applications is now carried out in the suppliers' installations rather than in those of the organization.

It can be deduced that this system is valid and has good behavior whenever it is necessary to make the security conditions of the system used independent in order to carry out different maintenance tasks on the systems in which these tasks must take place.

J. Consequences

The negative consequences of adopting this pattern as a solution, and the risks that the organization may run if it does not adopt this solution are the following:

- ***Negative consequences of tackling the solution:*** The adoption of this solution requires a great investment in the virtualization system, since the entire process moves from being carried out on the organization's computers to being carried out in a virtualized system. A high economic investment is also necessary to resize the infrastructures of the organization's systems in order to adapt them to the technological aspects needed to tackle the solution.
- ***Consequences of not tackling the solution:*** Organizations will run the risk of suffering situations of misuse, copying, illicit distribution and theft of the data to which the maintenance factory's personnel has access, thus making an impact on both the organization's image and exposing it to potential attackers.

K. Alternatives:

The different security alternatives with which to solve this aforementioned problem are as follows:

- ***Alternative 1:*** The first security alternative to the situation of the externalization of maintenance tasks is to centralize the security management in the same circumstances, conditions and restrictions as those used in the organization. This occurs by passing the security control of the factory's installations over to those responsible for security in the organization that requires the maintenance work to be done. It would thus be possible to ensure that the tasks carried out would be controlled exactly as the organization that had contracted the service wished. The fulfillment of this alternative is relatively low, since each supplier has its own security department and would find it difficult to adopt this measure.
- ***Alternative 2:*** Commitment on the part of the company supplying the service to adopt the security measures required by the organization that contracts them. These security measures consist of dedicating installations exclusively to the realization of maintenance work, the dedication of lines of communication, the adoption of the same security measures as the organization with regard to personnel, etc. This measure requires the signing of contracts containing all the aspects mentioned, confidentiality clauses, the periodical auditing of the company providing the service, etc.

V. LESSONS LEARNED

Our real-life experience of tackling an externalization project for the software suppliers in an organization in the banking sector is summarized as follows. We show the advantages, disadvantages and the lessons learned.

Before formalizing a solution using the previously explained pattern template, we carried out a security analysis, which is our usual course of action whenever we are confronted with a new project that affects the systems of the organization in which we work. The principal tasks carried out in this analysis can be seen in [18].

The initial requirement is clear: we must reduce costs in the organization, and one of the main reductions is achieved by externalizing the personnel that carry out the software development and maintenance tasks. To do this, it was necessary to design a solution so that the work of these groups of people was not affected, as far as possible, and so that they would continue working outside the organization's installations in as similar a way as possible.

One of the organization's main handicaps was that the information that these development factories accessed, when they were still at the organization's headquarters, was sensitive, and could have affected the business if it had leaked from the installations. In this new work model, they therefore had to access this information in a regular manner in order to continue functioning as normal.

After designing the solution, we decided to express it as a security pattern, since this type of situations is very common and recurrent in organizations, either for cost reduction or because the software developers cannot constantly displace themselves every time an incident occurs in the systems of the organization that they serve.

After analyzing and implementing the solution presented in the case study shown in the previous section, the principal advantages were the following:

In the first place, we attained the objective pursued, i.e., the reduction of costs. This objective was attained because in addition to obtaining a more accessible workforce, we also reduced investment in the organization's installations, such as electricity, gas, jobs, computational material, etc. With regard to what affects the information systems, and particularly the security environment of the information, one of the most notable advantages which had not initially been contemplated was that we obtained order and coherence in the access to productive environments on the part of the software development or maintenance teams when they were consulting, modifying or eliminating information that was, in some cases, critical. After deciding to externalize the software factories, the analysis concentrated on protecting the assets that would be accessed from outside the organization. This was done by designing the communications between the headquarters of the software factories and the organization's information systems. This allowed us to, on the one hand eliminate all the access routes that had previously existed in the organization and, on the other, to correctly censor all the accesses to the productive environments, analyzing each of the casuistries that the software developers requested to carry out their maintenance tasks. We thus managed to eliminate undesired accesses that might have been occurring. This resulted in a) a greater control of the actions carried out by the externalized software maintenance teams, without organization's information systems, b) a refinement of the entity's technological systems, c) a considerable increase in the security of the information systems that contained data that was critical to the organization, and d) we obtained an exhaustive census of all the accesses that were produced, from the software factories to the information systems, so that if necessary we could carry out a forensic analysis of hypothetical information leaks or violations of the service conditions on the part of the company providing the service.

On the other hand, the disadvantages discovered concerned incidents provoked, because in some cases, when the personnel in charge of the maintenance or installation of the software developed arrived at their new location they could not access the information systems with the same privileges as they had had when they were in the organization's systems, and this limited their actions when working at remote form. These work teams have therefore had to adjust their customs to the new form of work designed to carry out their function. One example of this type of cases is the following: given that the system which is prepared to access the production environment

is virtualized and watertight, i.e., no information can be moved from that system to others, one of the problems was that it was impossible to print out or download any type of document, because if this were permitted it might lead to uncontrolled information leaks. This was an habitual practice for the software maintenance teams, since when cataloging and resolving incidents they download the information to their computers to then substitute the modified code without having to always be connected to the organization's network. This situation obliged each of the developers who resolved and cataloged incidents in the organization's productive environments to always have an Internet connection at their disposal.

Leaving aside the advantages and disadvantages discovered after implementing the previously proposed solution, we consider that the structure of this solution in the form of a pattern will help to optimize the time effort and cost needed to analyze this type of problems, because it reduces the majority of similar cases to one specific case, which is the access to critical data from outside the installations of the organizations that own them.

With regard to the pattern template proposed, the analysis of this type of real cases has provided us with a huge amount of feedback with which to refine the proposal. We therefore consider the section in which the information assets that must be protected are cataloged to be primordial. This is owing to the fact that if there is an exhaustive cataloging of the assets that are accessed, in addition to obtaining the locations from which they can be accessed, the security measures to be applied are very specific, i.e., depending on the criticality of the data to be protected and from where they are accessed, the mechanisms used to guarantee the security of the information assets can either be very relaxed or very robust.

VI. Conclusion and Future Work

In this paper, we have presented a security pattern description template which had previously been published, and which has now been completed with a practical case, which validates its use in a specific problem in a real and complex organization.

After carrying out a systematic review of the state-of-the art of those works that present security patterns, we detected a series of lacks, which we have attempted to solve with this new template. Of the most outstanding lacks we can highlight: the proposals analyzed are oriented towards the construction of security mechanisms, and not towards the construction of security architectures in information systems; they do not contemplate the impact that the implementation of the security pattern will make on an information system; they do not carry out a classification on the basis of the criticality of the information assets to be protected, and do not therefore specify the appropriate security measures to be applied; and they do not take into consideration the regulations or laws that may apply to the solution in the sector or country in which the organization operates.

All these lacks have motivated our research, which has led us to design a template for the description of security patterns oriented towards the construction of secure architectures, which collects each of the lacks detected in the aforementioned works.

In this work, we have not only presented the new template with which to describe security patterns, but we have also shown a case study extracted from a real and recurrent problem faced by technological organizations, which is access to productive environments from outside the organization's security perimeter. With this proposal we intend to validate the previously defined template, with a real case in a complex organization.

This security problem is very common in any large organization and requires an exhaustive analysis to be mitigate as much as possible the leaking of the organization's information assets. We have therefore decided to describe this problem with a solution in the form of a pattern, which will thus serve to assist information security engineers to resolve problems of this type in an agile and effective manner, whilst maintaining the homogeneity in each of the systems in which it is implemented.

This exercise of adapting a real problem to the form of the security pattern template proposed has helped us to validate and refine the template on which we are working. For example, we have realized the importance of the template section in which the security assets to be protected on the basis of their criticality are classified. This is therefore one of the fundamental parts to which most attention is paid when designing security architectures in the form of a security pattern.

We are currently working on the implementation of new practical cases following the template, which will allow us to refine and validate it. We are also working on the modeling of a framework based on security pattern mining, whose principal objective is to discover, design and document security patterns that concentrate on supporting the design of security architectures.

Another of the lines on which we are working is the definition of a secure information system development methodology based on security patterns, which will guide information security engineers when systematically and homogonously resolving security problems in real complex organizations.

Acknowledgements

This research has been carried out in the framework of the following projects: MODEL-CAOS (TIN2008-03582/TIN) financed by the Spanish Ministry of Education and Science, BUSINESS (PET2008-0136) financed by the Ministry of Science and Innovation, and SISTEMAS (PII2I09-0150-3135) and SERENIDAD (PEII11-0327-7035), all financed by the Local Government of Castilla-La Mancha, in Spain.

REFERENCES

[1] C. Alexander, S. Ishikawa, and M. Silverstein "A Pattern Language: Towns, Buildings, Constructions" Oxford University Press, 1977.

[2] Z. Anwar, W. Yurcik, R. E. Johnson, M. Hafiz, and R. H. Campbell "Multiple design patterns for voice over IP (VoIP) security" in Performance, Computing, and Communications Conference (IPCCC 2006). 25th IEEE International, 2006.

[3] A. Barth, C. Jackson, and C. Reis "The Security Architecture of the Chromium Browser", Technical Report 2008.

[4] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. "Pattern-oriented software architecture: A system of patterns" Wiley, 1996.

[5] A. Cuevas, P. El Khoury, L. Gomez, and A. Laube "Security Patterns for Capturing Encryption-Based Access Control to Sensor Data" in SECURWARE '08. Second International Conference on Emerging Security Information, Systems and Technologies, 2008, pp. 62-67.

[6] "DATALOSS db - Open Security Fundation", http://datalossdb.org/, retrieved: January, 2012.

[7] E. Fernandez, J. Pelaez, and M. Larrondo-Petrie "Attack Patterns: A New Forensic and Design Tool" in Advances in Digital Forensics III, 2007, pp. 345-357.

[8] E. B. Fernández "Security patterns and secure systems design" ACM Southeast Regional Conference 2007.

[9] E. B. Fernandez, N. Yoshioka, and H. Washizaki "Modeling Misuse Patterns" in ARES '09. International Conference on Availability, Reliability and Security, 2009, pp. 566-571.

[10] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides "Design Patterns: Elements of Reusable Object Oriented Software" Addison Wesley, 1995.

[11] J. Garzás and M. Piattini "Object Oriented Microarchitectural Design Knowledge" IEEE Software, pp. 28-33, 2005.

[12] M. Hafiz, P. Adamczyk, and R. E. Johnson "Organizing Security Patterns" Software, IEEE, pp. 52-60, 2007.

[13] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt "Security patterns repository, version 1.0" 2006.

[14] M. Lennon, "Massive Breach at Epsilon Compromises Customer Lists of Major Brands", http://www.securityweek.com/massive-breach-epsilon-compromises-customer-lists-major-brands, retrieved: January, 2012.

[15] S. Moral-García, S. Moral-Rubio, and E. Fernández-Medina "Security Pattern Mining: Systematic Review and Proposal" in WOSIS '11. 8th International Workshop on Security in Information Systems, 2011, pp. 13-24.

[16] S. Moral-García, R. Ortiz, S. Moral-Rubio, B. Vela, J. Garzás, and E. Fernández-Medina "A new Pattern Template to Support the Design of Security Architectures" in PATTERNS 2010. 2nd International Conference on Pervasive Patterns and Applications, 2010, pp. 66-71.

[17] T. Okubo and H. Tanaka "Web security patterns for analysis and design" in Proceedings of the 15th Conference on Pattern Languages of Programs, Nashville, Tennessee, 2008.

[18] R. Ortiz, S. Moral-Rubio, J. Garzás, and E. Fernández-Medina "Towards a Pattern-Based Security Methodology to Build Secure Information Systems" in WOSIS '11. 8th International Workshop on Security in Information Systems 2011, pp. 59-69.

[19] OSA, "Open Security Architecture", http://www.opensecurityarchitecture.org/cms/index.php, retrieved: January, 2012.

[20] K. Poulsen, "7-Eleven Hack From Russia Led to ATM Looting in New York", http://www.wired.com/threatlevel/2009/12/seven-eleven/, retrieved: January, 2012.

[21] D. G. Rosado, C. Gutiérrez, E. Fernández-Medina, and M. Piattini "Security patterns and requirements for internet-based applications" Internet Research: Electronic Networking Applications and Policy, pp. 519-536, 2006.

[22] M. Schumacher "B. Example Security Patterns and Annotations" in Security Engineering with Patterns, 2003, pp. 171-178.

[23] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad "Security Patterns: Integrating Security and Systems Engineering" Wiley, 2006.

[24] "Serenity Project - System Engineering for Security & Dependability", www.serenity-project.org, retrieved: January, 2012.

[25] G. Spanoudakis, C. Kloukinas, and K. Androutsopoulos "Towards security monitoring patterns" in Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, 2007.

[26] W. Stallings "Network security essentials: applications and standards", Prentice Hall, 2007.

[27] C. Steel, R. Nagappan, and R. Lai "Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management", Prentice Hall ed., 2005.

[28] H. Washizaki, E. B. Fernandez, K. Maruyama, A. Kubo, and N. Yoshioka "Improving the Classification of Security Patterns" in DEXA '09. 20th International Workshop on Database and Expert Systems Application, 2009, pp. 165-170.

[29] J. Yoder and J. Barcalow "Architectural Patterns for Enabling Application Security" in Fourth Conference on Patterns Languages of Programs (PLoP'97), 1997.

[30] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen " An inventory of security patterns" Katholieke Universiteit Leuven, Department of Computer Science, 2006.

iPrivacy: A Distributed Approach to Privacy on the Cloud

Ernesto Damiani, Francesco Pagano
Department of Information Technology
Università degli Studi di Milano
Milano, Italy
{ernesto.damiani, francesco.pagano}@unimi.it

Davide Pagano
School of Engineering
Politecnico di Milano
Milano, Italy
davide1.pagano@mail.polimi.it

***Abstract*—The increasing adoption of Cloud storage poses a number of privacy issues. Users wish to preserve full control over their sensitive data and cannot accept that it is accessible by the remote storage provider. Previous research was made on techniques to protect data stored on untrusted servers; however we argue that the cloud architecture presents a number of open issues. To handle them, we present an approach where confidential data is stored in a highly distributed database, partly located on the cloud and partly on the clients. Data is shared in a secure manner using a simple grant-and-revoke permission of shared data and we have developed a system test implementation, using an in-memory Relational Data Base Management System with row-level data encryption for fine-grained data access control.**

Keywords-cloud; database; encryption; data sharing; privacy; distributed data.

I. INTRODUCTION

Cloud computing is the commercial evolution of grid computing [23]; it provides users with readily available, pay-as-you-go computing and storage power, allowing them to dynamically adapt their IT (Information Technology) costs to their needs. In this fashion, users need neither costly competence in IT system management nor huge investments in the start-up phase in preparation for future growth.

While the cloud computing concept is drawing much interest, several obstacles remain to its widespread adoption, including:

- Current limits of ICT infrastructure: availability, reliability and quality of service;
- Different paradigm of development of cloud applications with respect to those used for desktop applications;
- Privacy risks for confidential information residing in the cloud.

Hopefully, the first obstacle will diminish over time, thanks to increasing network availability; the second will progressively disappear by training new developers; the third issue however, is still far from being solved and may impair very seriously the real prospects of cloud computing.

In this paper, we illustrate some techniques for providing data protection and confidentiality in outsourced databases (Section II), analyze some possible pitfalls of these techniques in Cloud Computing (Section III), and propose a new solution based on distributed systems (Section IV), experimentally implemented and benchmarked (Section V).

I. THE PROBLEM OF PRIVACY

The cloud infrastructure can be accessible to public users (Public Cloud) or only to those operating within an organization (Private Cloud) [3]. Generally speaking, external access to shared data held by the cloud goes through the usual authentication, authorization, and communication phases. The access control problem is well acknowledged in the database literature and available solutions guarantee a high degree of assurance.

However, the requirement that the maintainer of the datastore cannot access or alter outsourced data is not easily met, especially on public clouds like Google App Engine for Business, Microsoft Azure Platform or Amazon EC2 platform.

Indeed, existing techniques for managing the outsourcing of data on untrusted database servers [13] [14] cannot be straightforwardly applied to public clouds, due to several reasons:

- The physical structure of the cloud is, by definition, undetectable from the outside; who is really holding the data stored on the cloud?
- The user often has no control over data replication; i.e., how many copies exist (including backups) and how are they managed?
- The lack of information on the geographical location of data (or its variation over time) may lead to jurisdiction conflicts when different national laws apply.

In the next section, we will briefly summarize available techniques for data protection on untrusted servers, and show how their relation to the problems outlined above.

A. Data Protection

To ensure data protection in outsourcing, the literature reports three major techniques [6]:

- Data encryption [15];
- Data fragmentation and encryption [16], which in turn can be classified into two major techniques:
  - non-communicating servers [17][18];
  - unlinkable fragments [19];
- Data fragmentation with owner involvement [20].

1) Data encryption

To prevent unauthorized access by the Datastore Manager (DM) of the outsourced Relational Data Base Management Systems (RDBMS), data is stored in encrypted form. Obviously, the DM does not know the encryption keys, which are stored apart from the data. The RDBMS

receives an encrypted database and works on bit-streams that only the clients, who hold the decryption keys, can interpret correctly.

Figure 1 shows the transformation of a plain text tuple into an encrypted one.

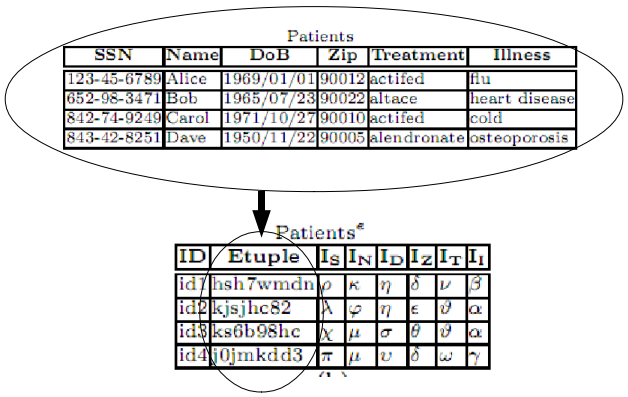


Figure 1. Data encryption, source: [6]

Decryption keys are generated and distributed to trusted clients by the data owner or by a trusted delegate.

Encryption can be performed at different levels of granularity: field, record, table, db [28]. Usually, the level adopted is the record (i.e., a tuple in relational databases).

It is important to remark that since data is encrypted, the DBMS cannot index it based on plaintext and therefore it cannot resolve all queries. Available proposals solve this problem by providing, for each encrypted field to be indexed, an additional indexable field, obtained by applying a non-injective transformation *f* to plaintext values (e.g., a hash of the field's content). Using this method, equality queries can be performed easily, although with a precision index < 1 (to prevent statistical data mining). The trusted client, after receiving the encrypted result set for the query, will decrypt it and exclude spurious tuples. However range queries are difficult to compute, since the transformation *f* in general will not (and should not) preserve the order relations of the original plaintext data. Specifically, it will be impossible for the outsourced RDBMS to answer range queries that cannot be reduced to multiple equality conditions (e.g., *1<=x<=3* can be translated into *x=1 or x=2 or x=3*) unless specific techniques are applied. In literature, there are several proposals for *f*, including:

1. *Domain partitioning* [24]*:* the domain is partitioned into equivalence classes, each corresponding to a single value in the codomain of *f*;
2. *Secure hashing* [13]: secure one-way hash function, which takes as input the clear values of an attribute and returns the corresponding index values. *f* must be deterministic and non-injective.

To handle range queries, a solution, among others, is to use an encrypted version of a B ± tree to store plaintext values, and maintain the values order. Because the values have to be encrypted, the tree is managed at the Client side and it is read-only in the Server side. Alternatively, the position information of each field in the original relation can be added to the encrypted data [33].

Let us, now, consider data protection strategies based on partitioning.

1) Data fragmentation

Normally, of all the outsourced data, only some columns and/or some relations are confidential, so it is possible to split the outsourced information in two parts, one for confidential and one for public data. Its aim is to minimize the computational load of encryption/decryption.

a) Non-communicating servers

In this technique, two *split databases* are stored, each in a different untrusted server (called, say, S_1 and S_2). The two untrusted servers need to be independent and non-communicating to prevent their alliance and reconstruction of the complete information. In this situation, the information may be stored as plaintext at each server.

Each Client query is decomposed in two subqueries: one for S_1 and one for S_2. The result sets have to be related and filtered, by the Client.

Figure 2 schematizes the resulting structure.

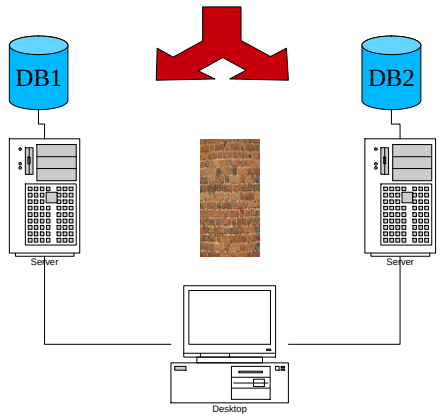


Figure 2. Non-communicating servers

b) Unlinkable fragments

In reality, it is not easy to ensure that split servers do not communicate; therefore the previous technique may be inapplicable. A possible remedy is to divide information in two or more fragments. Each fragment contains all the fields of the original information, but some are in clear form while the others are encrypted. To protect encrypted values from frequency attacks, a suitable *salt* is applied to each encryption. Fragments are guaranteed to be unlinkable (i.e., it is impossible to reconstruct the original relation and to determine the sensitive values and associations without the decrypting key). These fragments may be stored in one or more servers.

Each query is then decomposed in two subqueries:

- The first, executed on the Server, chooses a fragment (all fragments contain the entire information) and selects tuples from it according to clear text values.

It returns a result set where some fields are encrypted;

- The second, executed on the Client (only if encrypted fields are involved in the query), decrypts the information and removes the spurious tuples.

Figure 3 schematizes the resulting structure.

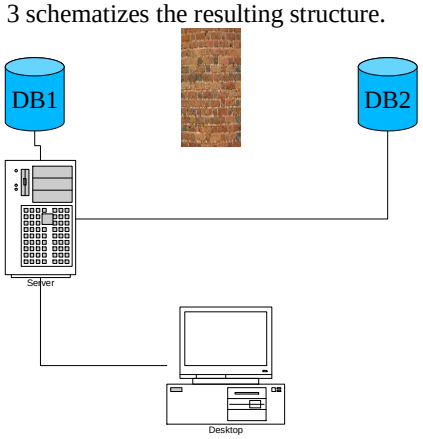


Figure 3. Unlinkable fragments

2) Data fragmentation with owner involvement

Another adaptation of the *non-communicating servers* technique consists of storing locally the sensitive data and relations, while outsourcing storage of the generic data. So, each tuple is split in a server part and in a local part, with the primary key in common. The query is then resolved as shown above.

Figure 4 schematizes the resulting structure.

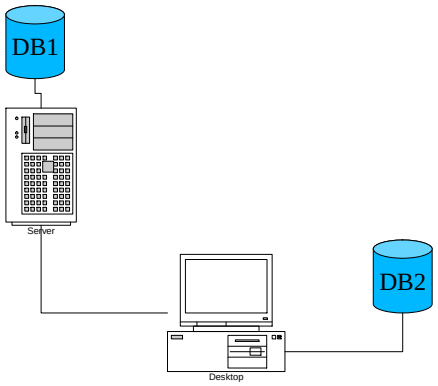


Figure 4. Data fragmentation with owner involvement

B. Selective access

In many scenarios, access to data is selective, with different users enjoying different views over the data. Access control can discriminate between read and write operations on an entire record or only on a part of it.

An intuitive way to handle this issue is to encrypt different portions of data with different keys that are then distributed to users according to their access privileges. To minimize overhead it is required that:

- No more than one key is released to each user;
- Each resource is encrypted not more than once.

To achieve these objectives, a hierarchical organization of keys can be envisioned. Basically, users with the same access privileges are grouped and each resource is encrypted with a key corresponding to the set of users that can access it. This way, a single key can be possibly used to encrypt more than one resource.

1) Dynamic rights management

Should the user's rights change over time (e.g., the user changes department) it is necessary to remove that user from a group/role as follows:

- Encrypt data by a new key;
- Remove the original encrypted data;
- Send the new key to the rest of the group.

Note that these operations must be performed by data owner because the untrusted DBMS has no access to the keys. This active role of the data owner goes somewhat against the reasons for choosing to outsource data in the first place.

a) Temporal key management

An important issue, common to many access control policies, concerns time-dependent constraints of access permissions. In many real situations, it is likely that a user may be assigned a certain role or class for only a limited time. In such case, users need a different key for each time period. A time-bound hierarchical key assignment scheme is a method to assign time-dependent encryption keys and private information to each class in the hierarchy in such a way that key derivation depends also on temporal constraints. Once a time period expires, users in that class should not be able to access any subsequent keys unless further authorized to do so [9].

b) Database replica

In [7], the authors, exploiting the never ending trend to a lower price-per-byte in storage, propose to replicate *n* times the source database, where *n* is the number of different roles having access to the database. Each database replica is a view, entirely encrypted using the key created for the corresponding role. Each time a role is created, the corresponding view is generated and encrypted with a new key expressly generated for the newly created role. Users do not own the real keys, but receive a token that allows them to address a request-to-cipher to a set KS of key servers on the cloud.

C. A document base sample: Crypstore

Crypstore is a non-transactional architecture for the distribution of confidential data, whose structure is shown in Figure 5. Crypstore's Storage Server contains data in encrypted form, so it cannot read them. User who wants to access data is authenticated at the Key Servers with the certificate issued by the Data Administrator and requires the decryption key. The Key Servers are *N* and, to ensure that none of them knows the whole decryption key, each of them contains only a part of the encryption key. To rebuild the key, only *M* (<*N*) parts of key are needed; redundancy

provides greater robustness to failures and attacks (e.g., Denial of Service attacks).

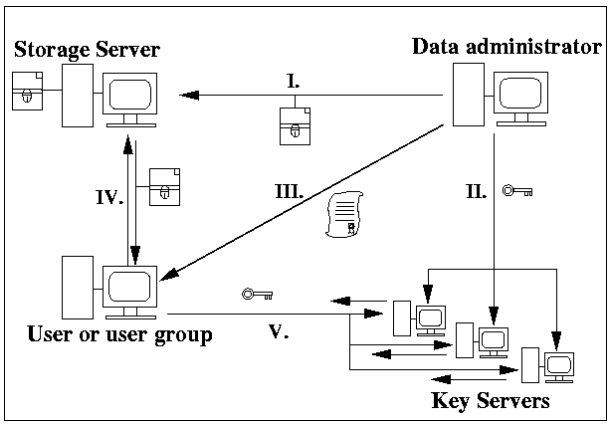


Figure 5. Crypstore

Really, Crypstore is an application of the time-honored "divide and conquer" technique, where data is separated from decryption keys.

Here privacy is not entirely guaranteed because, theoretically at least, the owner of Key Servers and the Storage Server may collude. The only way to exclude this (however remote) possibility is to have trusted Key Servers, but this would be equivalent to store the data directly, as plaintext, in a trusted storage. In practice, however, the probability of collusion decreases with the number of players involved and can be safely ignored in many cases.

II. PRIVACY WITHIN THE CLOUD

All techniques discussed above are based on data encryption and/or data fragmentation using full separation of roles and of execution environments between the user and the datastore (and possibly the keystore) used to manage the outsourced data.

Let us now compare the assumptions behind such techniques with two of the basic tenets of current cloud computing architectures: data and applications being on the "same side of the wall", and data being managed via semantic datastores rather than by a conventional RDBMS.

A. On the same side of the wall

Ubiquitous access is a major feature of cloud computing architectures. It guarantees that cloud application users will be unrestrained by their physical location (with internet access) and unrestrained by the physical device they use to access the cloud.

To satisfy the above requirements (in particular the second), we normally use thin clients, which run cloud applications remotely via a web user interface.

The three main suppliers of Public Cloud Infrastructure (Google App Engine for Business, Amazon Elastic Compute Cloud and Windows Azure Platform) all include a datastore, and an environment for remote execution summarized in Tables I and II:

TABLE I. DATASTORE SOLUTIONS USED BY PUBLIC CLOUDS

Environment	Datastore
Google	Bigtable
Amazon	IBM DB2 IBM Informix Dynamic Server Microsoft SQLServer Standard 2005 MySQL Enterprise Oracle Database 11g Others installed by users
Microsoft	Microsoft SQL Azure

TABLE II. EXECUTION ENVIRONMENTS USED BY PUBLIC CLOUDS

Environment	Execution environment
Google	J2EE (Tomcat + GWT) Python
Amazon	J2EE (IBM WAS, Oracle WebLogic Server) and others installed by users
Microsoft	.Net

In all practical scenarios, public cloud suppliers handle both data and application management.

If the cloud supplier is untrustworthy, it can intercept communications, modify executable software components (e.g., using aspect programming), monitor the user application memory, etc.

Hence, available techniques for safely outsourcing data to untrusted DBMS no longer guarantee the confidentiality of data outsourced to the cloud.

The essential point consists in having the data and the user interface application logic *on the same side of the wall* (see Figure 6).

This is a major difference w.r.t. outsourced database scenarios, where presentation was handled by trusted clients. In the end, the data must be presented to the user in an intelligible and clear form; that is the moment when a malicious agent operating in the cloud has more opportunities to intercept the data. To prevent unwanted access to the data at presentation time, it would be appropriate moving the presentation logics off the cloud to a trusted environment that may be an intranet or, at the bottom level, a personal computer.

However, separating data (which would stay in the cloud) from the presentation logics may enable the creation of local copies of data, and lead to an inefficient cooperation between the two parts.

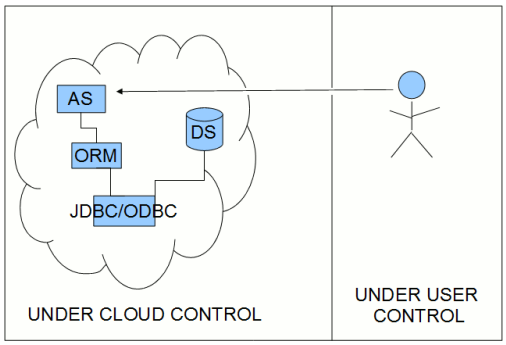


Figure 6. The wall

B. Semantic datastore

Cloud computing solutions largely rely on semantic (non-relational) DBMS. These systems do not store data in tabular format, but following the natural structure of objects. After more than twenty years of experimentation (see, for instance, [10] for the Galileo system developed at the University of Pisa), today, the lower performance of these systems is no longer a problem. In the field of cloud computing, there is a particular attention to Google Bigtable.

"*Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers. In many ways, Bigtable resembles a database: it shares many implementation strategies with databases.*" [11]

With a semantic datastore like Bigtable, there is a more strict integration between in-memory data and stored-data; they are almost indistinguishable from the programmer viewpoint. There are not distinct phases when the program loads data from disk into main memory or, in the opposite direction, when program serialize data on disk. Applications do not even know where the data is stored, as it is scattered over the cloud.

In such a situation, the data outsourcing techniques discussed before cannot be applied directly, because they were designed for untrusted RDBMS.

III. OUR APPROACH

We are now ready to discuss our new approach to the issue of cloud data privacy. We build over the notion introduced in [7] of defining a view for every user group/role, but we prevent performance degradation by keeping all data views in the user environment.

Specifically, we atomize the application/database pair, providing a copy per user. Every instance runs locally, and maintains only authorized data that is replicated and synchronized among all authorized users.

In the following subsections we will analyze our solution in detail.

A. Information sharing by distributed system

We will consider a system composed of:

1. Local agents distributed at client side;
2. A central synchronization point.

Figure 7 shows the proposed architecture:

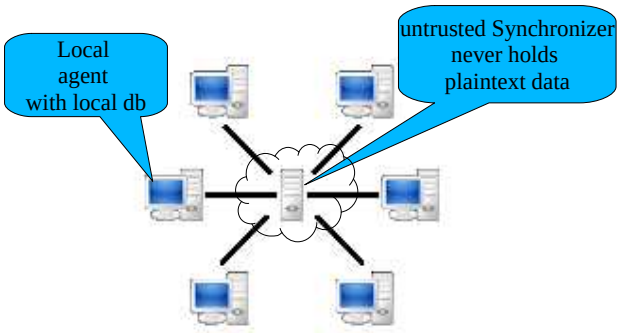


Figure 7. The architecture

1) The model

Henceforth, we will use the term *dossier* to indicate a set of related information. Our data model may be informally represented by the diagram in Figure 8.

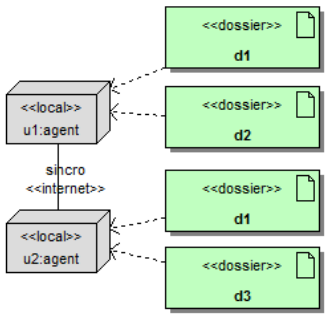


Figure 8. The model

In our model, each node represents a local, single-user application/database dedicated to an individual user (u_n). The node stores only the dossiers that u_n owns. Shared dossiers (in this example, d_1) are replicated on each node. When a node modifies a shared dossier, it must synchronize, also using heuristics and learning algorithms, with the other nodes that hold a copy of it. Below we give a simple SWOT analysis of this idea.

2) Strength/Opportunities

- Information sharing using untrusted Synchronizer;
- Small amount of local data, less attractive for attackers;
- Only the final user has clear-text information;
- Unrestrained individual nodes, that can also work offline (with deferred synchronization);
- Simplicity of data management (single user);
- Completeness of local information.

To clarify the last point, suppose that the user u_n wants to know the number of the dossier she is treating. In a classic intranet solution, where dossiers reside on their owners' servers, in addition to its database, u_n should examine the

data stores of all other collaborating users. With our solution, instead, u_n will simply perform a local query because the dossiers are replicated at each client.

3) Weaknesses/Threats

- Complexity of deferred synchronization schemes [21];
- Necessity to implement a mechanism for grant/revoke and access control permissions.

This last point is particularly important and it deserves further discussion:

- Each user (except the data owner) may have partial access to a dossier. Therefore each node contains only the allowed portion of the information;
- Authorization, i.e., granting to a user u_j access to a dossier d_k, can be achieved by the data owner simply by transmitting to each node only the data it is allowed to access;
- The inverse operation can be made in the case of a (partial or complete) revocation of access rights. An obvious difficulty lies in ensuring that data becomes no longer available to the revoked node. This is indeed a moot point, as it is impossible – whatever the approach - to prevent trusted users from creating local copies of data while they are authorized and continue using them after revocation. We are evaluating the opportunity to use watermarking for relational databases [26] to provide copyright protection and tamper detection.

B. Proposed solution

We are now ready to analyze our solution in detail. To simplify the discussion, we introduce the following assumptions:

- Each dossier has only one owner;
- Only the dossier's owner can change it.

These assumptions permit the use of an elementary cascade synchronization in which the owner will submit the changes to the receivers. However, they can be relaxed at the cost of a higher complexity in synchronization [34].

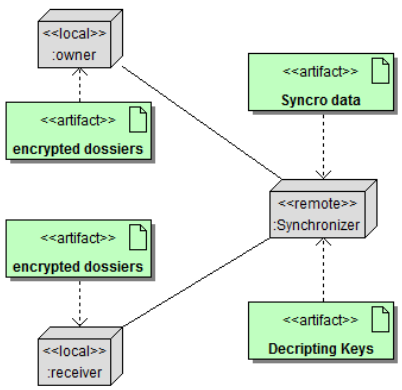


Figure 9. Deployment diagram of distributed system

Our solution consists of two parts: a trusted client and a remote untrusted synchronizer (see Figure 9).

The client maintains local data storage where:

- The dossiers that she owns are (or at least can be) stored as plain-text;
- The others, instead, are encrypted each with a different key.

The Synchronizer stores the keys to decrypt the shared dossiers owned by the local client and the modified dossiers to synchronize.

When another client needs to decrypt a dossier, she connects to the Synchronizer and obtains the corresponding decryption key.

The data and the keys are stored in two separate entities, none of which can access information without the collaboration of the other part.

1) Structure

From the architectural point of view, we divide our components into two packages, a local (client agent), which contains the dossier plus additional information such as access lists, and a remote (global synchronizer), which contains the list of dossiers to synchronize, their decryption keys and the public keys of clients.

A UML view of involved classes is shown in Figure 10.

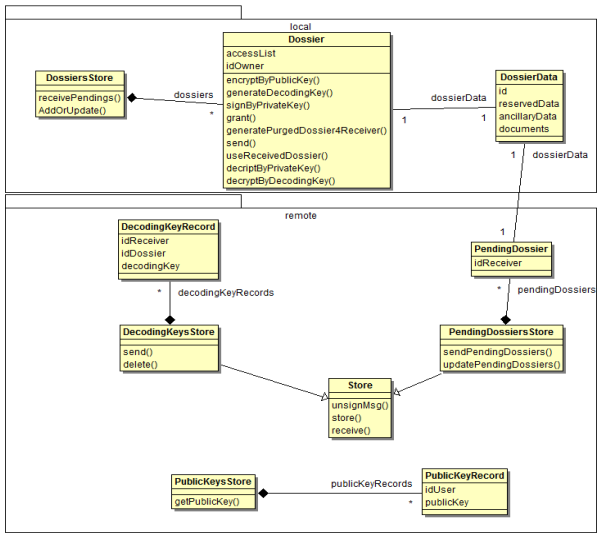


Figure 10. Class view

2) Grant

An owner willing to grant rights on a dossier must follow the sequence shown in Figure 11:

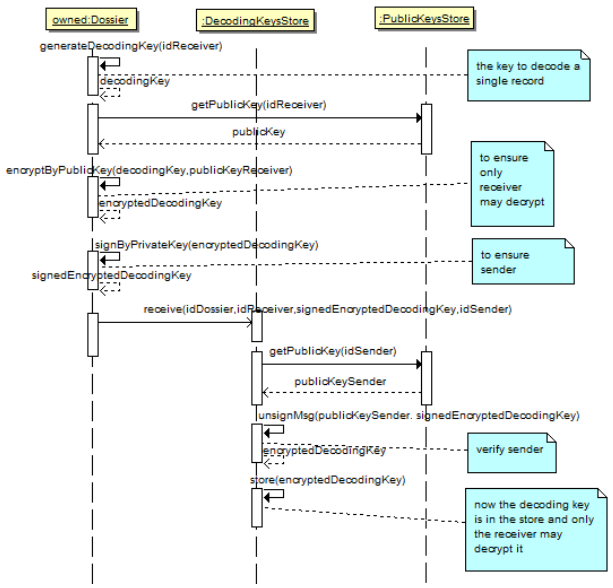


Figure 11. Grant sequence

Namely, for each receiver, the owner:

- Generates the decryption key
- Encrypts it with the public key of the receiver to ensure that others cannot read it
- Signs it with its private key to ensure its origin
- Sends it to the Synchronizer, which verifies the origin and adds it to the storage of the decoding keys. The key is still encrypted with the public key of the receiver, so only the receiver can read it.

3) Send

When an owner modifies a dossier, she sends it to the Synchronizer the sequence shown in Figure 12:

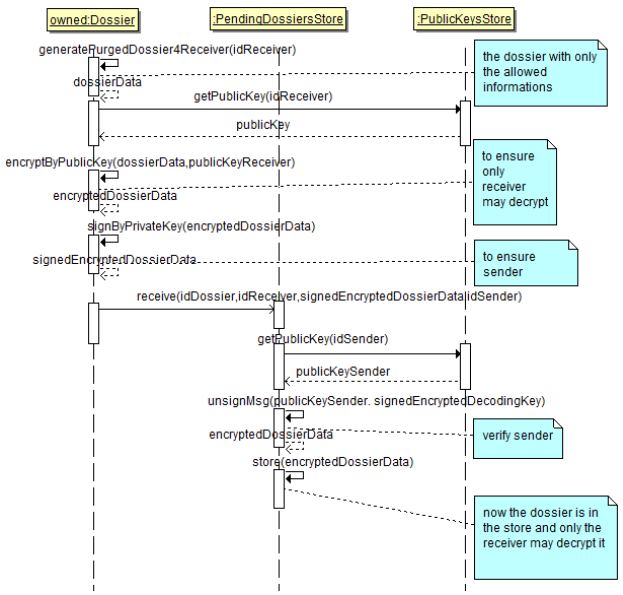


Figure 12. Send sequence

For each receiver, the owner:

- Generates a "pending dossier" by removing information that the receiver should not have access to;
- Encrypts the pending dossier with the previously generated decryption key;
- Signs with his own private key to certificate its origin;
- Sends it to the Synchronizer, which verifies the origin and adds it to the storage of "pending dossiers". Again, the dossier is still encrypted with the public key of the receiver, so only the receiver can read it.

4) Receive

Periodically, each client updates un-owned dossiers by following the sequence shown in Figure 13:

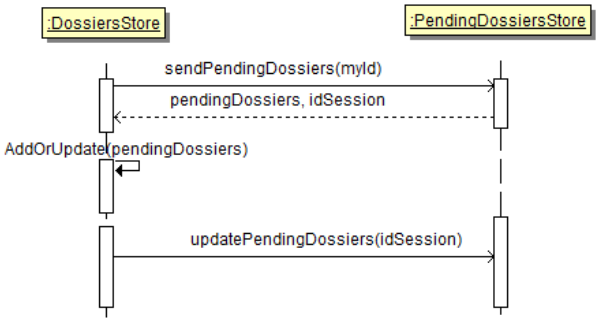


Figure 13. Receive sequence

Each client:

- Requests the "pending dossiers" to the Synchronizer.
- Stores the (still encrypted) dossier in the local storage;
- Removes the received dossiers from the Synchronizer.

5) Use

When a client needs to access an unowned (encrypted) dossier, the sequence shown in Figure 14 is used:

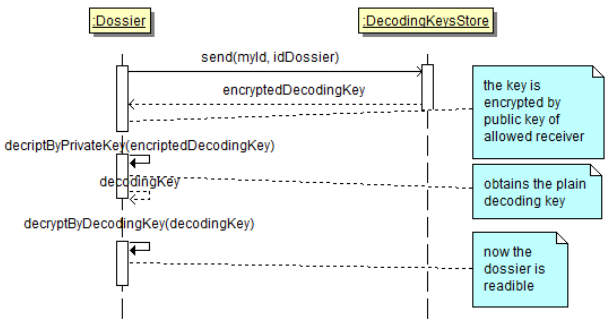


Figure 14. Use sequence

The client:

- Asks the Synchronizer for the decryption key (that is encrypted by her public key);
- Decrypts it with her private key;
- Decrypts the dossier by the resulting decryption key.

If the decryption key does not exist, two options are available:

- The record is deleted from the local datastore because a revoke happened;
- The record remains cached (encrypted) into the local datastore because access rights to it could be restored.

6) Revoke

To revoke access to a receiver, it is sufficient to delete the corresponding decryption key from the Synchronizer. The sequence diagram is shown in Figure 15.

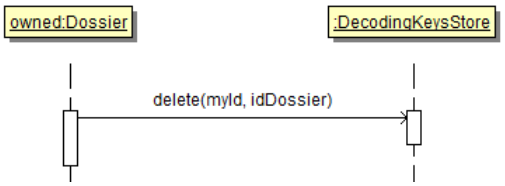


Figure 15. Revoke sequence

IV. EXPERIMENTATION

To experiment with our architecture we implemented the custom client and Synchronizer. The client needs to use row-level encryption. In a normal RDBMS, however, this technique has significant disadvantages in terms of performance and functionality: querying would be possible only through the construction of appropriate indexes for each column of the table (with a considerable waste of resources both in terms of time and space), while the constraints and foreign keys would be almost unusable.

Another major issue concerns the management of keys: row-level encryption could potentially lead to the generation and maintenance (and / or distribution) of a key for each row of each table encrypted with this method. To solve (or reduce) the concern, we use some advanced techniques of key management, such as:

- Broadcast (or Group) encryption [32]: rows are divided into equivalence classes, based on recipients. Every class is encrypted using an asymmetric algorithm where the encryption key is made in a way that each recipient can decrypt the information using only its own private key. Both the public and the private keys are generated by a trusted entity.
- Identity Based Encryption [30]: it bounds the encryption key to the identity of the recipient. Each recipient generates by itself a key pair used to encrypt/decrypt information.
- Attribute Based Encryption [31]: it bounds the encryption key to an attribute (a group) of recipient. Each recipient receives from a trusted entity the private key used to decrypt, while the sender calculates the encryption key.

The complexity of these techniques is a major reason why conventional RDBMSs do not use encryption at the row-level.

A. In memory databases

An in-memory database (IMDB, also known as main memory database system or MMDB and as real-time database or RTDB) is a database management system that primarily relies on main memory for computer data storage [35]. It is important to remark that, while a conventional database system stores data on disk but caches it into memory for access, in an IMDB the data resides permanently in the main physical memory and there is a backup copy on disk [27].

In-memory databases have recently become an intriguing topic for the database industry. With the mainstream availability of 64-bit servers with many gigabytes of memory a completely RAM based database solution is a tempting prospect to a much wider audience [36].

IMDBs are intended either for personal use (because they are comparatively small w.r.t. traditional databases), or for performance-critical systems (for their very low response time and very high throughput). They use main memory structures, so they need no translation from disk to memory form, and no caching and they perform better than traditional DBMSs with Solid State Disks.

Normally, the use of volatile memory-based IMDBs supports the three ACID properties of atomicity, consistency and isolation, but lacks support for the durability property. To add this when non-volatile random access memory (NVRAM) is not available, IMDBs use a combination of transaction logging and primary database check-pointing to the system's hard disk: they log changes from committed transactions to physical medium and, periodically, update a disk image of the database. Having to write updates to disk, the write operations are heavier than read-only. Logging policies vary from product to product: some leave the choice of when to write the application on file, others do all the checkpoints at regular intervals of time or after a certain amount of data entered / edited.

In Table III, we summarize pros and cons for IMDBs.

TABLE III. IMDBS PROS AND CONS

Pros	Cons
Fast transactions No translation High reliability Multi-User Concurrency (few locks)	Complexity of durability's implementation Size limited by main memory

Obviously, the limitation of this type of database is related to the amount of RAM on computer hosting the db. But given their nature, IMDBs are well suited to be distributed and replicated across multiple nodes to increase capacity and performance.

The proposed solution works around this limitation: not having a single central database containing the whole data,

we preferred to give one database for each client application. This database contains only owned data, while external data will be added (or removed) via the synchronizer, based on access permissions.

To minimize cryptography overhead, we encrypt only rows "received" by other nodes, while rows owned by the local node are stored in cleartext form.

Well-known open solutions of IMDB are Apache Derby, HyperSQL (HSQLDB) and SQLite. For our implementation, we chose to use HyperSQL rel. 2.0.

1) HyperSql

HyperSQL [37] is a pure Java RDBMS. Its strength is, besides the lightness (about 1.3Mb for version 2.0), the capability to run either as a Server instance either as a module internal to an application (in-process).

A database started "in-process" has the advantage of speed, but it is dedicated only to the containing application (no other application can query the database). For our purposes, we chose server mode. In this way, the database engine runs inside a JVM and will start one or more "in-process" databases, listening requests from processes in the local machine or remote computers.

For interactions between clients and database server, we can use three different protocols:

- HSQL Server: the fastest and most used. It implements a proprietary communication protocol;
- HTTP Server: it is used when access to the server is limited only to HTTP. It consists of a web server that allows JDBC clients to connect over http;
- HTTP Servlet: as the Http Server, but it is used when accessing the database is managed by a servlet container or by an application servlet (e.g., Tomcat). It is limited to using a single database.

Several different types of databases (called catalogs) can be created with HyperSQL. The difference between them is the methodology adopted for data storage:

- Res: this type of catalog provides for the storage of data into small JAR or ZIP files;
- Mem: data is stored completely in the machine's RAM, so there is no persistence of information outside of the application life cycle in the JVM;
- File: data is stored in files residing into the file system of the machine.

In our work, we used the latter type of databases.

A catalog file can use up to six files on the file system for its operations. The name of these files consists of the name of the database plus a dot suffix.

Assuming we have a database called "db_test", files will be:

- db_test.properties containing the basic settings of the DB;
- db_test.log: used to periodically save data from the database, to prevent data loss in case of a crash;
- db_test.script: containing the table definitions and other components of the DB, plus data of not-cached tables;
- db_test.data: containing the actual data of cached tables. It can be not present in some catalogs;
- db_test.backup: containing the compressed backup of last ".data" file, that may be not present in some catalogs;
- db_test.lobs: used for storing BLOB or CLOB fields.

Besides these files, HyperSQL can connect to CSV files.

A client application can connect to HyperSQL server using the JDBC driver (.Net and ODBC drivers are "in late stages of development"), specifying the type of database to access (file, mem or res).

HyperSQL implements the SQL standard either for temporary tables either for persistent ones. Temporary tables (TEMP) are not stored on the file system and their life cycle is limited to the duration of the connection (i.e., of the Connection object). The visibility of data in a TEMP table is limited to the context of connection used to populate it. With regard to the persistent tables, instead, HyperSQL provides three different types of tables, according to the method used to store the data:

- MEMORY: it is the default option when a table is created without specifying the type. *Memory table* data is kept entirely in memory, while any change to its structure or contents is recorded in .log and .script files. These two files are read at the opening of database to load data into memory. All changes are saved when closing the database. These processes can take a long time in the case of tables larger than 10 MB.
- CACHED: when this type of table is chosen, only part of the data (and related indexes) is stored in memory, thus allowing the use of large tables at the expense of performance.
- TEXT: the data is stored in formatted files such as .csv.

In our implementation, we use MEMORY tables.

The Loader and the Serializer are the main parts of HyperSQL that we analyzed and modified. They are the mechanisms that load the data from text files at the opening and save them to the database at closing.

B. Implemented solution

1) Client side

On the client side, using IMDBs, we have only two interactions between each local agent and the Synchronizer, as shown in Figure 16.

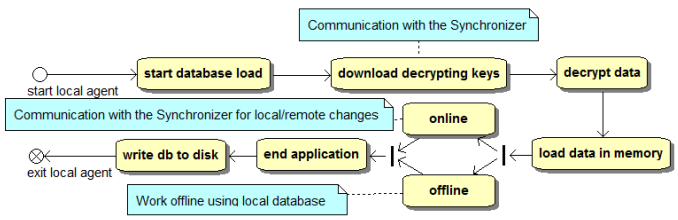


Figure 16. State diagram of client

We have modified the classes included in file hsqldb.jar to handle encryption. The basic idea was to manage encryption in the .log and .script text files. The rows that are

owned by the local client are stored in clear-text, while the shared rows "granted" by other owners are stored encrypted.

The values contained in tables are stored in form of SQL insert:

```
INSERT INTO table_name(field_1, field_2, …, field_n)
VALUES(value_1, value_2, …, value_n)
```

Earlier, to obtain control access granularity at the field level, we encrypted field by field. This way, the text contained in the database file is in the form of:

```
INSERT INTO table_name(field_1, field_2, …, field_n)
VALUES(pk, encrypted_value_2, …, encrypted_value_n)
```

The primary key *pk* needs to be in clear-text, since it is used to retrieve the decrypting keys from the central Synchronizer. We dropped this idea because it requires changing the I/O code for each possible database type and an attacker may obtain some information such as table, primary key and number of rows.

Our current solution is to encrypt the whole row by AES symmetric algorithm. The encryption overhead is lower than the previous solution and all information is hidden to curious eyes. To relate the encrypted row (stored locally) to the decrypting key (stored in the remote Synchronizer), we use a new key (*id_pending_row*). The encrypted row is prefixed by a clear-text header containing the id_pending_row delimited by "$" and "@". The encrypted value is then stored in a hexadecimal representation, so a generic row is of the form:

```
$27@5DAAAED5DA06A8014BFF305A93C957D
```

a) Load time

At load time, the .script file will contain clear-text and encrypted rows, e.g.:

```
INSERT INTO students(id,name) VALUES(12,'Alice');
INSERT INTO students(id,name) VALUES(31,'Bob');
$27@5F3C25EE5738DAAAED5DA06A80F305A93C95A
$45@5DA67ADA06AAED580FA914BF3C953057D387F
INSERT INTO students(id,name) VALUES(23,'Carol');
```

The class whose task is reading the file and loading the appropriate data in memory is *ScriptReaderText.*

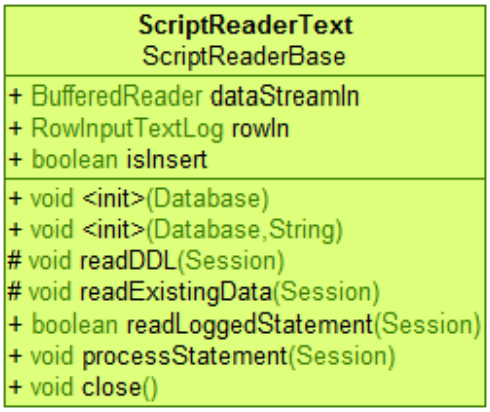


Figure 17. UML of ScriptReaderText class

The *readLoggedStatement* method parses each line of text in the .log or .script files and forwards the result to the *processStatement* method, which loads data into memory.

We changed the *readLoggedStatement* method to make a preprocessing: if it finds a record header (enclosed between $ and @) in the text line, it extracts the *id_pending_row_received*. Using this id, the client requests to the central Synchronizer the related decoding key, which it uses to decrypt the entire text line and to proceed with normal HyperSQL management. If the decoding key is unavailable, the text line is temporarily discarded (it is not deleted if it was not received for communication problem with the Synchronizer).

b) Save time

The class ScriptWriterText manages the write operations in .log and .script files.

The affected methods are *writeRow* and *writeRowOutToFile.*

The former deals with building the string that will be written into the text file (INSERT INTO ....) wich corresponds to the in memory data. A *Table* instance contains the information about the table structure (table name, field names, types of data, constraints, etc.). The values of fields are in an array of *Object.* The SQL *insert* is written in a text buffer that is stored in the .script file by the method *writeRowOutToFile*. Because each table has an *id_pending_row_received* column, we modified the *writeRow* method to check if the row is owned or shared by another user. In the latter case (*id_pending_row_received* not null), the custom *writeRowOutToFileCrypto* method is used instead of the *writeRowOutToFile* method. *WriteRowOutToFileCrypto* uses the parameter *id_pending_row_received* to query the related symmetric encryption key from the Synchronizer, needed to encrypt the whole buffer. The result is a hexadecimal sequence, which is prefixed by the below header with the id_*pending_row_received.*

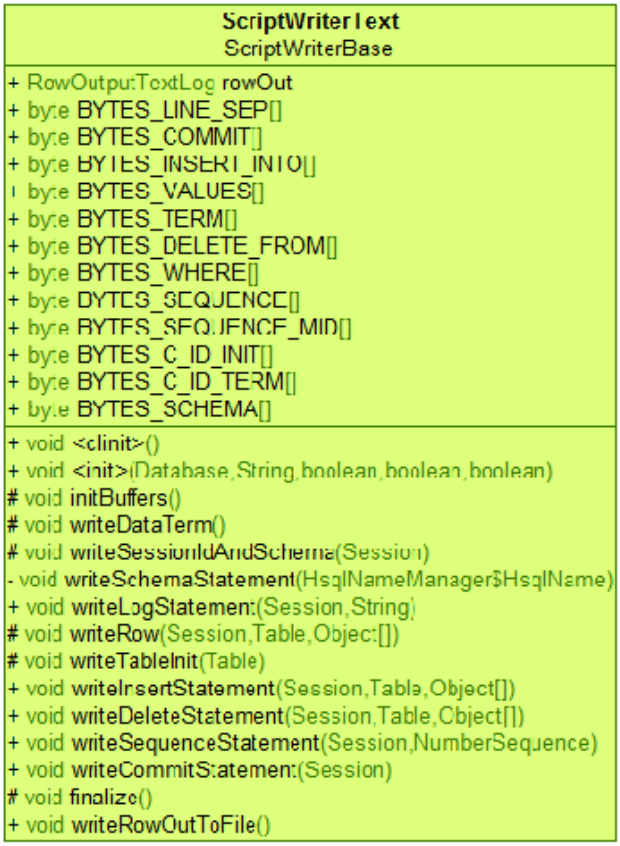


Figure 18. UML of ScriptWriterText class

2) Server side

When a data owner adds or updates a row in the local database, it needs to distribute this change to all the related users. To do this, we put the cloud a central Synchronizer server that acts as a mailbox.

It uses a simple database with the following tables:

- Users: containing, among others, the id and public key of each user;
- Pending Rows: it contains the rows that are added/modified in the local database of the owner, until they are delivered to destination. A unique row_id is automatically assigned to each pending row. Other information is submission date, sender and receiver. The changed row is stored in encrypted form in field encrypted_row;
- Decrypting keys: contains the keys that are used to decrypt the pending rows. Other information is: sender, receiver, expiry date, id_row.

At modification time, the owner (client side) has to:

- Serialize the row;
- Generate a symmetric key to encrypt it;
- Encrypt the row;
- Encrypt the key by the public keys of receivers;
- Send the encrypted row and the decoding keys to receiver.

Because we store the serialized row, we haven't to worry about columns data types.

The Synchronizer uses RMI to expose its services to clients. The services are grouped in three interfaces:

- KeyInterface with methods related to encryption keys: depositKey, deleteDecriptingKey, getDecriptingKeyByIdPendingRow, getPublicKeyByUser;
- SynInterface with methods for sharing the rows: sendRow, getPendingRowForUser, getAllUsers, resendRow;
- RegistrationInterface to register and manage users: registerUser, SelectUserById, selectUserByIdAndPassword.

C. Performances

1) Read operations

The system uses decryption only at start time, when records are loaded from the disk into the main memory. Each row is decrypted none (if it is owned by local node) or just once (if it is owned by a remote node), so this is optimal for read operations. Each decryption implies an access to the remote Synchronizer to download the related decrypting key and, eventually, the modified row.

2) Write operations

Write operations occur when a record is inserted / updated into the db. There is no overload until the client, when online, explicitly synchronizes data with the central server. At this moment, for each modified record, the client need to:

- Generate a new (symmetric) key
- Encrypt the record
- Dispatch the encrypted data and the decrypting key to the remote synchronizer

3) Benchmark

We wrote a test application that uses our modified HyperSQL driver and interacts with the other clients through our Synchronizer. It has these distinct activities:

- Creation of database and sample tables
- Population of tables with sample values
- Sharing of a portion of data with another user
- Receipt of shared dossiers from other users
- Opening of the newly created (and populated) database

The application receives three parameters:

- Number of dossiers
- Number of clients involved in sharing
- Percentage of shared dossiers

To minimize communication delay, the central Synchronizer and the clients ran on the same computer. For testing purpose, it was sufficient to use only two clients (to enable data sharing). The tests used a number of dossiers varying from 1,000 to 500,000. We tested the system with 20% and 40% of shared dossiers.

The application was compared with an equivalent one with the following differences:

- It used the unmodified HyperSQL driver
- It did not share data with other clients
- When populating the database, it created the same number of dossiers than the previous application; but, after benchmarking, it added the number of shared dossiers to have the same final number of dossiers.

We benchmarked the system using single-table dossiers of about 200 bytes, in two batteries of tests; the first with 20%, and the second with 40% of shared dossiers, which numbered from 1,000 to 500,000. The results are represented by the graphs in Figures 19-21. It is worth noting that the overhead percentage of the modified solution rapidly decreases (with 100,000 dossiers it is around 10%), either in the first battery of tests (Figure 19), and either in the second (Figure 20). In the tests, the total delay (load + create + populate + receive) stay linear in the number of dossiers and is limited, even with a huge number of dossiers (Figure 21). Local results can be slightly altered by external events not preventable (e.g., garbage collector).

D. Results

The delay of the system is tightly bound to communications effort with the central Synchronizer. Computing overhead is limited to just one encryption per record at write time and no more than one decryption per record at read time. Since I use symmetric encryption, these operations are very fast. The benchmark demonstrates that the delay is substantially concentrated in database opening, while the subsequent use does not involve additional delays, compared to the unmodified version.

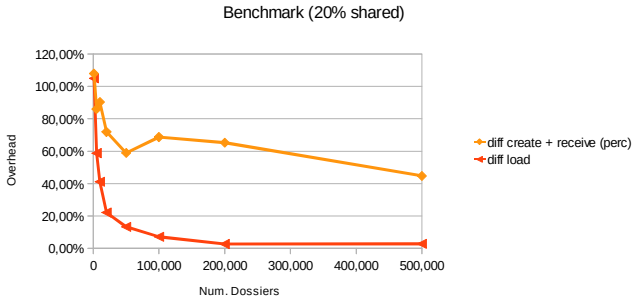


Figure 19. Overhead when 20% of dossiers are shared

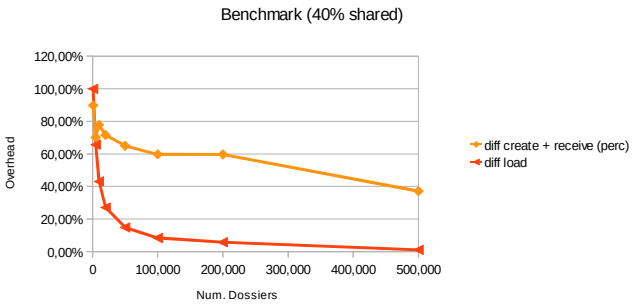


Figure 20. Overhead when 40% of dossiers are shared

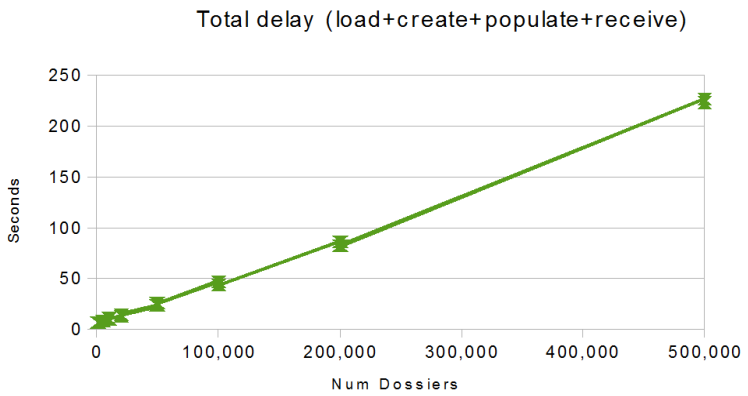


Figure 21. Total delay

V. CONCLUSIONS AND OUTLOOK

In this paper, we discussed the applicability of outsourced DBMS solutions to the cloud and provided the outline of a simple yet complete solution for managing confidential data in public clouds.

We are fully aware that a number of problems remain to be solved. A major weakness of any data outsourcing scheme is the creation of local copies of data after it has been decrypted. If a malicious client decrypts data and then it stores the resulting plain-text data in a private location, the protection is broken, as the client will be available to access its local copy after being revoked. In [22], obfuscated web presentation logic is introduced to prevent client from harvesting data. This technique, however, exposes plaintext data to cloud provider. The plain-text data manager is always the weak link in the chain and any solution must choose whether to trust the client-side or the server-side. A better solution [26] is to watermark the local database to provide tamper detection.

Another issue concerns the degree of trustworthiness of the participants. Indeed, untrusted Synchronizer never holds plain-text data; therefore it does not introduce an additional Trusted Third Party (TTP) with respect to the solutions described at the beginning of the paper. However, we need to trust the Synchronizer to execute correctly the protocols explained in this paper. This is a determining factor that our

technique shares with competing solutions and, although an interesting topic, it lies beyond the scope of this paper.

In experiment phase, we introduced a simple solution to row-level encryption of databases using IMDBs. It can be used in the cloud to manage very granular access rights in a highly distributed database. This allows for stronger confidence in the privacy of shared sensitive data.

An interesting field of application is the use in (business) cooperative environments, e.g., professional networks. In these environments, privacy is a priority, but low computing resources don't allow the use of slow and complex algorithms. IMDBs and our smart encryption, instead, achieve the goal in a more effective way.

REFERENCES

[1] E. Damiani and F. Pagano, "Handling confidential data on the untrusted cloud: an agent-based approach," Cloud Computing 2010, pp. 61-67. Lisbon, 2009. IARIA.

[2] D. Pagano and F. Pagano, "Using in-memory encrypted databases on the cloud," in press

[3] M. Armbrust, A. Fox, R. Griffith, Anthony D. Joseph, Randy H. Katz, Andy Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia: "A view of cloud computing", Commun. ACM 53(4), pp. 50-58 (2010)

[4] C. Jackson, D. Boneh, and J.C. Mitchell: "Protecting Browser State from Web Privacy Attacks", 15th International World Wide Web Conference (WWW 2006), Edinburgh, May, 2006.

[5] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu: "Data Management for Peer-to-Peer Computing : A Vision", WebDB 2002, pp. 89-94

[6] Pierangela Samarati and Sabrina De Capitani di Vimercati: "Data protection in outsourcing scenarios: issues and directions", ASIACCS 2010, pp. 1-14

[7] Nadia Bennani, Ernesto Damiani, and Stelvio Cimato: "Toward cloud-based key management for outsourced databases", SAPSE 2010, draft

[8] Mikhail J. Atallah, Marina Blanton, and Keith B. Frikken: "Incorporating Temporal Capabilities in Existing Key Management Schemes", ESORICS 2007, pp. 515-530

[9] Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci: "New constructions for provably-secure time-bound hierarchical key assignment schemes", Theor. Comput. Sci. 407, pp.213-230 (2008)

[10] Antonio Albano, Giorgio Ghelli, M. Eugenia Occhiuto, and Renzo Orsini: "Object-Oriented Galileo", On Object-Oriented Database System 1991, pp. 87-104

[11] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Michael Burrows, Tushar Chandra, Andrew Fikes, and Robert Gruber: "Bigtable: A Distributed Storage System for Structured Data", OSDI 2006, pp. 205-218

[12] Victor R. Lesser: "Encyclopedia of Computer Science", 4th edition. John Wiley and Sons Ltd. 2003, pp.1194–1196

[13] Ernesto Damiani, Sabrina De Capitani di Vimercati, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati:"Balancing confidentiality and efficiency in untrusted relational DBMSs",ACM Conference on Computer and Comm. Security 2003, pp. 93-102

[14] Ernesto Damiani, Sabrina De Capitani di Vimercati, Mario Finetti, Stefano Paraboschi, Pierangela Samarati, and Sushil Jajodia: "Implementation of a Storage Mechanism for Untrusted DBMSs", IEEE Security in Storage Workshop 2003, pp. 38-46

[15] Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, and Pierangela Samarati: "Privacy of outsourced data", In Alessandro Acquisti, Stefanos Gritzalis, Costos Lambrinoudakis, and Sabrina De Capitani di Vimercati: Digital Privacy: Theory, Technologies and Practices. Auerbach Publications (Taylor and Francis Group) 2007

[16] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati: "Fragmentation and Encryption to Enforce Privacy in Data Storage", ESORICS 2007, pp. 171-186

[17] Richard Brinkman, Jeroen Doumen, and Willem Jonker: "Using Secret Sharing for Searching", in Encrypted Data. Secure Data Management 2004, pp. 18-27

[18] Ping Lin and K. Selçuk Candan: "Secure and Privacy Preserving Outsourcing of Tree Structured Data", Secure Data Management 2004, pp. 1-17

[19] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati: "Combining fragmentation and encryption to protect privacy in data storage", ACM Trans. Inf. Syst. Secur. 13(3): (2010)

[20] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati: "Keep a Few: Outsourcing Data While Maintaining Confidentiality", ESORICS 2009, pp. 440-455

[21] Miseon Choi, Wonik Park, and Young-Kuk Kim: "A split synchronizing mobile transaction model", ICUIMC 2008, pp.196-201

[22] Henk C. A. van Tilborg: "Encyclopedia of Cryptography and Security", Springer 2005

[23] Ian T. Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu: "Cloud Computing and Grid Computing 360-Degree Compared CoRR", abs/0901.0131: (2009)

[24] Hakan Hacigümüs, Balakrishna R. Iyer, Chen Li, and Sharad Mehrotra: "Executing SQL over encrypted data in the database-service-provider model", SIGMOD Conference 2002, pp. 216-227

[25] Dirk Düllmann, Wolfgang Hoschek, Francisco Javier Jaén-Martínez, Ben Segal, Heinz Stockinger, Kurt Stockinger, and Asad Samar: "Models for Replica Synchronisation and Consistency in a Data Grid", HPDC 2001, pp. 67-75

[26] Raju Halder, Shantanu Pal, and Agostino Cortesi: "Watermarking Techniques for Relational Databases: Survey, Classification and Comparison", in Journal of Universal Computer Science, vol. 16 (21), pp. 3164-3190

[27] H. Garcia-Molina and K. Salem, "Main Memory Database Systems: An Overview," IEEE Trans. Knowl. Data Eng. 4(6), 1992, pp. 509-516

[28] L. Bouganim and Y. Guo, "Database encryption," in Encyclopedia of Cryptography and Security, Springer, 2010, 2nd Edition

[29] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Key management for multi-user encrypted databases," StorageSS, 2005, pp. 74-83

[30] D. Boneh and M. Hamburg, "Generalized Identity Based and Broadcast Encryption Schemes," ASIACRYPT, 2008, pp. 455-470

[31] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded Ciphertext Policy Attribute Based Encryption," ICALP, 2008, pp. 579-591

[32] A. Fiat and M. Naor, "Broadcast Encryption," CRYPTO, 1993, pp. 480-491

[33] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Computing range queries on obfuscated data," IPMU, 2004

[34] C. Pu and A. Leff, "Replica Control in Distributed Systems: An Asynchronous Approach," SIGMOD, 1991, pp. 377-386

[35] http://en.wikipedia.org/wiki/In-memory_database. Retrieved 2011-07-22

[36] http://www.remote-dba.net/t_in_memory_cohesion_ssd.htm. Retrieved 2011-07-22

[37] www.hsqldb.org. Retrieved 2011-07-22

A Trust-based Approach for Secure Packet Transfer in Wireless Sensor Networks

Yenumula B. Reddy
Grambling State University
Grambling, LA 71245, USA
ybreddy@gram.edu

Rastko R. Selmic
Louisiana Tech University
Ruston, LA 71270, USA
rselmic@latech.edu

***Abstract*—Trust is an important factor in transferring data from the source to destination in wireless sensor networks. If any sensor node fails to transfer the data, the Dynamic Source Protocol calculates the alternate path. Currently, the Dynamic Source Protocol does not have any built-in functionality to calculate an alternate path if the path has a malicious node. Intruder detection system can detect the malicious node. However, intruder detection system is very expensive for wireless sensor networks and there is no guarantee in detecting a malicious node. In the current research, a trust-based approach is recommended to minimize the overheads of intruder detection system and detect the abnormal behavior nodes. The proposed model uses the repeated games to detect faulty (malicious) nodes through the cooperative effort in the sensor network and judges the trust of successive nodes. Further, the research includes the trust model with reliable neighbors and query-based trust calculation. Simulations were presented for normalized payoff of packet dropping, average discount payoff, and trust relation.**

Keywords – wireless sensor networks; repeated games; packet transfer; trust-based approach; secure transfer of data.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are used to collect important data in sensitive areas including military surveillance, fire monitoring dangerous forests, and hazardous places including biological and chemical areas [42-46]. Secure communication is required for these applications, since sensors are deployed massively and unorganized way [1-6]. Due to the unorganized massive deployment, the black holes are common and malicious nodes will be created through hackers. Most of the times, eliminating the malicious nodes or further deployment of sensors at sink holes is very difficult [37].

WSNs are used in different applications including Structural Health Monitoring (SHM), Industrial Automation (IA), Civil Structure Monitoring (CSM), Military Surveillance (MS), and monitoring the Biologically Hazardous Places (BHP) [47-52]. In CSM, MS, and BHP the data is transferred over a number of nodes and any malicious node in the path leads to a dangerous situation. Due to the WSN topology, injecting bad nodes is not difficult. Therefore, there is a need to create a secure transmission model with minimum overheads and transmit the data securely.

Design of secure communication model with minimum overheads is very difficult [53]. The information security models (Intruder Detection System (IDS) and cryptography techniques) for wireless communications are not suitable to WSNs due to resource (processing and memory) limitations. Further the WSNs topology changes dynamically due to failure of nodes and the distance between the nodes is limited. Due to limited distance, frequent failure of nodes, and possible injection of malicious nodes, the trust of successive nodes and cooperation of neighboring nodes is very important.

The trust depends upon the predictable behavior of successive nodes [1]. The Dynamic Source Protocol (DSR) cannot detect the malicious node, and the IDS package has overheads as well as more false alarms [54]. Hence, we need an alternative approach to detect the malicious node on the communication path with minimum overheads. The alternative approach includes trusting the next node in the path generated by DSR. Trust means transfer the packets above expected percentage (for example more than 95%). The trust level is calculated as the difference of packets received to transfer of packets by that node.

The trust depends upon the predictable behavior of nodes within communication distance with their continuous positive behavior. The trust is the degree of belief, which is based upon the continuous or repeated experience. Trust is non-transferable, reputation-based, time dependent, subjective, contextual, and unidirectional. Due to the nature of the trust, researchers are recently diverted towards these simple models (trust base models).

Since trust depends upon the closeness, the successive and neighboring nodes are included in the trust model. The successive node in the path is to communicate the data and the neighboring nodes (cooperation) are useful to confirm the trust factor, if the trust of successive node in the path is below the threshold (below the dependable value).

Trust-based packet transfer uses the Belief-Based Packet Transfer (BBPT) [7]. The BBPT uses the history of the other nodes transferring the data through its successive node. The BBPT requires the cooperation of its neighbors. The BBPT works better with agent-based systems, where

the agent collects the history of nodes, sets the neighborhood, and processes the data.

The rest of the paper introduces the related work, trust management, repeated games to model the trust level of successive nodes, and formulates the trust-based model in a cooperative environment. The paper further discusses the trust based packet forwarding, trust interaction with neighbor nodes, query-based trust calculation, conclusions, and the future research.

II. Related Work

Trust management is not a new concept in the electronic market. Reputation and trust are the basics of product sales. Establishing trust on a product manufacture industry and reputation of a product is the source of sales. Similarly, establishing trust on a node transferring the packets and reputation of the node is very important to keep the sensor node on data transfer path. In recent applications, trust calculation and update the node ratings uses reputation-based trust calculation [37], [40], event-based trust management [39], and agent-based trust management [30-32]. Further, repeated games help to detect the trustworthiness of a node in the path [37].

The sinkhole detection, selective forwarding attacks, acknowledgement spoofing, detection of malicious node, and utility-based decision making were discussed in [3], [5], [8], [9], [12], [13], [15], [16], [17], [21], [22]. None of these results attempted to verify that the next node in the path was malicious or trustworthy to transfer the data. Failure to transfer the packets depends upon the normal failure of a node (communication path or battery loss or complete node failure) or a node compromises. The research of selective forward attacks and detection of malicious nodes provides an extra effort if the data does not reach the destination. A trusted path is needed at the time of transferring the data (packets).

Perrig et al. [17] introduced the modified TESLA protocol [16] for sensor networks and named it μTESLA. The new protocol (μTESLA) is designed to show that security is possible in sensor networks by usage of a simple model to authenticate and transfer the data. Therefore, it is necessary to develop a simple model that eliminates unnecessary checks, avoids sinkholes, detect selective forward packet drops, and improve processing time. The Checkpoint-based Multi-hop Acknowledgement Scheme (CHEMAS) [22] identifies the localization of the suspected node that requires extra processing to detect a malicious node. The authors claim that the scheme (CHEMAS) has a high detection rate with communication overhead.

Isolating misbehavior and stabling trust routing in wireless sensor networks was studied in [21]. The trust routing algorithm uses the μTESLA scheme to form the chain of trust. The chain of trust is an expensive process and has more overheads compared to trusting the next successive node. However, it is difficult to keep track of the complete communication path particularly in WSN. The authors in [21] discussed various search methods to detect the insecure locations and isolate those locations from communication paths.

Zhang and Huang [24] used reinforcement learning to establish a secure path for packet transfer from source to the base-station. They concluded that adaptive spanning trees could maintain the best connectivity for transferring the packets between source and destination. The authors further discussed the energy-aware and congestion-aware problems for successful delivery of packets.

The trust management in wireless sensor networks was discussed by Carmen et al. [4]. A trust management system helps to detect the node (faulty or malicious) behaving in an unexpected way. Liu et al. [10] presented a dynamic trust model for ad-hoc networks, where each node is assigned a trust value according to its identity. Sometimes trust level is also calculated by evaluation of nodes over other nodes. Evaluation of trust factor is done with IDS data and statistical data of packet transfer rate. Rebahi et al. [19] discussed a reputation based trust mechanism in ad hoc networks where each node monitors the neighboring nodes activities, sends the information to the reputation manager, and stores it in a matrix for evaluation of nodes. Probst and Kasera [18] developed a distributed, statistical method for reputation-based trust in sensor networks. The method computes statistical trust based on sensor nodes behavior in terms of experiences in order to isolates faulty sensor nodes.

The belief-based packet-forwarding model in mobile networks using repeated games was discussed in [7]. The authors described the belief-based packet-forwarding model as being dependent upon history of other nodes' information transfer. The model further enforces cooperation in the ad hoc networks. The performance of packet transfer slightly degrades due to enforcing the cooperation of nodes compared to unconditionally cooperative outcomes. The model further provides the ad hoc networks and needs to modify for WSNs.

In this research, role of repeated games to detect the malicious or faulty node through a cooperative effort is discussed. The trust relation model and simulations were presented. Further, we discussed the trust model with reliable neighbors and query-based trust model.

III. Trust Management

Trust is used differently in different fields. A person is trustworthy, if he/she is dependable and reliable. That is, if a person completes the work on time, with satisfaction then we say the person is trust worthy. Trust depends upon the satisfaction of completing work repeatedly and as expected. The concept is used in credit cards, bank loans, and work places. Different procedures are used at different places. Sensor networks do not deviate much from the original concept.

The conceptual differences between trust, security, and reputation were explained in [29]. Further, the authors explained the WSNs security issues and innovative approaches. The authors suggested the future researchers may use these approaches to model the trust in respective fields. The suggestions conclude that the research needs to divert to create innovative approaches for trust-based WSNs.

Task-based trust management, event-based trust management and an agent-based trust management were studied in [30-34]. In [30], a general approach for task-based trust management is used similar to economics to detect the malicious node. The event-based approach [31] uses several trust ratings to enforce the security in WSN. The agent-based trust models in [31-34] discuss the attacks on WSN, packet dropping, and local storage management using the trust policy. The models can further discusses the trust aggregation, Hello flood attack, and detect the malicious nodes.

Hur et al. [35] presented a trust-based approach to distinguish illegal nodes from legal nodes. They claim that their approach detects insider attacks and uses trust evaluation model. The trust management model in [36] uses the Bayesian probabilistic approach. The model calculates the trust factor by using the current trust factor plus the second hand information received from its neighboring nodes.

Trust is a subjective term used for reliability of an entity. It is a subjective probability of an individual A that expects another individual B to perform a given task. The trust management model helps to detect the intruders (malicious nodes) and discard them from the communication path [4], [6], [14], [19]. The concept of reputation (collecting data about the status of a successive node) linked to the trustworthiness [2] of a person's example. In the current situation, trust depends upon the ratings of successive the node. If the ratings of the successive node are above the expected value (threshold) then the node will be trusted for transfer of data. Further, relaying on self-detecting misbehavior of nodes is dangerous. Therefore, collaborating between neighboring nodes is suggested.

The data transfer scenario from node A through the node D (Figure 1) establishes the trust of node D for future data transfers. For example, node A sends data to node D and node D receives the data and acknowledges to node A. There is no guarantee that node D transfers the data to the next successive node in the communication path. If the node A knows that node D transferred the data successfully, then the node A assumes that the node D can be trusted. After repeated transfers (successive node activity), if the trust factor reaches below the threshold, then node A compares the trust factors of its neighboring node B and the node C that are transferring their data through node D. If nodes B and C trust the node D, then node A establish a new route for successful transfer of data and avoids node D. Trust of the next successive node in data path is a kind of watchdog approach to detect the malicious node.

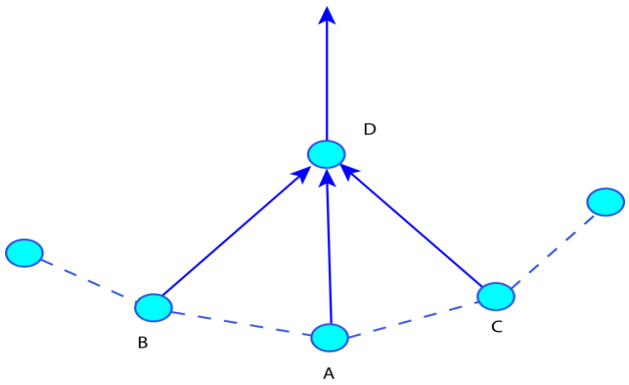


Figure 1: Scenario for node *A* establishing a trust of node *D*.

In the proposed approach, each node maintains a rating of its successive node (number of successful packet transfers) in the path. If the ratings of a node are above the threshold (expected minimum error rate), then the current node continues to transfer the packets. The current approach does not expect to calculate all ratings (packet transfer, noise, jamming, and infection factor) of its neighboring nodes and selects the path of highest ratings [17]. Selecting the highest rating path requires additional processing time and is a burden on the energy budget in the sensor node. The proposed approach detects the malicious node using the trust factor. For example, if node D selectively drops the packets from node A but not from nodes C and D then node A concludes that the path from node A through node D cannot be trusted. Since the communication path from node A to the node D is not trusted, node A establishes the alternative path. The alternate path is selected only if the successive node is not trusted.

IV. GAME MODEL

In games [11][23], the interaction between the players is inherently dynamic, so players always observe the actions of other players and decide their optimal response. Often, the game is played repeatedly to conclude the outcome. In repeated games, players have more opportunity to learn to coordinate their actions depending upon the previous outcome. In Figure 1, Player 1 and Player 2 (node A and node D) are involved in transferring the information where Player 1 transfers data to Player 2. Player 1 then waits for successful transfer of data packets from Player 2 to the next step in the path. Player 1's trust on Player 2 depends upon Player 2's successful transfer of

data packets. The problem is how these two players coordinate their actions.

The outcome of Player 1 depends upon the actions (repeated outcome conclusion) of Player 2. In the cooperative effort, we must consider the outcome of neighboring players (within communication distance) of Player 1; i.e., Player 3 and Player 4 (node B and node C in Figure 1) and have the similar interaction with Player 2. If Player 3 and Player 4 have same outcomes as Player 1 that is no better than Player 1, then the Player 1 concludes its decision to select communication path. If the trust of Player 1 on Player 2 depends upon the outcomes of its neighbor nodes and consistent, then we say it reaches Pareto optimality.

In repeated games, the behavior of Player 1 depends upon its opponent's (Player 2) actions (behavior). Further, no threat, punishment, or revenge is considered. The strategy is that Player 2 must transfer the packets received from Player 1. The trigger strategy is that the malicious behavior of Player 2 will permanently disconnect the path from Player 1 and its neighbors that have the current path through Player 2. For example, the stage game *G* is of the form

$$G = (N, A, U) \tag{1}$$

where N is a set of users (set of sensor nodes), A is a set of pure strategy profiles (action may be the missing packets for each transmission), and U is a vector of payoffs.

A simple stage game is defined with two players. If the two players n_1 and n_2, $(n_1, n_2 \in N)$, played with set of strategies $a_i, a_j (a_i, a_j \in A)$ in time unit t_i. In a repeated game, a player n_i plays with strategy a_i in time unit t_i to generate payoff u_i, $(u_i \in U)$. Let τ be the missing number of packets in a time period $T(t_1, t_2, \ldots, t_n)$. The number of missing packets in a unit time is τ / T.

The payoff β at node *D* in a period *T* is given by [7]

$$\beta = \frac{1 - \tau / T}{1 - (\tau / T)^{T+1}} \tag{2a}$$

Equation (2a) represents the normalized payoff. If $\beta > threshold$ then the player is trustworthy. Figure 2a is drawn for the dropping of packets at different time period. The Figure 2a shows that, the payoff is better if the packet dropping slot is in a larger time period. Consider an example with the threshold value is fixed at 95%. Figure 2a concludes that the larger time periods are suggested for better payoff (Figure 2a). The same may not be true in a smaller time period for the current random data. Therefore, the time period is very important to calculate the trust of a node. If we consider the smaller time periods, then the trust value must be kept at a lower rate.

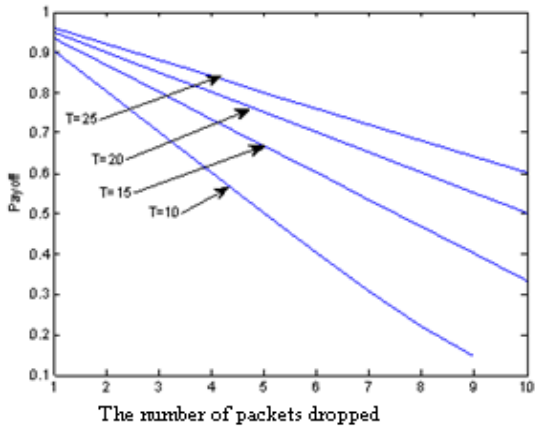


Figure 2a: Variation of time units and packet dropping

The payoff can also be calculated using a different method. If Ω is the common discount payoff and $g_i(a^t)$ is the per-period payoff of the i^{th} node related to current action a^t, then the normalized payoff β (relation to utility of sequence $(a^0, a^1, \ldots . a^T)$ at any node is given by [11]

$$\beta = \frac{1 - \Omega}{1 - \Omega^{T+1}} t \sum_{t=0}^{t=T} g_i(a^t) \tag{2b}$$

The trust of the player depends upon the outcome of β. Figure 2b is drawn using Equation 2b.

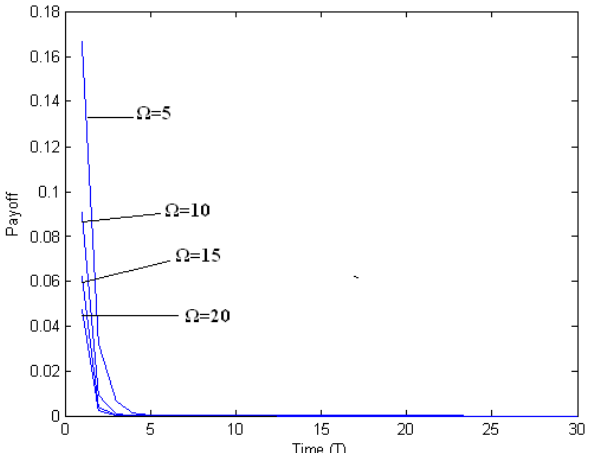


Figure 2b: Payoff β verses packet dropping in a given time period.

The Figure 2b shows that the payoff is higher with a lower number of packets dropped in the same time period. But the average payoff will be very close in a large time period. Therefore it is necessary to consider frequent averages for packet dropping for appropriate decision.

From Figures 2a and 2b, we conclude that larger periods must be considered to calculate the trust of a node. The smaller periods will panic the system, since small number of packet dropping will show the trust below the threshold.

V. TRUST MODEL AND GAME APPLICATION

Each node in the sensor network maintains a dynamic table to store the information about packet transfers of the successive node in the path. The values in the table include the packets transmitted from the node and packets transferred from the successive node (recorded through over hearing). These values are used for trust calculations of the successive node. The values are also used to calculate the risk involved in order to carry out packet transfer. In other words, trust value is a simple mathematical representation. The problem with no successive node will be dealt with different models [20].

Consider a sensor network of *N* nodes deployed in a field. Let the nodes be connected as shown in the Figure 3 and represented through a matrix of equation (3). The filled nodes are existing nodes and unfilled are drawn to complete the matrix. Unfilled means no node exists or a dead node. The Equation (3) helps to verify the isolated node (black-hole).

$$M = [M_{i,j}] = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (3)$$

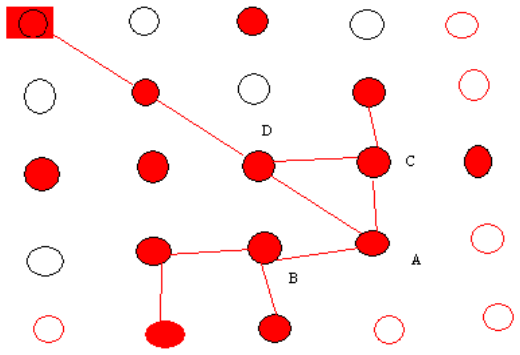


Figure 3: Sensor network nodes and their relation with neighboring nodes.

Reputation is used to predict the behavior of the node. We create a table at node *i* (values stored in table at node i are overhearing from node 2) to predict the behavior of the node j. Let $R_{i,j}$ represents the reputation of node j represented by node i. The reputation table RT_i stores the reputations maintained by node *i* and is represented as:

$$RT_i = \{R_{i,j}\} \quad (4)$$

The periodic quantification of reputations at node *j* is $Q_{i,j}$ and is stored at RT_i as part of node *j*. The missing is calculated as $(1-Q_{i,j})$. Further, each node has direct and indirect observations of reputations. Direct observation is the reputations stored at node i and indirect observations are received from neighboring nodes. The indirect observations are represented as $IQ_{i,j}$. The trust prediction of the node j depends upon $Q_{i,j}$ and $IQ_{i,j}$.

In repeated games, expected payoff depends upon the action profile and its observation. The action profile is given by

$$U_i = (\frac{1}{Q_{i,j}})\lambda \quad (5)$$

where λ is the difference between $Q_{i,j}$ and $IQ_{i,j}$. If $\lambda = 0$ then the packets transferred at a node and its neighboring node are the same. The trust of the node depends upon the factor β. Further we calculate the average discount factor in order to calculate the stable state of the node. The average discount payoff is given by

$$UA_i = \frac{\beta \sum_{t=1,n} \Omega_i(t) U_i(t)}{n} . \quad (6)$$

If the average discount payoff is above the threshold then node is trustworthy. If the trust state is consistent, then we say it reaches Nash equilibrium. If the Nash equilibrium exists in repeated games, then it satisfies the Folk theorem [1] and Pareto optimality (payoff in Nash equilibrium). The simulations for average discount payoff are shown in Figure 4a and Figure 4b.

Figure 4a shows the number of packets transmitted to average discount payoff. The system stabilizes after transmission reaches 1500 and above. The trust calculation in large time periods and packets transfer provides the stable results. In Figure 4b, average discount payoff is better in larger period of time.

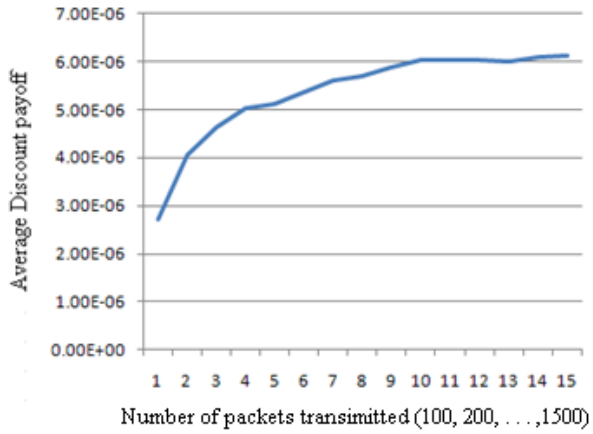


Figure 4a: Average discount payoff verses number of packets dropped

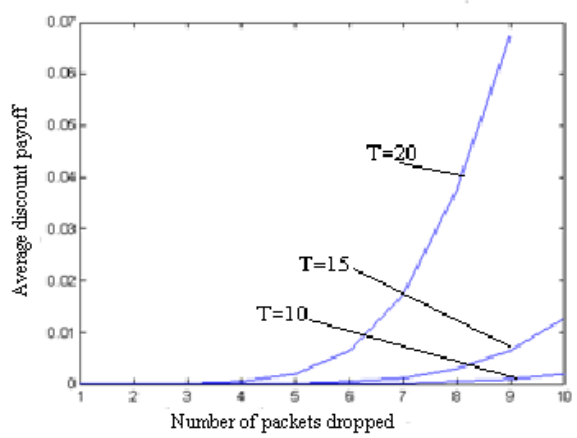


Figure 4b: Average discount payoff verses number of packets dropped

For a small value of λ (0.001) and probability of more than 90% successful packet transfer rate, the payoff increases in a smaller period of time (if lower number of packets is dropped). In average discount payoff, the number of packets dropped is set approximately the same. The number of transmitted packets is numbered in small or many. In the beginning, the average discount payof increases (from 100 packet transmission to 900 packet transmission) and settles after it reaches a transmission rate of 1000 packets with the same number of drops. This shows, for a selected action strategy of a player, the game reaches Nash equilibrium at action profile during the time period of higher number of packet transmission with lower dropouts. That means the successive node can be trusted at current state.

VI. TRUST-BASED PACKET FORWARDING

In trust-based systems, we begin to believe all nodes in the path are trusted. Trust of node 2 at node 1 will be developed after repeated transfer of packets from node 1 (n_i) to node 2 (n_j) and then successfully transferred from node 2. The trust of interaction between these nodes is

$$T_{i,j}^{t} = (n_j, s_k, TE_{i,j,t}) \tag{7}$$

where $T_{i,j}^{t}$ is a trust of node n_i on node n_j at time t, s_k is a set of possible specifications to perform task at n_j where $s_k \in S$, and $TE_{i,j,t}$ is the set of tasks.

Further, the node n_i, the initiator node must store the data about the reliability of node n_j when the packets are transferred repeatedly. The node n_i experience in repeated operation of packet transfer is

$$R_{i,j}^{t} = (n_j, s_k, P_{i,j,t}) \tag{8}$$

where $P_{i,j,t}$ is satisfaction achieved by node n_i at node n_j at any time t and $P_{i,j,t} \in (0,1)$.

The experience of each particular task will be updated at n_i and represented as

$$I^{t}(n_j, s_k) = (n_j, w_j) \tag{9}$$

where w_j is the response from n_j in the interaction. By updating the process combinations of I^t and storing the experiences of T^t and R^t we get the quality satisfaction measurements.

The equations (2), (6), and (9) will provide the needed information to trust the node n_i for future transformation of information.

To create trust level we generated random data to test the equation (9). In the test process, 100 random samples were generated for node n_j. If node n_j is trusted more than 90%, we note that the trust level is above threshold. This process was repeated 100 times to reach correct trust level. The process was repeated and the percentage of trust in hundred attempts is shown in Figure 5.

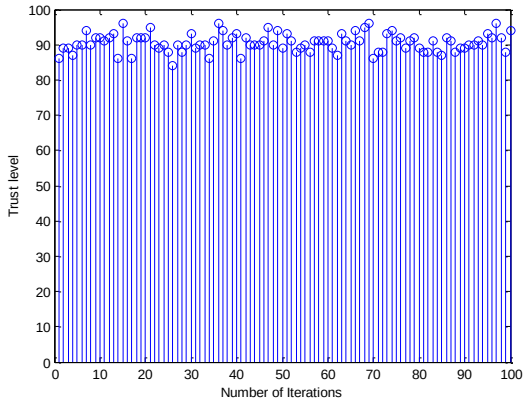


Figure 5: Trust relation generated in 100 iterations.

The random generation of trust data is not a correct process but it helps in simulations. The average trust of a hundred samples in Figure 5 is approximately 90.42. The average of hundred samples is approximately 90.42. The threshold was set as 90 and above and satisfies the simulation results. Therefore, we can assume that if the transfer rate is above 90% the node can be trusted.

VII. Trust Model With Reliable Neighbors

The nodes within the communication distance are the neighbors of the node. The neighbor nodes confirm the trust of common successive node. For example, node B and node C are the neighbors of node A (Figure 1). The neighbors of the node n_i can be represented as:

$$N_i = (n_j \mid n_k \in N), \; if \; (n_i, n_k) = true \tag{10}$$

To confirm the neighbor nodes, we use the Boolean function in equation (10). If the Boolean function value is true in equation (10), the nodes are neighbors. Identify trusted neighbors and keeps the superior nodes (trustable nodes) and ignores the inferior nodes, the node A interacts with several of its neighbors (node B and node C). For example, if we denote ζ_i as the inferior neighbor node and ζ_s as the superior neighbor node then their values will be represented as $0 \le \zeta_i \le \zeta_s \le 1$. If ζ_s is close to 1 then the neighbor will be identified as superior. Therefore, the most trusted node is

Therefore, the representation of most trusted node is

$$NT^t_{\sup}(n_i, s_k) = \{n_k \mid n_k \in N\}, \; if \; trust \; of \; n_k \ge threshold \tag{11}$$

Similarly, the set of nodes with doubtful confidence is given by

$$NT^t_{\inf}(n_i, s_k) = \{n_k \mid n_k \in N\}, \; if \; trust \; of \; n_k < threshold \tag{12}$$

The most reputed nodes (established complete trust over time) will be grouped into reliable nodes and represented as

$$NR^t_{\inf}(n_i, s_k) = \{n_k \mid n_k \in N\}, \; if \; trust \; of \; n_k < threshold \tag{13}$$

The reliable nodes are useful to verify the trust of successive nodes. If the reliable node is not available, it will verify trust of a successive node based on reputation values or node ratings (economic market place) done using economics models [39, 27].

The calculation of the threshold value is very important and will be calculated using equation (8). The agent updates the threshold value in preset time instances.

VIII. Query-based Direct Trust Calculation

The query-based approach is useful to establish the communication path from source nodes to the base station. The query system helps to infer the future status of the trusted communication path. Further, the information obtained through query system will predict the future actions of the nodes in the path.

The performance of node n_j over the (change of) time t depends upon the successful transfer of packets that were received from the node n_i. The reliability of a node n_j is the trust measure associated with the task (packet transfer). Sabater and Sierra [25] stated that the outcome of the reputation measure of node n_i depends upon delivery time, quality, and percent of transfers. Using these factors the trust (T) of the node n_i to the node n_j is

$$T = f(O, \pi, t, R) \tag{14}$$

where O is the outcome, π is variable outcome to be judged, t is recorded time, and R is rating $R \in \{-1, 1\}$. The value -1 is absolutely negative and 1 is positive. Since R is the ratings at time t of outcome π of a task, the equation (14) must satisfy the equation (7). The reliability value is obtained from the number of experiences used to calculate the trust and variability of these ratings experiences.

Using repeated game model, the outcome of node n_j with imperfect history of packet forwarding and dropping is calculated as

$$U_j(\delta) = (1-\delta)\sum_{t=0}^{n} \delta^t u_j^t(a_j^t) \quad (15)$$

where the discount factor $\delta \in (0,1)$, a_j^t is the action part of j^{th} node at time *t*, and u_j^t is expected payoff profile. Folk's theorem for repeated games [1] asserts that there exists $\bar{\delta}$ such that $0 < \bar{\delta} < 1$ will be enforced based on the information shared by the players. Therefore, we rewrite the equation (15) using the Fork's theorem as:

$$U_j(\bar{\delta}) = (1-\bar{\delta})\sum_{t=0}^{n} \bar{\delta}^t u_j^t(a_j^t) \quad (16)$$

The Folk's theorem further assumes that the players share the common information about each other's actions. The strategy can be extended to inference of the other player's future actions. That is, depending upon the current information of successive player, the current player can infer the next (future) actions of successive player. Using this information, the player can decide to recalculate the communication path.

The player n_i shares the common information from other players (Folk's theorem) and the rating of the node n_j will be calculated using Automatic Collaborating Filtering (ACF) [26]. The ACF uses the mean squared difference formula [26] with two users. Let the performance of node n_j is rated by nodes *G* and *H*. Let G_f and H_f denote the ratings of *G* and *H* on a feature (packet transfer) *f* of the node n_j. Let χ be the set of features of the node n_j. Both *G* and *H* are rated the node n_j and $f \in \chi$. The difference between two nodes *G* and *H* in terms of their interests in a node n_j is given by [9]:

$$\Delta = \delta_{U,j} = \frac{1}{|\chi|}\sum_{f \in S}\left(G_f - H_f\right)^2 \quad (16)$$

If Δ is very small, the ratings provided by neighboring nodes are helpful for decision. Otherwise, the node n_i need to collect more facts from other neighbors before any further decision to be made.

There are two types of ACF recommendations: invasive and noninvasive based on the user preferences [27], [28]. The invasive approach uses explicit user feedback having the preferences between 0 and 1. The preferences are interactive and Boolean in noninvasive approach. In the noninvasive approach, the rating 0 means the user not rated and the rating 1 means the user rated. Therefore in noninvasive cases, it requires more data for any decision. In ACF systems, all user recommendations will be taken into account even though they are entered at different times. The ACF system gets more strength with more recommendations and new recommendations depend upon the current data updates in the system.

IX. Conclusion and Future Research

The available security models for packet transfer in wireless networks are useful for intruder detection, sinkholes, and black holes. These methods need a lot of processing, storage, and energy. There is no literature available for a simple security model for wireless sensor networks that confirm the trusted successive node to transfer the packets. The proposed model is a unique approach to transfer the data securely and at the same time confirms the trust of next level node.

The paper discusses the trust models and trust-based approach in sensor net works. The role of repeated game in trust models was introduced and calculated the average discount payoff verses number of packets dropped. The model identifies that large time slots provide better results than observing the packet dropping in a short period of time.

Further, the model for trust relation among the nodes was presented and prediction of a trusted node in the path was discussed using game model and Automatic collaborative filtering approach. The models presented are useful to transfer the data with minimum overheads.

The future research includes the rating of a successive node using electronic marketplace model [39] to calculate the trusted path. Further, the trusted successive node will be calculated using an agent with a set of nodes (cluster). The cluster-based approach saves the energy at the node level, since calculations are done at agent node. Further, an event-based [38] approach with electronic marketplace concept can be developed depending upon the situation of sensor networks. The mixed approaches are suggested depending upon the topology of sensor networks and type of environment.

Acknowledgement

The research work was supported by the ONR with award No. N00014-08-1-0856. The first author wishes to express appreciation to Dr. Connie Walton, Grambling State University and Dr. S. S. Iyengar, LSU Baton Rouge for their continuous support.

References

[1] Y. B. Reddy and Rastko Selmic., "Secure Packet Transfer in Wireless Sensor Networks – A Trust-based Approach", IARIA- ICN 2011, January 23-28, 2011 - St. Maarten.

[2] Audun, J., Ismail, R., and Boyd, C., "A survey of Trust and Reputation Systems for Online Service Provision", *Decision Support Systems*, 2006.

[3] Byers, J., and Nasser, G., "Utility-based decision-making in wireless sensor networks", *Proc. of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing*, November 2000, Boston, Massachusetts.

[4] Fernandez-Gago, M., Roman, R., Lopaz, J., "A Survey on the Applicability of Trust Management Systems for Wireless Sensor Networks", *3rd International Workshop on Security, Privacy, and Trust in Parvasive and Ubiquitous Computing*, July 2007.

[5] Garth, V. C. and Niki, P., "Evolution of Cooperation in Multi-Class Wireless Sensor Networks", *LCN* 2007.

[6] Hur, J., Lee, Y., Hong, S., and Yoon, H., "Trust-based secure aggregation in Wireless Sensor Networks", *Sensor and Ad Hoc Communications and Networks (SECON '06)*, 2006.

[7] Ji, Z., Yu, W., and Liu, K. J., "Belief-based Packet Forwarding in Self-organized Mobile Ad Hoc Networks with Noise and Imperfect Observation", *IEEE WCNC* 2006.

[8] Kannan, R. and Iyengar, S.S., "Game-theoretic models for reliable path-length and energy-constrained routing with data aggregation in wireless sensor networks", *IEEE J. of Selected Areas in Communications*, Aug 2004.

[9] Kanno, J., Buchart, J. G., Selmic, R. R., and Phoha, V., "Detecting coverage holes in wireless sensor networks," *17th Mediterranean Conference on Control and Automation*, June, 2009.

[10] Liu, Z., Joy, A., and Thomson, R., "A Dynamic Trust Model for Mobile Ad Hoc Networks", *IEEE International workshop on Future Trends of Distributed Computing Systems (FTDCS)*, 2004.

[11] Machado, R. and Tekinay, S., "A survey of game-theoretic approaches in wireless sensor networks", *Computer Networks*, Nov. 2008.

[12] Mark, F., Jean-Pierre, H., and Levente, B., "Cooperative Packet Forwarding in Multi-Domain Sensor Networks", *PERCOM* 2005.

[13] Miler, D., Tilak, S., Fountain, T., "Token equilibria in sensor networks with multiple sponsors", *CollaborateCom* 2005.

[14] Momani, M., and Challa, S., "Trust management in Wireless Sensor Networks", *5th IEEE/ACM International Conference on Hardware/Software Codes and System Synthesis*, 2007.

[15] Narayanan, S., Mitali S., and Bhaskar K., "Decentralized utility-based sensor network design", *Mobile Networks and Applications*, June 2006.

[16] Perrig, A., Canetti, R., Tygar, J. D., and Song, D., "Efficient authentication and signing of multicast streams over lossy channels", *IEEE Symposium on Security and Privacy*, May 2000.

[17] Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J. D., "SPINS: Security Protocols for Sensor Networks", *MOBICOM* 2001, Rome, Italy, June 2001.

[18] Probst, M.J., and Kasera, S.K., "Statistical trust establishment in wireless sensor networks," *Proc. ICPADS '07 the 13th International Conference on Parallel and Distributed Systems*, 2007.

[19] Rebahi, Y., Mujica, V., and Sisalem, D., "A Reputation-Based Trust Mechanism for Ad Hoc Networks", *the 10th IEEE Symposium on Computers and Communications (ISCC'05)*, 2005.

[20] Reddy, Y. B., "Potential Game Model to Detect Holes in Sensor Networks", *IFIP/NTMS*, 2009.

[21] Tanachaiwiwat, S., Dave, P., Bhindwale, R., Helmy, A., "Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks", *IEEE IPCC*, October 2004.

[22] Xiao, B., Yu, B., Gao, C., "CHEMAS: Identify Suspect Nodes in Selective Forwarding Attacks", Journal of Parallel Distributed Computing, vol. 67, 2007.

[23] Yuan, J. and Yu, W., "Distributed cross-layer optimization of wireless sensor networks: a game theoretic approach", Proc. of IEEE Global Telecommunications Conference, 2006.

[24] Zhang, Y., and Huang, Q., "A Learning-based Adaptive Routing Tree for Wireless Sensor Networks", *J. of Communications*, 1 (2), 2006..

[25] Sabater, J., and Sierra, C., REGRET: A Reputation Model for Gregarious Societies, *First Int. conf. on Autonomous Agents and Multi-agent Systems*, 2002.

[26] Cunningham, P., Intelligent Support for E-commerce, http://www.cs.tcd.ie/Padraig.Cunningham/iccbr99-ec.pdf, 1999

[27] Hays, C., Cunningham, P., and Smyth, B., A Case-based Reasoning View of Automated Collaborative Filtering, *4th International Conference on Case-Based Reasoning*, 2001.

[28] Sollenborn, M., and Funk, P., Category-Based Filtering and User Stereotype Cases to Reduce the Latency Problem in Recommender Systems, 6th European Conference on Case Based Reasoning, *Springer Lecture Notes*, 2002.

[29] Momani, M., and Challa, S., "Survey of trust models in different network domains", *Int. J. of Ad hoc sensor & Ubiquitous Computing (IJASUC)*, vol. 1, no. 3, September 2010.

[30] Chen, H., Wu, H., Hu, J., and Gao, C., "Agent-based Trust Management Model for Wireless Sensor Networks", *International Conference on Multimedia and Ubiquitous Engineering*, 2008.

[31] Chen, H., Wu, H., Hu, J., and Gao, C., "Agent-based Trust Model in Wireless Sensor Networks," *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007.

[32] Boukerche, A., and Li, X., "An Agent-based Trust and Reputation Management Scheme for Wireless Sensor Networks", *IEEE GLOBECOMM, 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp 66-77, October 2004.

[33] Marmol, F. G., and Perez, G. M., "Providing Trust in Wireless Sensor Networks using a Bio-Inspired Technique", *NAEC 2008*.

[34] Haiguang Chen, Huafeng Wu, Xi Zhou, Chuanshan Gao, "Agent-based Trust Model in Wireless Sensor Networks", *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007.

[35] Hur, J., Lee, Y., Hong, S., and Yoon, H., "Trust-based Secure Aggregation in Wireless Sensor Networks", SECON 2006.

[36] Momani, M., and Challa, S., "Trust Management in Wireless Sensor Networks", *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007*.

[37] Y. B. Reddy and Rastko Selmic., "Trust-based Packet Transfer in Wireless Sensor Networks", *Communications and Information Security (CIS2010), IASTED*, Nov 8-10, 2010,USA.

[38] Chen, H., Wu, H., Hu, J., and Gao, C., "Event-based Trust Framework Model in Wireless Sensor Networks", *International Conference on Networking, Architecture, and Storage*, 2008.

[39] Zacharia, G., Moukas, A, and Mae, P., "Collaborative Reputation Mechanisms for Electronic Marketplaces", *Decision Support Systems*, vol. 29, no. 4, December 2000.

[40] Ganeriwal, S., and Srivastava, M. B., "Reputation-based Framework for High Integrity Sensor Networks", *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, 2004.

[41] Abreu, D., Dutta, P., and Smith, L., "The Folk Theorem for Repeated Games: A NEU Condition", *Econometrica*, vol. 62, 1996.

[42] Akyildiz, I.F., Weilian Su, Sankarasubramaniam, Y., and Cayirci, E., "A survey on sensor networks", *IEEE Communications Magazine*, Aug. 2002.

[43] Rentala, P., Musunnuri, R., Gandham, S., and Saxena, U., "Survey on Sensor Networks", *Technical report*, University of Texas at Dallas, 2000.

[44] Papageorgiou, P., "Literature Survey on Wireless Sensor Networks", *Technical Report*, University of Texas, Dallas, July 16, 2003.

[45] Bharathidasan, A., and Ponduru, V., "Sensor Networks: An Overview", *Technical report, University of California*, Davis, 2000

[46] Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W., "A Taxonomy of Wireless Micro-Sensor Network Models", *ACM SIGMOBILE Mobile Computing and Communications Review*, April 2002.

[47] Goldsmith, A.J., and Wicker, S. B., "Design challenges for energy-constrained ad hoc wireless networks", *IEEE Wireless Communications*, Aug. 2002.

[48] Stark, W., Hua Wang, Worthen, A., Lafortune, S., and Teneketzis, D., "Low-energy wireless communication network design", *IEEE Wireless Communications*, Aug. 2002.

[49] Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W., "Infrastructure tradeoffs for sensor networks", *ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[50] Chien-Chung Shen, Srisathapornphat, C., and Jaikaeo, C., "Sensor information networking architecture and applications", *IEEE Personal Communications*, Aug. 2001.

[51] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K., "System architecture directions for networked sensors", *ACM ASPLOS*, 2000.

[52] Da Silva Jr, J. L., Shamberger, J., Ammer, M.J., Guo, C., Li, S., Shah, R., Tuan, T., Sheets, M., Rabaey,J. M., Nikolic, B., Sangiovanni-Vincentelli, A., and Wright, P., "Design methodology for PicoRadio networks", *Proceedings of Design, Automation and Test in Europe*, 2001.

[53] Shenker, S., "Fundamental Design Issues for the Future Internet", *IEEE Journal on Selected Areas in Communications*, Sep. 1995.

[54] Reddy, Y. B., Durand, J., and Sanjeev Kafle, S., "Detection of Packet Dropping in Wireless Sensor Networks", *7th International Conference on Information Technology: New Generations*, 2010.

Advancement Towards Secure Authentication in the Session Initiation Protocol

Lars Strand

Norwegian Computing Center / University of Oslo
Oslo, Norway
Email: lars.strand@nr.no

Wolfgang Leister

Norwegian Computing Center
Oslo, Norway
Email: wolfgang.leister@nr.no

***Abstract*—The Digest Access Authentication method used in the voice over IP signaling protocol, SIP, is weak. This authentication method is the only method with mandatory support and widespread adoption in the industry. At the same time, this authentication method is vulnerable to a serious real-world attack. This poses a threat to VoIP industry installations and solutions. In this paper, we propose a solution that counters attacks on this wide-spread authentication method. We also propose a two-step migration towards a stronger authentication in SIP. We add support for a Password Authenticated Key Exchange algorithm that can function as a drop-in replacement for the widely adopted Digest Access Authentication mechanism. This new authentication mechanism adds support for mutual authentication, is considered stronger and can rely on the same shared password used by the digest authentication. A long-term solution is to replace the authentication scheme in SIP with a security abstraction layer. Two such security frameworks are introduced, discussed and evaluated: the Generic Security Services Application Program Interface and the Simple Authentication and Security Layer, which both enable SIP to transparently support and use more secure authentication methods in a unified and generic way.**

***Index Terms*—SIP, authentication, Digest Access Authentication, PAKE, SASL.**

I. Introduction

Considering the growing market share for Voice over IP (VoIP) technologies, VoIP services need to be stable and secure for the benefit of both users and service providers. Authentication methods are an important part of this and need to be thoroughly examined. We base our current work on a conference article [1], where we analyzed and implemented an attack on the Digest Access Authentication used in the Session Initiation Protocol (SIP) and proposed a correction to mitigate this attack. Since there is a need for better authentication methods in SIP, we add support for a security abstraction layer in SIP [2] and propose a migration strategy towards a secure authentication in SIP [3].

The importance of analyzing and improving the SIP authentication methods comes from the fact that there has been a steady increase in the number of VoIP users since 2002, as well as a decrease in the number of PSTN installations [4]. With two billion users worldwide having access to the Internet by the end of 2010 [5], the VoIP growth potential is huge. For example, at the end of 2009, 29.1 % of the private land-line phone market in Norway used VoIP.

VoIP is the emerging technology that will eventually take over from the traditional Public Switched Telephone Network (PSTN) [6] due to VoIP's improved flexibility and functionality, such as improved sound quality ("HD sound") using wideband codecs like G.722 [7], instant messaging (IM), presence, mobility support, and secure calls. VoIP also reduces maintenance and administration costs since it brings convergence to voice, video and data traffic over the IP infrastructure.

Although there exist several competing network protocols that are capable of delivering VoIP, the Session Initiation Protocol (SIP) [8] and the Real-time Transport Protocol (RTP) [9] developed by the IETF have become the *de facto* industry standard. These two protocols fulfill two different functions – SIP is used for signaling, e.g., responsible for setting up, modifying and tearing down multimedia sessions, while RTP transports the actual media stream (voice). Although the SIP protocol is flexible and rich in functionality [10], several vulnerabilities and security attacks have been found [11]–[13].

Securing a SIP-based VoIP system has proven challenging and the reasons are multi-faceted:

- The scale and complexity of the SIP protocol specification, with primary focus on functionality rather than a sound security design [14].
- SIP usage of intermediaries, expected communication between nodes with no trust at all, and its user-to-user operation make security far from trivial [8, page 232].
- A large number of threats against VoIP systems have been identified [15]. Several security mechanisms for countermeasures have been proposed, but no single security mechanism is suited to address all these security threats concerning VoIP and SIP [16], [17].
- Since the SIP and RTP protocols share the same infrastructure as traditional data networks, they also inherit the security problems of data communication.
- VoIP services have strict requirements to the network performance with respect to Quality of Service since it is a duplex communication with low tolerance for latency, packet loss and saturation. Introducing strong security mechanisms might affect network performance [18].

PSTN is a mature and stable technology providing 99.999% uptime [19], and users will expect VoIP to perform at similar

service level. But with an increasing number of VoIP users, VoIP will become a target for attackers looking for financial gain or mischief. A clear threat taxonomy is given by the "VoIP Security Alliance" [15] and is discussed by Keromytis [20].

In VoIP, authentication tries to validate the identity of the communication peers and to bind that identity to a subject (peer). It must be stressed that the user's phone is authenticated rather than the user herself. In VoIP terminology, a subject could be a User Agent (UA), such as a phone, identified by a phone-number/username and IP-address/hostname pair, denoted as an Address-of-Record (AoR). The authentication in VoIP is therefore the assurance that a communicating entity, the UA, is the one that it claims to be [21]. Equally important for the UA is to establish the identity of the communicating peer, i.e., the SIP server. If the client does not authenticate the SIP server, it might risk to communicate and send content to a hostile SIP server.

SIP supports several security services, and the RFC specification documents recommends their use. These security services can provide protection for authentication, confidentiality, and more. Yet, only one such security service is mandatory: the SIP Digest Access Authentication (DAA) method [8, page 193]. In the EUX2010sec research project [22], we revealed, in close collaboration with our project partners, that most VoIP installations only use the mandatory, Digest Access Authentication (DAA) method [23]. DAA is primarily based on the HTTP Digest Access Authentication [24], and is considered to be weak and vulnerable to serious real-world attacks [25].

One contribution of this paper is to present and analyze the seriousness of a vulnerability we presented in our earlier work – the registration attack [25]. We implement a real-world attack, and propose a solution to the DAA that will counter this vulnerability. Further, we introduce an authentication method based on the Password Authenticated Key Exchange (PAKE) [26], which provides mutual authentication based on a shared secret, and can function as a drop-in replacement of the digest authentication currently used. However, a more flexible authentication method is desired. Different security requirements may require different authentication mechanisms. Instead of adding support for many different authentication mechanisms in SIP, we introduce support for a security abstraction layer. Two such security frameworks are introduced, discussed and evaluated. The Generic Security Services Application Program Interface (GSS-API) [27] and Simple Authentication and Security Layer (SASL) [28], which both enables SIP to transparently support and use more secure authentication methods in a unified and generic way.

The rest of the paper is organized as follows: Related work and the current state of authentication in SIP is given in Section II, and show our method in Section III. We explain and implement the registration attack, and propose a solution on how to counter the attack in Section IV. In Section V we show how a modified PAKE can be used to add mutual authentication in SIP. Support for the security abstraction layers GSS-API and SASL is added, discussed and evaluated in Section VI. We present the conclusion and future work in Section VIII.

II. State of knowledge

The DAA is currently the most common authentication mechanism for SIP. DAA is simple, but rather insecure. It is the only authentication mechanism which support in SIP is mandatory [8, Section 22]. DAA uses the MD5 hash function and a challenge-response pattern, and relies on a shared secret between client and server within a SIP domain [24]. DAA is performed during the SIP `REGISTER` handshake between the UA and the SIP server, as depicted in messages 1-3 and 6 in Fig. 10. The UA receives a nonce value from the SIP server, computes a digest hash value over the nonce, the shared secret and some other SIP header values, and send it to the SIP server. The SIP server computes the same digest hash. If both digests are identical, the UA is authenticated. The DAA is weak and vulnerable to a serious real-world attack, as described in Section IV-A.

Based on the DAA, Undery [29] proposed a more flexible use of variables protected by the digest. His paper addresses the shortcomings of DAA and suggests to allow the server to decide which headers it requires to be included and protected by the digest computation. Unfortunately, his approach does not require specific headers fields to be included. His approach is therefore vulnerable to the same vulnerability presented and implemented in this paper.

Yang et al. [30] also conclude that DAA is weak. They argue that, since DAA is vulnerable to an off-line password guessing attacks, a more secure authentication method is required. They propose an authentication method based on Diffie-Hellman. Unfortunate, they do not discuss nor add any additional SIP headers in their new authentication scheme. Therefore, their solution is also vulnerable to the registration attack implemented in this paper.

Secure MIME (S/MIME) [31] is an authentication mechanism presented in the SIP core specification document RFC3261 [8]. S/MIME intends to achieve end-to-end authentication between UAs. The entire SIP message is encapsulated in a specific SIP message using MIME, which is signed and optionally encrypted. The receiving UA checks whether the sending UA's certificate is signed by a trusted authority. Since S/MIME depend on end-user certificates, the UAs must support multiple root certificates since no consolidated certificate authority exists. Additionally, certificate handling issues, such as revocation and renewal, complicate the use of certificates. There has been rather limited industry support for S/MIME.

Transport Layer Security (TLS) [32] support for SIP, called "Secure SIP" and denoted "SIPS", has gained some industry momentum. TLS is designed to make use of TCP to provide a protected end-to-end communication between two endpoints. The application data, here SIP, are encrypted and integrity-protected. The communicating endpoints authenticate using digital certificate, usually X.509 certificates, and thus require a public key infrastructure (PKI). TLS does not offer end-to-end confidentiality and integrity protection of SIP messages,

since the TLS connection must be terminated and initiated for each hop between intermediate SIP servers. The use of TLS also restricts SIP to use TCP as transport protocol. By using TLS, SIP relies on a lower communication layer protocol to enforce security mechanisms.

Two other authentication methods have emerged within the Internet Engineering Task Force (IETF):

1) The *P-Asserted Identity* [33] is intended to work within a trusted environment. An unprotected SIP header is appended by the UAs SIP server that informs the receiving SIP server that the identity of the UA has been checked and thus can be trusted. However, since the SIP header is sent in clear rather than protected by cryptography methods, it can easily be removed by an attacker without any of the communicating peers noticing this.
2) The *SIP Strong Identity* [34] introduces a new SIP service, the "authentication service", which signs a hash over selected SIP header values, and includes the signature as a SIP header along with a URI that points to the sender's certificate. The receiver computes the same hash and compares the results. However, using this method, only the client is authenticated and an attacker can remove these headers without implications.

Note that both "P-Asserted Identity" and "SIP Strong Identity" rely on a successful DAA authentication to be applicable. These are also applied by the SIP servers rather than the clients themselves, and are thus only providing indirect authentication of the client since the server is authenticating on behalf of the client. None of these authentication methods have seen any widespread deployment yet [14].

Palmieri et al. [35], [36], dismiss DAA as a usable authentication method, and instead craft a new authentication schema with digital signatures based on public-key encryption. But since they rely on certificates, their solution suffers under similar certification handling issues as S/MIME and TLS. They also admit that relying on PKI is both difficult and costly to implement. Liao et at. [37], propose an improved authentication in SIP with self-signed public keys on elliptic curves. However, Liao's proposal uses smart-cards to store authentication data and rely on a trusted third party [38].

The H.323 recommendation for the VoIP protocol from the International Telecommunication Union (ITU) has failed to see widespread adoption by industry players, and is considered abandoned in favor of SIP/RTP [10]. The authentication methods in H.323, specified in H.235 [39], [40] uses well established security mechanism, like certificates, and Diffie-Hellman key exchange, to enforce authentication. Further analysis is needed to see whether the H.235 standard protects the signaling better than SIP.

The Inter-Asterisk eXchange (IAX) [41], also published by the IETF, establishes a competing protocol to SIP/RTP. IAX has several security properties that are better than SIP. By multiplexing channels over the same link and transporting both signaling and media over the same port, enforcing security mechanisms is easier. IAX supports two authentication methods: *1*) MD5 Message Digest authentication [42] computed

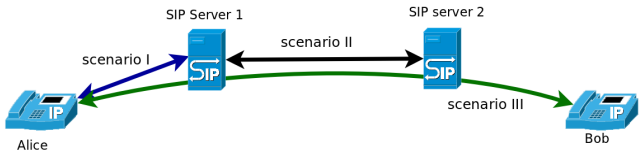


Fig. 1: Three different usage scenarios where authentication in SIP is desired.

over a pre-shared secret and a challenge (nonce), or *2*) using RSA public-key encryption on the challenge. In both methods, the nonce value is the only protocol parameter that is integrity protected by the authentication. Future work needs to investigate whether the IAX authentication method is adequately secure.

The SIP protocol needs an authentication mechanism that avoids the security vulnerabilities the currently used DAA has. A replacement authentication mechanism should preferably not rely on PKI, have support for strong mutual authentication, and support all three scenarios listed in the upcoming Section III.

III. Method and Case Study

In Norway, both private companies and public authorities are migrating from PSTN to VoIP [23]. To create suitable scenarios we study the VoIP installation of three companies in Norway; one medium sized company with 150 employees, and two larger companies with 3000 and 4700 employees. We have gathered several of these VoIP configurations and setups, and replicated the installations in our test lab [43]. In these companies, most of the employees have their own VoIP phone, called a User Agent (UA). All VoIP servers run the Linux operating system with the open source telephony platform *Asterisk* [44]. We found in these configurations that the digest authentication is the only authentication method for the UAs.

In the following paragraphs, the numbers in parentheses refer to the numbers in Fig. 2, where the workflow in our method is shown.

In order to gain knowledge of the SIP protocol we use the specification documents (1), here the SIP standard. Then, we analyze VoIP network traffic going through the test lab (5). We have implemented two VoIP setups based on configurations from our industry partners ((2) and (3)). The network traffic is intercepted and saved to file using the network tool *tcpdump* (4). The network traffic is then analyzed off-line using the packet analyzer, *Wireshark* (5). An example of such an analysis is shown in Fig. 3.

As an additional input we consider threats given by [15] and given in earlier work, such as a SIP attack analyzed by Hagalisletto and Strand [25], using the protocol analyzer PROSA (6). We explain this attack in more detail in Section IV-A, and implement and execute the attack using the network tool *NetSED* (7) as shown in Fig. 7. Based on the security requirements (9) obtained from the SIP specification, we then checked if the authentication method (10) was compromised

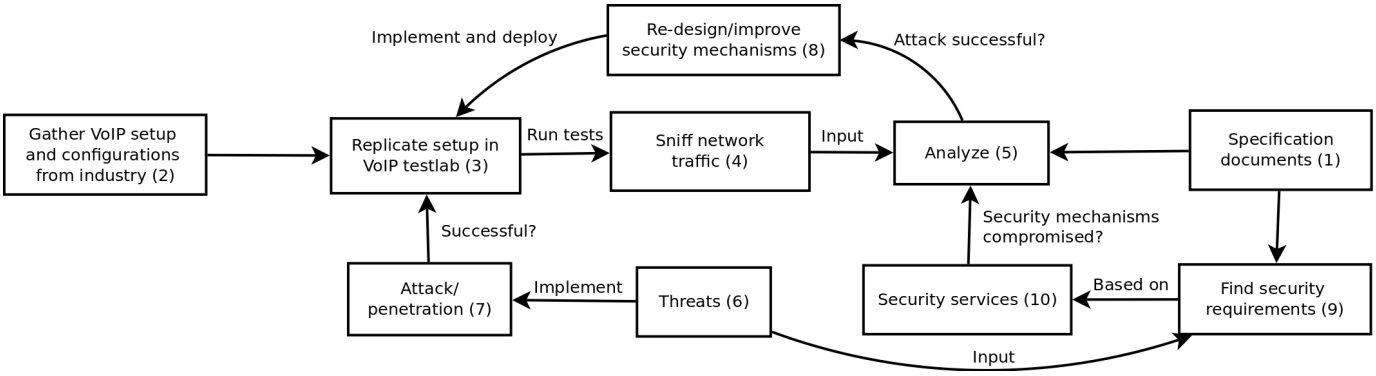


Fig. 2: Workflow for analysis of the SIP authentication method.

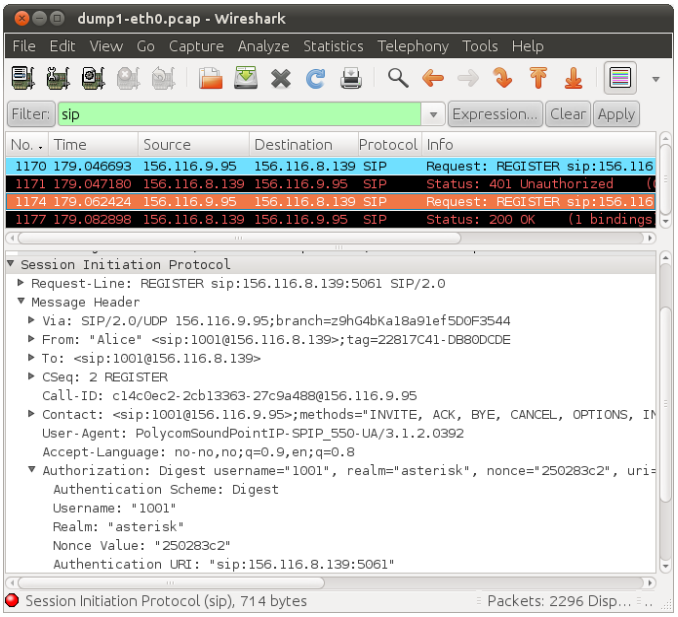


Fig. 3: Network analysis using the network tool Wireshark.

by the real-world attack. After careful analysis of the SIP headers we found that the SIP registration attack could be countered by a modification of the SIP authentication method (8).

We identify three scenarios where identity in SIP needs to be handled, as depicted in Fig. 1: Scenario *I* between the UA and the local SIP server; Scenario *II* between SIP servers; and Scenario *III* end-to-end.

Scenario I between the UA and the local SIP server is relevant when the UA comes online and before any outgoing calls can be placed. Then, the UA must register itself to a local SIP server. During the SIP register handshake, the server usually challenges the UA to authenticate. Before placing a call (sending a SIP `INVITE`), the UA might be challenged again by the server to authenticate. The most common authentication method used between UA and server today is the DAA.

Scenario II handles the authentication between SIP servers to achieve trust between SIP servers. It is not desirable to have SIP traffic handled by an unknown or untrusted SIP server that might have malicious intent. However, since most SIP servers today use some kind of SIP peering [6], the relationships between servers are often static and pre-defined. Therefore the identities between SIP servers are often predetermined by other security mechanisms than what are offered by SIP (like IPSec, TLS etc.).

Scenario III is about end-to-end authentication, which determines the identity of both the caller and the callee across different SIP domains. This is of particular importance and not easily attained in SIP. There is an increased threat and fear for both VoIP phishing and SPIT (Spam over Internet Telephony), that might seriously affect SIP-based VoIP services. By enforcing end-to-end authentication in SIP, these threats might be mitigated or prevented.

We list authentication mechanisms in SIP and their support in these three SIP scenarios in Table I on the facing page.

IV. Digest Access Authentication

The SIP Digest Access Authentication (DAA) [24] is currently the most common authentication scheme for SIP. Other authentication schemes have emerged, but DAA is the only mandatory authentication scheme [8, Section 22]. DAA uses a challenge-response pattern, and relies on a shared secret between client and server. Since the DAA relies on a shared secret and is only meaningful for a specific realm, its usage is limited to Scenario I.

SIP is heavily influenced by the HTTP request-response model, where each transaction consists of a request that requires a particular response. The SIP messages are also similar in syntax and semantics to both HTTP and SMTP [10]. A SIP message consists of several headers and a body. The SIP header fields are textual, always in the format `<header_name>: <header_value>`. The header value can contain one or more parameters. We show an example SIP header message in Fig. 5.

Any SIP request can be challenged for authentication. We show an example SIP DAA handshake in Fig. 4, and refer to the protocol clauses with a number in parentheses. The initial SIP `REGISTER` message (1) from Alice is not authorized and must be authenticated. The SIP server responds with a

TABLE I: List of SIP authentication mechanisms and their support.

Authentication mechanisms	Supported authentication scenarios			Supported SIP methods	
	scenario I	scenario II	scenario III	REGISTER	INVITE
Digest Access Authentication (DAA)	yes	no	no	yes	yes
Secure MIME (S/MIME)	no	no	yes	yes[a]	yes
Secure SIP (SIPS) using TLS	yes	yes	no[b]	yes	yes
P-Asserted Identity	no	yes	no	no	yes[c]
SIP Strong Identity	no	yes	no	no	yes[d]
Password Authenticated Key Exchange (PAKE)	yes	no	no	yes	yes
Generic Security Service API (GSS-API)	yes	yes	yes	yes	yes
Simple Authentication and Security Layer (SASL)	yes	yes	yes	yes	yes

[a] Not intended to be used with SIP REGISTER, however there are no constrains in the SIP specification for using S/MIME in addition to DAA.
[b] SIPS only offers hop-by-hop confidentiality and authentication protection and thus no end-to-end protection.
[c] Does not provide an authentication method *per se*, but provide identity authentication in a trusted environment.
[d] The authentication service is handled by intermediate SIP servers to verify UAs across SIP domains.

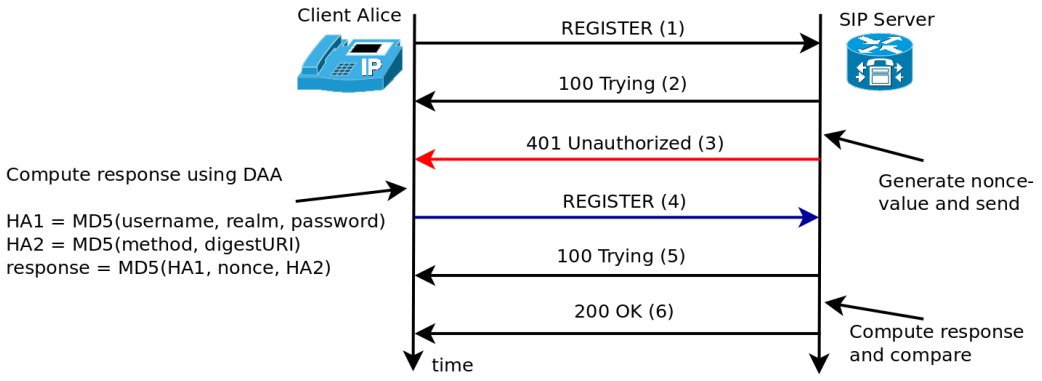


Fig. 4: The SIP Digest Access Authentication method during a SIP REGISTER transaction.

401 Unauthorized status message (3) which contains a WWW-Authenticate header with details of the challenge, including a *nonce* value. The client computes the required SIP digest that is embedded in (4) as an Authorization header. The SIP server, upon receiving the Authorization header, must perform the same digest operation, and compare the result. If the results are identical, the client is authenticated, and a 200 OK message (6) is sent.

The SIP DAA is almost identical to the HTTP digest access authentication [24]. As we will show later, too few attributes (SIP header values) are included in the digest computation, thus leaving some values unprotected. Formally, the DAA is expressed as follows:

$$\begin{aligned} HA1 &= \mathrm{MD5}(A1) \\ &= \mathrm{MD5}(username : realm : password) \\ HA2 &= \mathrm{MD5}(A2) = \mathrm{MD5}(method : digestURI) \\ response &= \mathrm{MD5}(HA1 : nonce : HA2) \end{aligned}$$

In this context, $A1$ is the concatenated string of Alice's $username$, the $realm$ (usually a hostname or domain name) and the shared secret $password$ between Alice and the server. For $A2$, the $method$ is the SIP method used in the current transaction, in the above example that would be REGISTER. In a REGISTER transaction the $digestURI$ is set to the URI in the *To:*-field. The digest authentication $response$ is the hash of the concatenated values of $HA1$, the $nonce$ received from the server, and $HA2$. A SIP REGISTER message with a computed digest embedded in the Authorization header is shown in Fig. 5. DAA provides only reply protection due to the nonce value and one-way message authentication. There is no encryption of the content, nor confidentiality support, except the shared secret $password$ between client and server. All messages are sent in clear. DAA only works within a local domain so cross-domain authentication is not supported, which

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    156.116.9.95;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@156.116.9.95:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: Digest
    username="alice",realm="asterisk",nonce="3b7a1395",response="
    ccbde1c3c129b3dcaa14a4d5e35519d7",uri="sip:CompanyA",algorith
    m=MD5
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

Fig. 5: The only attributes included in the digest response (blue) are depicted in green.

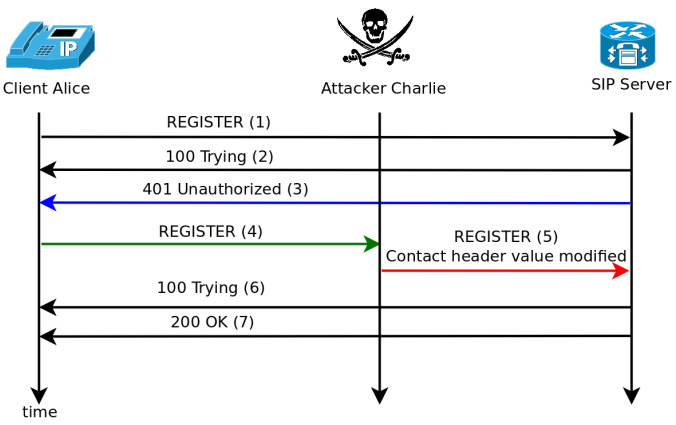


Fig. 6: The attacker Charlie can modify the `Contact` header value, and thereby have all Alice's calls redirected to him.

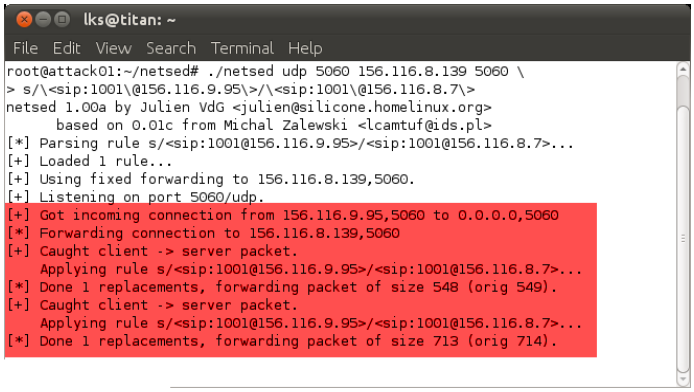


Fig. 7: The network packet stream editor NetSED modifies network packets in real time based on a regular expression (in red).

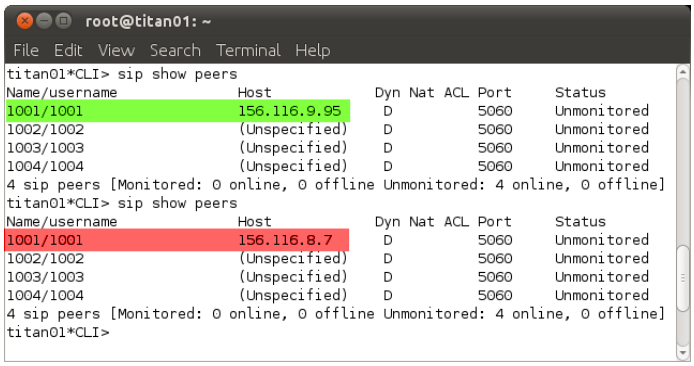


Fig. 8: Host name before (green) and after a successful attack (red), which makes Asterisk believe that Alice's phone (with number 1001) is reachable at an IP-address of the attacker's choice.

implies that end-to-end authentication is not supported. There is no provision in the DAA for the initial secure arrangement between a client and server to establish the shared secret. However, DAA has low computation overhead compared to other methods [18].

A. Attack on Digest Access Authentication

When a UA comes online it registers its contact point(s) to a *location service*. Contact points are the preferred methods a user can be contacted by, for example using SIP, mail, or IM. Usually, only a SIP URI contact method is present. The location service is responsible to redirect SIP requests (for VoIP calls) to the correct SIP end-point. For example, an incoming SIP call destined to `alice@CompanyA.org` does not contain information about which hostname or IP-address Alice's phone can be reached. Therefore, a SIP proxy will query the location service to receive Alice's phone's hostname or IP-address, and then forward the call to this address.

The binding of Alice's phone to a hostname or IP-address is done during the `REGISTER` transaction, as depicted in Fig. 4. Before the binding, or registration, the SIP server should ask the client to authenticate itself, as explained in the previous section. After a successful authentication, the client's hostname or IP-address is registered. A re-registration is normally done at regular intervals. This registration is repeated usually every 3-15 minutes, depending on the configuration. The client's preferred contact methods, including hostname or IP-address, is carried in the SIP header `Contact`, as depicted in Line 5 in Fig. 5. However, this SIP header value is sent in clear, and is not protected by DAA. Thus, the registration is vulnerable to a man-in-the-middle attack [25].

If an attacker modifies the hostname or IP-address in the *contactURI* header value during a `REGISTER` phrase, as depicted in Fig. 6, all requests, and hence calls, to the client will be diverted to a hostname or IP-address controlled by an attacker. Here, Alice cannot perceive that she is unreachable. An attacker can modify Alice's `REGISTER` session in real-time using NetSED [45] as depicted in Fig. 7. The SIP server (Asterisk), will not detect nor suspect that anything is wrong, and register Alice's phone number with the attackers IP address, as seen on Asterisk's terminal in Fig. 8. When Asterisk receives a call to Alice, the call will be forwarded to the attackers registered IP address. If this vulnerability is left incorrect, it constitutes a fatal flaw.

B. Improving the Digest Access Authentication

The SIP digest authentication is weak, which is stated in both the SIP specification [8], and the digest specification [24]. Specifically, DAA only offers protection of the value in the `To` header called the `Request-URI` and the *method*, but no other SIP header values are protected.

A minor modification of DAA can counter the registration hijack attack [25], which is caused by having too few SIP header parameters protected by the digest. Since an attacker can modify and redirect all requests, we protect the header by including the `Contact` header value in the digest. By including the `Contact` value, which we name *contactURIs* in the digest, we effectively counter the registration hijack attack.

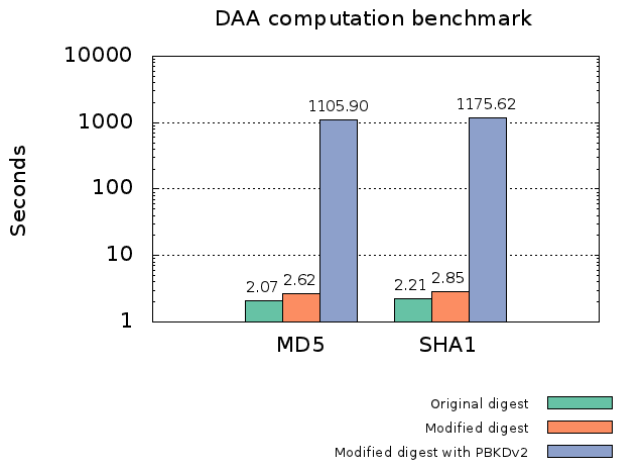


Fig. 9: The computation overhead for 100.000 iterations for original DAA, our modified DAA, and modified DAA with PBKDFv2 for both MD5 and SHA1.

We define $HA0$ with $contactURIs$. The new digest computation algorithm is as follows:

$$\begin{aligned} HA0 &= \text{MD5}(A0) = \text{MD5}(contactURIs) \\ HA1 &= \text{MD5}(A1) \\ &= \text{MD5}(username : realm : password) \\ HA2 &= \text{MD5}(A2) = \text{MD5}(method : digestURI) \\ response &= \text{MD5}(HA0 : HA1 : nonce : HA2) \end{aligned}$$

Weaknesses in the MD5 hash have been found. In particular we mention collision attacks where two different input values produce the same MD5 hash [46]. This weakness is not known to be exploitable to reveal a user's password [47]. Nonetheless, a stronger hash function, like SHA1 [48], is recommend.

We implemented and tested our modified DAA by using the Python Twisted [49] networking engine, using both MD5 and SHA1. According to our test, the computation overhead by including $HA0$ with the $ContactURIs$ is minimal, as shown in Fig. 9. The difference between the original DAA and our modified DAA with MD5 for 100.000 authentication requests on a 2.2Ghz Intel CPU, is only 0.55 seconds, a negligible amount.

A modified DAA means a modification of the SIP standard. Since the SIP standard has seen widespread industry adoption, it can be difficult to re-deploy a non-standardized SIP DAA. To prevent a modification of the SIP standard, we can use the DAA parameter `auth-param` to store our modified digest response. The parameter `auth-param` is reserved "for future use" [24, page 12], and can be a part of the `Authorization` header.

SIP devices that do not support the updated and more secure digest, can and will ignore this value, and use the original DAA for authentication. However, we cannot recommend this approach, since an attacker could remove this value and force the usage of the original standardized DAA. We would prefer to modify the DAA digest computation to force an upgrade to the new improved DAA method, instead of compromising on security.

C. Using a Password-Based Key Derivation Function

The improved DAA, described above, is still vulnerable to dictionary-based off-line (brute-force) attacks. The attacker can intercept the message exchange, and do an exhaustive (brute-force) search for the password. To increase the cost of such search, we add support for a key derivation technique with the purpose of increasing the cost of producing the digest from the shared secret, thereby also increasing the difficulty of the brute-force attack.

We introduce support for "Password Based Key Derivation Function version 2" (PBKDFv2) as specified by Kaliski [50] from RSA Laboratories. PBKDFv2 works by using a key derivation function (KDF) on the password (P) and salt (S) to derive the key (DK) as:

$$DK = \text{KDF}(P, S)$$

When applied to the DAA, P is the shared secret and S is the nonce issued from the SIP server. The DK is derived by these required steps:

1) The maximum length $dkLen$ of the derived key DK is given as:
$$dkLen > (2^{32} - 1) * hLen$$
where $hLen$ denotes the length in octets of the pseudo-random function output, which is 16 for MD5 and 20 for SHA-1. We implement and benchmark both MD5 and SHA-1. However, the use of MD5 is not recommended due to weaknesses and attacks found [51].
2) We let l be the number of $hLen$-octet blocks in the derived key, rounding up, and r the number of octets in the last block:
$$\begin{aligned} l &= \left\lceil \frac{dklen}{hLen} \right\rceil \\ r &= dkLen - (l-1) * hLen \end{aligned}$$
3) For each block of the derived key, the function F is applied. The function F take password P, salt S, the iteration count c and the block index to compute the block:
$$\begin{aligned} T_1 &= F(P, S, c, 1) \\ T_2 &= F(P, S, c, 2) \\ &... \\ T_l &= F(P, S, c, l) \end{aligned}$$
Here, function F is defined as the exclusive-or sum of the first c iterates of the underlying pseudo-random function PRF (using HMAC-SHA1 [52]) applied to the password P and the concatenation of the salt S and the block index i:
$$F(P, S, c, i) = U_1 \oplus U_2 \oplus ... \oplus U_c$$

where:

$$\begin{aligned} U_1 = &\quad PRF(P, S \parallel INT(i)) \\ U_2 = &\quad PRF(P, U_1) \\ &\ldots \\ U_c = &\quad PRF(P, U_{c-1}) \end{aligned}$$

Here, $INT(i)$ is a four-octet encoding of the integer i, most significant octet first.

4) Then the blocks are concatenated and the first $dkLen$ octets is extracted to produce a derived key DK:

$$DK = T_1 \parallel T_2 \parallel ... \parallel T_l < 0..r-1 >$$

5) The derived key DK is returned base64-encoded [53].

We implemented and tested DAA with PBKDFv2 with c iterations set to the recommended value of 1000. Input to the PBKDFv2 is the password (shared secret) and the nonce from the SIP server. The new modified DAA replaces the $password$ with the derived key DK from PBKDFv2, thus a modified DAA algorithm is as follows:

$$\begin{aligned} HA0 = &\quad \mathrm{MD5}(A0) = \mathrm{MD5}(contactURIs) \\ HA1 = &\quad \mathrm{MD5}(A1) \\ = &\quad \mathrm{MD5}(username : realm : DK) \\ HA2 = &\quad \mathrm{MD5}(A2) = \mathrm{MD5}(method : digestURI) \\ response = &\quad \mathrm{MD5}(HA0 : HA1 : nonce : HA2) \end{aligned}$$

As shown in Fig. 9, the computation overhead using PBKDFv2 is significant compared to the original DAA. The result is as expected, since each DAA computation using PBKDFv2 calls a HMAC function 1000 times. This increase the cost of an exhaustive brute-force search for the shared secret used by DAA, without a significant impact of deriving individual DK used by a UA to authenticate with DAA.

While DAA with PBKDFv2 reduces much of the risk of a brute-force dictionary attack, it does not provide us with means to authenticate the SIP server.

V. Password Authenticated Key Exchange

In the following, we discuss how to add support for a variant of PAKE denoted as "Key Agreement Method 3" (KAM3) as a cryptographic protocol [26, page 17]. PAKE has the following attractive features: *1*) PAKE provides mutual authentication between UA and the SIP server, and thus a rogue SIP server can not claim that the authentication succeed without knowing the shared password. PAKE assures the UA that the SIP server knows the UA's encrypted password. *2*) Reuse of the shared password used by DAA as the UA's credential, which enables our approach to easily replace DAA used within a local SIP domain (scenario I). *3*) PAKE offers strong protection of the shared secret if the communication is eavesdropped, that prevents brute-force attacks, including dictionary-based off-line attacks, to which the DAA is vulnerable to.

Our approach follows the work of Oiwa et al. [54]. They use KAM3 to introduce a stronger authentication in HTTP and their initial design and specification is submitted to the IETF as an Internet Draft [55]. We have adapted their approach to SIP, since SIP closely resembles HTTP in both message structure and flow, and we need to prevent the `REGISTER` hijack attack presented earlier [1].

In KAM3, the UA and the SIP server compute cryptographic keys based on the shared password. These keys are exchanged, and a shared session secret is computed based on these keys. Each peer sends then a hash value computed of the session secret and some other values, to the requesting peer. The receiving peer computes the same hash value, and compares it with the received hash value. If these are identical, the sending peer is authenticated.

PAKE supports several authentication algorithms, which differ in their underlying mathematical groups and security parameters [55]. The only mandatory supported authentication algorithm, the *iso-kam3-dl-2048-sha256*, uses the 2048-bit discrete-logarithm defined in RFC3526 [56] and the SHA-256 hash function.

A. Initial requirements

In the following section, we let q an odd prime integer defining the number of elements in $F(q)$ which is a representation of a finite group. We let g the generator of a subgroup of r elements in $F(q)$. The one-way hash function is denoted as H.

Before the authentication starts, username and password must be set and configured. We compute a weak secret π used by the client as a one-way hash of the values $realm$, $username$ and $password$:

$$\pi = H(realm, username, password)$$

Here, $realm$ is the protection domain where SIP authentication is meaningful for a set of $username$ and $password$. The server does not need to store the shared password directly, only a specially encrypted version $J(\pi)$, where J is the password verification element derivation function defined as:

$$J(\pi) = g^{\pi} \bmod q$$

B. PAKE message exchange

We need to extend the current SIP `REGISTER` handshake by one extra round-trip of SIP messages between the UA and the SIP server. These two extra messages are depicted in blue and numbered (4) and (5) in Fig. 10. A more detailed specification is given in the following paragraphs, where the numbers refer to the protocol clauses depicted in Fig. 10.

The UA registers to a SIP *location service* (SIP server). The initial SIP `REGISTER` message (1) from the UA is not authorized, and must be authenticated. The SIP server responds with a `401 Unauthorized` status message (2), which contains a `WWW-Authenticate` header with details of the challenge, including *realm* and *algorithm*. The UA constructs a cryptographic value w_a generated from a random integer s_a:

$$w_a = g^{s_a} \bmod q$$

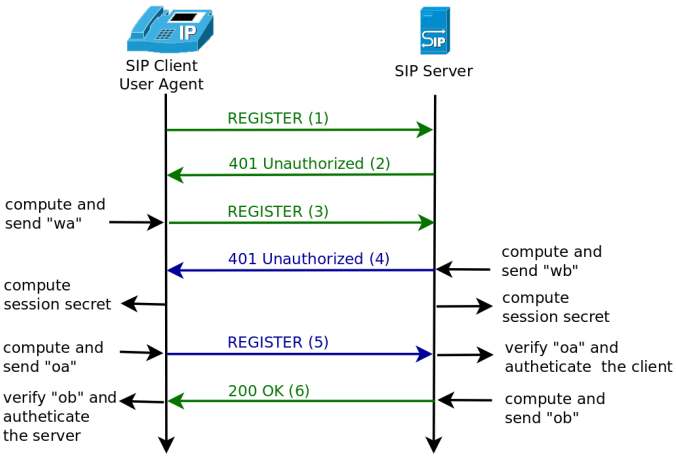


Fig. 10: SIP `REGISTER` message flow with mutual authentication security using PAKE.

This value is sent in a new SIP `REGISTER` message (3) to the SIP server. The SIP server proceeds to generate and send another cryptographic value w_b, which is generated from $J(\pi)$, the received value w_a and a random integer s_b:

$$w_b = (J(\pi) \times w_a^{H(1,w_a)})^{s_b} \bmod q$$

At the next step, each peer computes a session secret z. The UA derives z based on π, s_a, w_a and w_b:

$$z = w_b^{(s_a + H(2,w_a,w_b))/(s_a * H(1,w_a) + \pi) \bmod r} \bmod q$$

Likewise, the SIP server derives z based on s_b, w_a and w_b using the following function:

$$z = (w_a \times g^{H(2,w_a,w_b)})^{s_b} \bmod q$$

The session secret z matches only if both peers have used the secret credentials generated from the same shared secret. The above equations are directly derived from the PAKE HTTP authentication specifications [55]. The next step is to validate the value of z at the communicating peer.

The UA sends a third SIP `REGISTER` message (5) and includes the value o_a which is a hash value computed as:

$$o_a = H(4, w_a, w_b, z, contactURIs)$$

Here, $contactURIs$ is the value of the UA's `Contact` SIP header value. This value is integrity-protected to prevent register hijacking attacks as presented in [1]. The SIP server, upon receipt of o_a, performs the same hash operation, and compares the results. If these results are identical, the UA is authenticated. The SIP server then sends a final message (6), with the value o_b computed as:

$$o_b = H(3, w_a, w_b, z, contactURIs)$$

When the UA receives o_b, it verifies this value by computing its hash value. If the results are identical, the SIP server is authenticated to the UA. After a complete message exchange, the UA is authenticated to the SIP server, and the SIP server has been authenticated to the UA.

C. SIP message support for PAKE

We embed the cryptographic values derived in the previous section as base64-encoded [53] SIP header values. We re-use the SIP DAA headers to carry PAKE authentication data, so that PAKE can be used as a drop-in replacement for DAA. A SIP `REGISTER` message with a DAA `Authorization` header is depicted in Fig. 15. Again, we refer to the protocol clauses with a number in parentheses as depicted in Fig. 10.

The UA first sends a SIP `REGISTER` without any authentication credentials (1). The SIP server responds with a `401 Unauthorized` status message (2), which contains a `WWW-Authenticate` header with header values *realm* and *algorithm*:

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: Mutual realm="asterisk",
  algorithm="iso-kam3-dl-2048-sha256"
```

The UA then computes w_a and sends it to the SIP server using a new SIP `REGISTER` message (3), with the required values embedded in the `Authorization` header:

```
SIP/2.0 REGISTER
Authorization: Mutual user="alice",
  algorithm="iso-kam3-dl-2048-sha256",
  wa="Q29tcHV0ZWQgd2E...ljaCBcyBsb25nCg=="
```

The next required values in the authentication mechanism w_b, o_a and o_b are embedded and sent using these two SIP headers.

VI. Security Programming Interfaces

A modified PAKE authentication can more easily replace the current digest (DAA) authentication used in SIP, since they both rely on a shared secret and use the same SIP headers. PAKE also introduces a stronger authentication than DAA. However, a more flexible authentication mechanism is desired. Different VoIP scenarios require different security requirements, and the communicating peers should be able to negotiate the best possible authentication mechanism supported.

Instead of adding numerous different authentication mechanisms to SIP based on different security requirements, it is desirable to keep the changes to the SIP standard to a minimum. The industry might also be reluctant to adopt immature and non-standardized security services, like different (new) authentication mechanisms. Adding support to a security programming interface will require only small changes to the SIP standard.

A security programming interface provides a generic interface for application layer protocols like SIP, with a layer of abstraction for different security services like authentication, integrity or confidentiality. Using a security programming interface, an application does not need to support or implement every authentication method, but use the provided security API [57]. Support for two security programming interfaces, the "Generic Security Services API" (GSS-API) and "Simple Authentication and Security Layer" (SASL), are added to SIP.

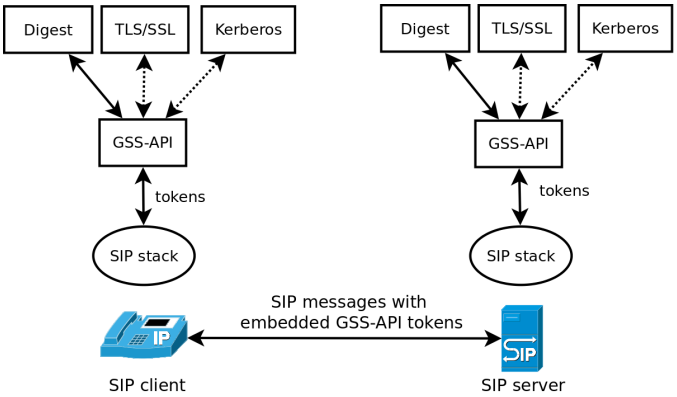


Fig. 11: The GSS-API interface in SIP.

Both are developed by the IETF, have been extensively tested, and are now classified as mature standards by the IETF.

A. Generic Security Services API

The GSS-API is not a communication protocol in itself, but relies on the application to encapsulate, send, and extract data messages called "tokens" between the client and server. The tokens' content are opaque from the viewpoint of the calling application, and contain authentication data, or, once the authentication is complete, portion of data that the client and server want to sign or encrypt. The tokens are passed through the GSS-API to a range of underlying security mechanisms, ranging from secret-key cryptography, like Kerberos [58], to public-key cryptography, like the Simple Public-Key GSS-API Mechanism (SPKM) [59]. The GSS-API interface to SIP is depicted in Fig. 11. For an application, the use of the GSS-API becomes a standard interface to request authentication, integrity, and confidentiality services in a uniform way. However, GSS-API does not provide credentials needed by the underlying security mechanisms. Both server and client must acquire their respective credentials before GSS-API functions are called.

To establish peer entity authentication, a security context is initialized and established. After the security context has been established, additional messages can be exchanged, that are integrity and, optionally, confidentially protected. To initiate and manage a security context, the peers use the *context-level* GSS-API calls. The client calls `GSS_Init_sec_context()` that produces a "output_token" that is passed to the server. The server then calls `GSS_Accept_sec_context()` with the received token as input. Depending on the underlying security mechanism, additional token exchanges may be required in the course of context establishment. If so, `GSS_S_CONTINUE_NEEDED` status is set and additional tokens are passed between the client and server until a security context is established, as depicted in Fig. 14.

After a security context has been established, *per-message* GSS-API calls can be used to protect a message by adding a Message Integrity Code (MIC) with `GSS_GetMIC()` and verifying the message with `GSS_VerifyMIC()`. To encrypt and decrypt messages, the peers can use `GSS_Wrap()` and `GSS_Unwrap()`. Thus, two different token types exist:

1) *Context-level tokens* are used when a context is established.
2) *Per-message tokens* are used after a context has been established, and are used to integrity or confidentiality protect data.

In addition to send and receive tokens, the application is responsible to distinguish between token types. This is necessary because different tokens types are sent by the application to different GSS-API functions. But since the tokens are opaque to the application, the application must use a method to distinguish between the token types. In our solution, we use explicit tagging of the token type that accompanies the token message.

1) SIP message support for GSS-API: When a SIP client is authenticated to a server using DAA, the authentication handshake data is encapsulated in the `WWW-Authenticate` header from server to client, and the `Authorization` header from client to server. We reuse these headers for GSS-API support, and instead of encapsulate DAA data, we send the GSS-API tokens. An example of both DAA `Authorization` header and the new `Authorization` header with GSS-API data is depicted in Fig. 12.

During the initialization of a security context it is necessary to identify the underlying security mechanism to be used. The caller initiating the context indicates at the start of the token the security (authentication) mechanism to be used. The security mechanism is denoted by a unique Object Identifier (OID). For example, the OID for the Kerberos V5 mechanism is `1.2.840.113554.1.2.2`. However, the initiating peer cannot know which security mechanism the receiving peer supports. If an unsupported "mech_type" is requested, the authentication fails. The GSS-API standard resolves this by recommending to manually standardizing on a fixed "mech_type" within a domain. Since SIP addresses are designed to be global [6], and not confined to a local domain, a GSS-API *negotiation* mechanism is required. The SPNEGO is such a GSS-API negotiation mechanism.

The "Simple and Protected GSSAPI Negotiation Mechanism" (SPNEGO [60] is a pseudo security mechanism that enables peers to negotiate a common set of one or more GSS-API security mechanisms. The GSS-API stack with SPNEGO is shown in Fig. 13. The client sends a prioritized list of supported authentication mechanisms to the server. The server then chooses the preferred authentication method based on the received list from the client. The client initiates `GSS_Init_sec_context()` as with an ordinary GSS-API security mechanism, but requests that SPNEGO is used as the underlying GSS-API mechanism ("mech_type"). The SPNEGO handshake between client and server is communicated by sending and receiving tokens. After the handshake, the client and server initiate and set up a security context (authentication) using the agreed GSS-API security mechanism.

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP 192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: Digest
    username="alice",realm="asterisk",nonce="3b7a1395",response="ccbde1c3c129b3dcaa14a4d5e35
    519d7",uri="sip:CompanyA",algorithm=MD5
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP 192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: GSSAPI ttype="context"
    token="0401000B06092A864886F712010202DACD139402AAF44350CDE32"
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

Fig. 12: A SIP `REGISTER` message with the original DAA `Authorization` header to the left, and the same header carrying GSS-API data to the right.

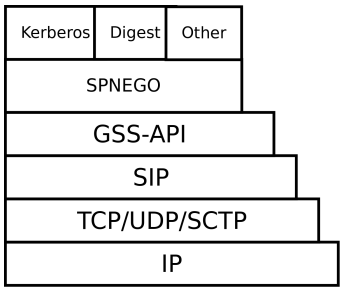


Fig. 13: The GSS-API protocol stack with the SPNEGO negotiation mechanism and underlying security mechanisms.

2) SIP authentication using GSS-API and SPNEGO: When discussing PAKE authentication earlier, we added one round-trip of SIP messages between the UA and the SIP server. When using GSS-API with the SPNEGO, the number of SIP messages going back and forth depends on the underlying authentication mechanism. We therefore extend the SIP `REGISTER` handshake with an arbitrary number of round-trips, until the underlying authentication mechanism has completed communication.

In the following paragraphs, the numbers in parentheses refer to the numbers in Fig. 14. When a client comes online and registers itself to a "location service" (SIP server), it does so by sending a SIP `REGISTER` message (1). We define the token type in the variable `ttype`. In the following messages, the `ttype` is set to "context" indicating that these tokens are *context-level tokens*. The first message (1) does not contain any `Authorization` header. The server responds with an empty `WWW-Authenticate` header (3):

```
REGISTER SIP/2.0
WWW-Authenticate: GSSAPI ttype="context"
  token=""
```

The client then calls `GSS_Init_sec_context()` with SPNEGO as underlying GSS-API mechanism to negotiate a common authentication mechanism (4). The GSS-API "mech_type" is set to SPNEGOs OID `1.3.6.1.5.5.2`. The token data might be in binary format, depending on the security mechanism used. Since the SIP headers are in ASCII string format, the token data is base64 encoded:

```
SIP/2.0 401 Unauthorized
Authorization: GSSAPI ttype="context"
  token="0401000B06092A864886F7120..."
```

The server retrieves the GSS-API data, the token, and passes this to the SPNEGO GSS-API mechanism. In this first initial token, the client embeds authentication data for its first preferred authentication mechanism. This way, should the server accept the clients preferred mechanism, we avoid an extra SIP message round trip. If the client's preferred method was accepted by the server, the server passes the relevant authentication data to the selected authentication mechanism in a `401` SIP message (5). The selected authentication method continues to pass tokens between client and server as many times as necessary to complete the authentication (6-7-N) and establish a security context. Once the security context is established, it sends a `200 OK` SIP message (N+2). Should the server have some last GSS-API data to be communicated to the client to complete the security context, it can be carried in a `WWW-Authenticate` header embedded in the `200 OK` message:

```
SIP/2.0 200 OK
WWW-Authenticate: GSSAPI ttype="context"
  token="dd02c7c2232759874e1c20558701..."
```

If the client's preferred mechanism is not the server's most preferred mechanism, the server outputs a negotiation token and sends it to the client embedded in a new `401` SIP message (5). The client processes the received SIP message and passes the authentication data to the correct authentication mechanism. The GSS-API then continues as described in the previous paragraph.

B. Simple Authentication and Security Layer

The Simple Authentication and Security Layer (SASL), defined in RFC4422 [28], provides an interface for authentication and an authentication negotiation mechanism. It provide the same security services as GSS-API and is implemented and used in several popular communications protocols applications like IMAP, SMTP and LDAP[1].

As with GSS-API, the SASL framework does not provide authentication mechanisms in itself, but supports different underlying authentication mechanisms through a standardized

[1]The Carnegie Mellon University's implementation: http://asg.web.cmu.edu/sasl/ and the GNU SASL library: http://www.gnu.org/software/gsasl/ are two popular and freely available SASL libraries.

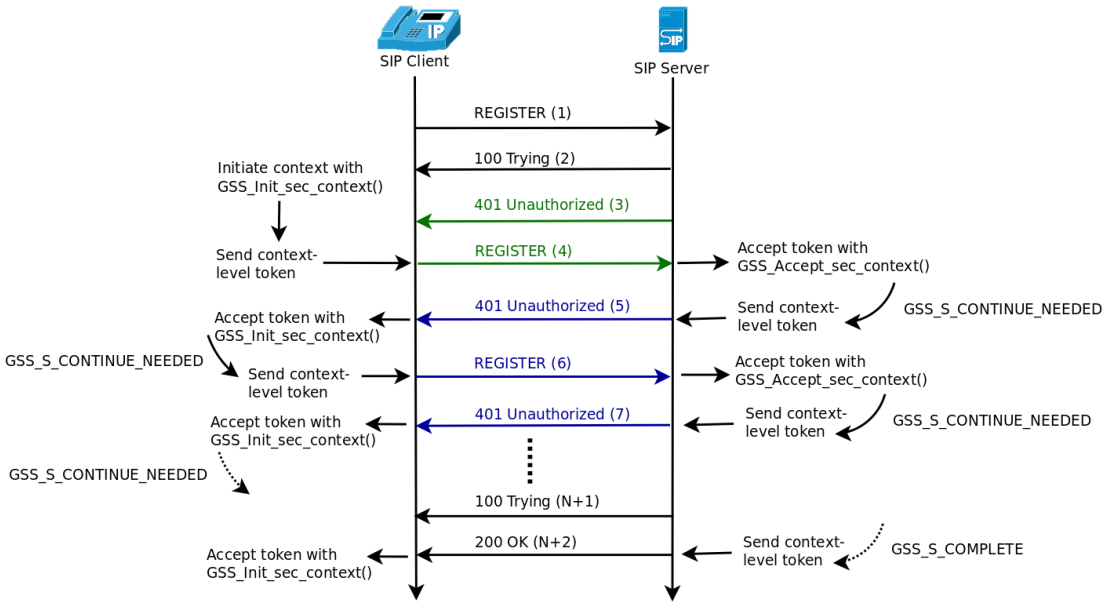


Fig. 14: SIP `REGISTER` message flow with GSS-API security context establishment (authentication).

interface[2]. SASL does not provide a transport layer and thus relies on the application, to encapsulate, send and extract SASL messages between client and server, which in our case is the SIP protocol. The SASL messages sent between client and server contain authentication data, and are opaque from the viewpoint of the calling application (SIP). The application only needs to add support to a SASL software library implementation, and thus have support to a range of underlying authentication mechanisms the library supports.

While the GSS-API is primarily intended for use with applications, SASL is used in, and intended for, communication protocols. The functionalities offered by the GSS-API and SASL are alike, but the SASL specification is more high-level, and allows more freedom in implementing the SASL requirements. SASL also supports more underlying security mechanisms than the GSS-API. By using the "GS2" mechanism family, the GSS-API can be used as an underlying security mechanism in SASL. However, the GSS-API negotiation mechanism SPNEGO cannot be used due to security concerns [61, Section 14].

1) SIP message support for SASL: In SASL terminology, the description on how to encapsulate SASL negotiation and SASL messages for a given protocol, is called a "SASL profile". The SIP protocol stack with SASL is shown in Fig. 16. We create a SASL profile for SIP by reusing the `WWW-Authenticate` and `Authorization` SIP headers used by the digest authentication, shown earlier. Instead of encapsulating DAA data, we embed SASL messages, as depicted in Fig. 15.

As with the GSS-API, we need to increase the number of messages going back and forth between the SIP client and server. The number of messages depends on the required message exchange needed by the used underlying authentication mechanism.

In the following paragraphs, the numbers in parentheses refer to the SIP message numbers in Fig. 14. The SASL specification only outlines a very high-level method of how the server should advertise its supported mechanisms to the client. We implement the mechanism negotiation in the first three messages in the SIP `REGISTER` handshake (1-4). The UA starts by requesting authentication from the SIP server, with no `Authorization` header (1). The SIP server responds with a `401 Unauthorized` SIP message (3), with the supported and available mechanisms embedded in the `WWW-Authenticate` header:

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: SASL
  negotiate="DIGEST-MD5 NTLM GS2-KRB5"
```

The client selects the best mechanism from the received list that it supports and sends a new SIP `REGISTER` message (4). This message includes an `Authorization` header requesting authentication with "GS2-KRB5" as the preferred mechanism. The initial authentication data is embedded base64 encoded to the *data* parameter:

```
SIP/2.0 REGISTER
Authorization: SASL mechanism="GS2-KRB5",
  data="SUZZT1VDQU5SR...JUPVVQU5FUkQK="
```

The server retrieves the SASL data, and passes the message to the SASL library which handles the authentication. The selected authentication method continues to pass SASL messages between client and server as many times as necessary to

[2]A list of registered SASL mechanisms is maintained by IANA: http://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xml

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: Digest
    username="alice",realm="asterisk",nonce="3b7a1395",response=
    "ccbde1c3c129b3dcaa14a4d5e35519d7",uri="sip:CompanyA",
    algorithm=MD5
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: SASL mechanism="DIGEST-MD5"
    data="YAzgusSGGeRFGw9nfUvOAxcEDzZCBmKY1H2E1negaccBcx3DUSkGNW
    Y4qfiSwcXwjLtoqW0eBNog7ixHN"
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

Fig. 15: A SIP `REGISTER` message with the original DAA `Authorization` header to the left, and the same header carrying SASL data to the right.

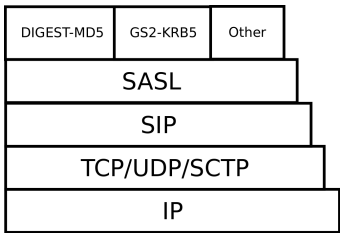


Fig. 16: The SIP SASL stack is similar to the SIP GSS-API stack with underlying security mechanisms.

complete the authentication (messages 5-6 are repeated). Once the authentication is complete, the SIP server sends a `200 OK` SIP message. Should the server have some last SASL data to be communicated to the client to complete the authentication, it can be carried in a `WWW-Authenticate` header embedded in the `200 OK` message (N+2):

```
SIP/2.0 200 OK
WWW-Authenticate: SASL mechanism="GS2-KRB5",
  data="TFoG9rP56zrVH...YaAOndwPew6NdxKr"
```

As soon as the `200 OK` message is received and processed, the client is authenticated to the SIP server. Since the mechanism negotiation is not integrity-protected, the UA is vulnerable to a "down-grade" attack. An attacker can intercept and modify the negotiation messages so that the least favorable authentication method is used.

VII. Migration towards a secure authentication

We propose a two step migration towards a secure authentication in SIP. While our attack on the DAA could be countered by including the SIP header value `ContactURI` in the digest, it did not provide any protection against off-line dictionary attacks. We implemented and showed that the use of "Password-Based Key Derivation Function version 2" (PBKDFv2) on the shared secret to make dictionary- and brute-force attacks significant harder to execute on the DAA. However, this method does not authenticate the SIP server, only the client.

Our *first* migration step suggests to replace the DAA with a modified "Password Authenticated Key Exchange" (PAKE) that is more secure than the DAA, introduce mutual authentication and re-use the shared secret used by the DAA. These properties make PAKE a preferred mechanism over the DAA with PBKDFv2. However, using PAKE does not leave any room for future extensions nor modification of authentication in SIP once implemented.

The *second* migration step takes the limitations of the previous mechanisms into consideration, and is seen as the most viable way of solution. The last authentication method introduces support for a GSS-API/SASL security layer which enables SIP to transparently support and use more secure authentication methods in a unified and generic way without the need for later changes to the SIP protocol specification.

Support for the GSS-API/SASL security layer in SIP, have the following attractive properties that address real-world security concerns:

1) Mature, stable and industry adopted standards: The industry might be reluctant to adopt immature and non-standardized security services, like different (new) authentication mechanisms. Both the GSS-API and SASL are stable, mature standards that have been adopted by the industry. Thus, implementing GSS-API or SASL should not be considered a drastic nor radical change by the relevant standardizing bodies (the IETF) nor the VoIP industry.
2) Minimal changes to the SIP standard required: The authentication data re-use the existing SIP DAA headers, so minimal changes to the SIP *message contents* are required. Also, minimal changes are required to the SIP *message flow*, since the authentication handshake is just extended with a number of required SIP message round-trips to complete the new authentication exchange.
3) Flexible and adaptive to new requirements and future changes: Instead of adding numerous different authentication mechanisms to SIP based on different security requirements, it is desirable to keep the changes to the SIP standard to a minimum. By adding support to a security layer in SIP, adding new or modifying existing underlying authentication mechanisms does not need any redesign of the SIP specification standard. In this case, only the GSS-API/SASL software library needs to be updated.

Thus, authentication in SIP becomes adaptive to future extensions.

VIII. Conclusion and Future Work

We have seen that the widely deployed authentication method DAA in SIP is weak and vulnerable to attacks. Moreover, we have confirmed and verified that the attack analyzed earlier [25] can be performed on the SIP protocol in real-time. We have examined this authentication method, and proposed a solution to counter the serious registration attack. By including more SIP header parameters in the authentication digest this attack can be countered.

The original SIP designers focused on functionality and compliance at the cost of security. A more thorough investigation of the SIP DAA in the design phase would have revealed the vulnerability presented here, and the vulnerability could have been prevented early on. Our remedy presented here solves a serious problem with the DAA.

Therefore, we wanted to replace DAA with support for an better, more robust authentication scheme. We have added support for a improved authentication mechanism that can easily replace DAA based on a modified PAKE algorithm. This new authentication mechanism adds support for mutual authentication and is more secure than DAA. We have also shown that the modified PAKE authentication can easily function as a drop-in replacement for DAA. However, a more flexible authentication mechanism is desired in the long-term. Different VoIP installations have different security requirements that may require different security services.

We introduced a security programming interface, which provides a security abstraction layer. This abstraction layer adds support to a range of underlying authentication mechanism in a unified way. As long as SIP supports the security layer, new authentication mechanisms can be added later, without requiring any change to the SIP protocol. Support for two security layers were added, the GSS-API and SASL. We recommend the use of SASL, as SASL has more industry deployment, has support for more underlying authentication mechanisms, and is specifically designed for communications protocols.

We envisage a two-step migration towards a stronger authentication scheme in SIP. First, the modified PAKE authentication is implemented and deployed. Second, the long-term solution is to deploy SASL with support for a range of underlying authentication mechanisms.

Future work will look into implementing a proof of concept for PAKE-enabled UA and SIP server, including overhead evaluation benchmarks for the new authentication algorithm. We also plan to evaluate different SASL security mechanisms and their implications for SIP, and decide which authentication mechanisms should be mandatorily supported through SASL.

We plan to co-operate with the IETF and the "kitten" WG to further elaborate GSS-API and SASL support for SIP. We hope our work will gain acceptance and industrial deployment, so that the previously mentioned security attacks can be countered.

Acknowledgment

This research is funded by the EUX2010SEC project in the VERDIKT framework of the Norwegian Research Council (Norges Forskningsråd, project 180054).

References

[1] L. Strand and W. Leister, "Improving SIP authentication," in *Proceedings of the Tenth International Conference on Networking (ICN2011)*. Xpert Publishing Services, Jan 2011, pp. 164 – 169.

[2] L. Strand, J. Noll, and W. Leister, "Generic security services API authentication support for the session initiation protocol," in *Proceedings of The Seventh Advanced International Conference on Telecommunications (AICT2011)*. Xpert Publishing Services, Mar 2011, pp. 117 – 122.

[3] L. Strand, W. Leister, and A. Duric, "Migration towards a more secure authentication in the session initiation protocol," in *The Fifth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE2011)*. Xpert Publishing Services, Aug 2011.

[4] "Det norske markedet for elektroniske kommunikasjonstjenester 2009 (The Norwegian market for electronic communication services 2009)," Post- og teletilsynet (The Norwegian Post and Telecommunications Authority), 2010. [Online]. Available: http://www.npt.no/ikbViewer/Content/119027/Ekomrapport_2009_.pdf [Accessed: 1. Jul 2011]

[5] Telecommunication Development Sector (ITU-D), "The world in 2010," ITU-T ICT facts and figures, 2010.

[6] L. Strand and W. Leister, "A Survey of SIP Peering," in *NATO ASI - Architects of secure Networks (ASIGE10)*, May 2010.

[7] International Telecommunication Union, "7 kHz Audio-Coding within 64 kbits/s," ITU-T Recommendation G.722, 1993.

[8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt [Accessed: 1. Jul 2011]

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051, 6222. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt [Accessed: 1. Jul 2011]

[10] H. Sinnreich and A. B. Johnston, *Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., August 2006.

[11] H. Dwivedi, *Hacking VoIP: Protocols, Attacks, and Countermeasures*, 1st ed. No Starch Press, Mar. 2009.

[12] D. Endler and M. Collier, *Hacking Exposed VoIP: Voice over IP Security Secrets and Solutions*. McGraw-Hill Osborne Media, November 2006.

[13] A. M. Hagalisletto and L. Strand, "Designing attacks on SIP call setup," *International Journal of Applied Cryptography*, vol. 2, no. 1, pp. 13–22(10), July 2010.

[14] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, and H. Schulzrinne, *SIP Security*. WileyBlackwell, Mar. 2009.

[15] VoIPSA, "VoIP security and privacy threat taxonomy," Public Realease 1.0, Oct. 2005. [Online]. Available: http://voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf [Accessed: 1. Nov 2011]

[16] D. York, *Seven Deadliest Unified Communications Attacks*. Syngress, Apr. 2010.

[17] P. Park, *Voice over IP Security*. Cisco Press, Sep. 2008.

[18] S. Salsano, L. Veltri, and D. Papalilo, "SIP security issues: The SIP authentication procedure and its processing load," *Network, IEEE*, vol. 16, pp. 38–44, 2002.

[19] D. Kuhn, "Sources of failure in the public switched telephone network," *Computer*, vol. 30, pp. 31–36, 1997.

[20] A. D. Keromytis, *Voice over IP Security - A Comprehensive Survey of Vulnerabilities and Academic Research*, 1st ed. New York, NY: Springer New York, 2011, vol. 1.

[21] International Telecommunication Union (ITU), "Security Architecture For Open Systems Interconnection (OSI)," The International Telegraph and Telephone Consultative Comittee (CCITT), X.800 Standard X.800, 1991.

[22] "Research project: EUX2010SEC – Enterprise Unified Exchange Security"." [Online]. Available: http://www.nr.no/pages/dart/project_flyer_eux2010sec [Accessed: 1. Nov 2011]

[23] L. Fritsch, A.-K. Groven, L. Strand, W. Leister, and A. M. Hagalisletto, "A Holistic Approach to Open Source VoIP Security: Results from the EUX2010SEC Project," *International Journal on Advances in Security*, no. 2&3, pp. 129–141, 2009.

[24] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), Internet Engineering Task Force, Jun. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt [Accessed: 1. Jul 2011]

[25] A. M. Hagalisletto and L. Strand, "Formal modeling of authentication in SIP registration," in *Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE '08*. IEEE Computer Society, August 2008, pp. 16–21.

[26] International Organization for Standardization and ISO, "ISO/IEC 11770-4:2006: Information technology – Security techniques – Key management – Part 4: Mechanisms based on weak secrets," 2006.

[27] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1," RFC 2743 (Proposed Standard), Internet Engineering Task Force, Jan. 2000, updated by RFC 5554. [Online]. Available: http://www.ietf.org/rfc/rfc2743.txt [Accessed: 1. Jul 2011]

[28] A. Melnikov and K. Zeilenga, "Simple Authentication and Security Layer (SASL)," RFC 4422 (Proposed Standard), Internet Engineering Task Force, Jun. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4422.txt [Accessed: 1. Jul 2011]

[29] J. Undery, "Ieft draft: SIP authentication: SIP digest access authentication," IETF, Tech. Rep., Jul. 2001.

[30] C. Yang, R. Wang, and W. Liu, "Secure authentication scheme for session initiation protocol," *Computers & Security*, vol. 24, no. 5, pp. 381–386, Aug. 2005.

[31] J. Peterson, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)," RFC 3853 (Proposed Standard), Internet Engineering Task Force, Jul. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3853.txt [Accessed: 1. Jul 2011]

[32] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFCs 5746, 5878, 6176. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt [Accessed: 1. Jul 2011]

[33] C. Jennings, J. Peterson, and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks," RFC 3325 (Informational), Internet Engineering Task Force, Nov. 2002, updated by RFC 5876. [Online]. Available: http://www.ietf.org/rfc/rfc3325.txt [Accessed: 1. Jul 2011]

[34] J. Peterson and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4474.txt [Accessed: 1. Jul 2011]

[35] F. Palmieri, "Improving authentication in voice over IP infrastructures," in *Advances in Computer, Information, and Systems Sciences, and Engineering*, K. Elleithy, T. Sobh, A. Mahmood, M. Iskander, and M. Karim, Eds. Springer Netherlands, 2006, pp. 289 – 296.

[36] F. Palmieri and U. Fiore, "Providing true end-to-end security in converged voice over IP infrastructures," *Computers & Security*, vol. 28, no. 6, pp. 433–449, Sep. 2009.

[37] Y. Liao and S. Wang, "A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves," *Computer Communications*, vol. 33, no. 3, pp. 372–380, Feb. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366409002631 [Accessed: 1. Jul 2011]

[38] Y. Liao, "Secure password authenticated key exchange protocols for various enviroments," Ph.D. dissertation, Tatung University, Dec. 2009.

[39] International Telecommunication Union, "H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems," ITU-T Recommendation H.235.0, 2005.

[40] ——, "H.323 security: Framework for secure authentication in RAS using weak shared secrets," ITU-T Recommendation H.235.5, 2005.

[41] M. Spencer, B. Capouch, E. Guy, F. Miller, and K. Shumard, "IAX: Inter-Asterisk eXchange Version 2," RFC 5456 (Informational), Internet Engineering Task Force, Feb. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5456.txt [Accessed: 1. Jul 2011]

[42] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321 (Informational), Internet Engineering Task Force, Apr. 1992, updated by RFC 6151. [Online]. Available: http://www.ietf.org/rfc/rfc1321.txt [Accessed: 1. Jul 2011]

[43] L. Strand, "VoIP lab as a research tool in the EUX2010SEC project," Norwegian Computing Center, Department of Applied Research in Information Technology, Tech. Rep. DART/08/10, April 2010.

[44] "Asterisk: The Open Source PBX & Telephony Platform." [Online]. Available: http://www.asterisk.org/ [Accessed: 1. Nov 2011]

[45] "NetSED: The network packet stream editor." [Online]. Available: http://silicone.homelinux.org/projects/netsed/ [Accessed: 1. Nov 2011]

[46] X. Wang and H. Yu, "How to break MD5 and other hash functions," *IN EUROCRYPT*, vol. 3494, 2005.

[47] P. Hawkes, M. Paddon, and G. G. Rose, "Musings on the wang et al. md5 collision," Cryptology ePrint Archive, Report 2004/64, 2004.

[48] D. Eastlake 3rd and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174 (Informational), Internet Engineering Task Force, Sep. 2001, updated by RFCs 4634, 6234. [Online]. Available: http://www.ietf.org/rfc/rfc3174.txt [Accessed: 1. Jul 2011]

[49] "Twisted Matrix Labs." [Online]. Available: http://twistedmatrix.com [Accessed: 1. Nov 2011]

[50] B. Kaliski, "PKCS #5: Password-Based Cryptography Specification Version 2.0," RFC 2898 (Informational), Internet Engineering Task Force, Sep. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2898.txt [Accessed: 1. Jul 2011]

[51] S. Turner and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms," RFC 6151 (Informational), Internet Engineering Task Force, Mar. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6151.txt [Accessed: 1. Jul 2011]

[52] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104 (Informational), Internet Engineering Task Force, Feb. 1997, updated by RFC 6151. [Online]. Available: http://www.ietf.org/rfc/rfc2104.txt [Accessed: 1. Jul 2011]

[53] S. Josefsson, "The Base16, Base32, and Base64 Data Encodings," RFC 4648 (Proposed Standard), Internet Engineering Task Force, Oct. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4648.txt [Accessed: 1. Jul 2011]

[54] Y. Oiwa, H. Watanabe, and H. Takagi, "Pake-based mutual http authentication for preventing phishing attacks," *CoRR*, vol. abs/0911.5230, 2009. [Online]. Available: http://arxiv.org/abs/0911.5230 [Accessed: 1. Jul 2011]

[55] Y. Oiwa, H. Watanabe, H. Takagi, Y. Ioku, and T. Hayashi, "Mutual Authentication Protocol for HTTP," Internet Engineering Task Force, Oct. 2010. [Online]. Available: http://tools.ietf.org/html/draft-oiwa-http-mutualauth-08 [Accessed: 1. Jul 2011]

[56] T. Kivinen and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)," RFC 3526 (Proposed Standard), Internet Engineering Task Force, May 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3526.txt [Accessed: 1. Jul 2011]

[57] D. Todorov, *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*, 1st ed. Auerbach Publication, Jun. 2007.

[58] L. Zhu, K. Jaganathan, and S. Hartman, "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2," RFC 4121 (Proposed Standard), Internet Engineering Task Force, Jul. 2005, updated by RFC 6112. [Online]. Available: http://www.ietf.org/rfc/rfc4121.txt [Accessed: 1. Jul 2011]

[59] C. Adams, "The Simple Public-Key GSS-API Mechanism (SPKM)," RFC 2025 (Proposed Standard), Internet Engineering Task Force, Oct. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2025.txt [Accessed: 1. Jul 2011]

[60] L. Zhu, P. Leach, K. Jaganathan, and W. Ingersoll, "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism," RFC 4178 (Proposed Standard), Internet Engineering Task Force, Oct. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4178.txt [Accessed: 1. Jul 2011]

[61] S. Josefsson and N. Williams, "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family," RFC 5801 (Proposed Standard), Internet Engineering Task Force, Jul. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5801.txt [Accessed: 1. Jul 2011]

Eruption of Policy in the Charging Arena

Marc Cheboldaeff
Payment & Charging Solutions
Alcatel-Lucent
Ratingen, Germany
Marc.Cheboldaeff@alcatel-lucent.com

***Abstract*—In the early days of mobile Internet, bandwidth was not an issue, so price plans were quite simple, very often expressed in the form of a flat rate or "all you can eat" pricing. As long as capacity was greatly available, this was a convenient and simple way to define a tariff, both for the end-user and the service provider. With the tremendous growth of data traffic observed recently, bandwidth becomes more and more a scarce resource. Consequently, flat rate pricing leads to a minority of heavy users cannibalizing the whole resource, while being subsidized by low users. This is of course not acceptable! It is neither fair for the majority of end-users, nor profitable for the service provider. The goal of this paper is to study how service providers can grant the required quality of service to the "right" users, in other words to users who will generate revenue for their use. This will improve overall customer experience in the end, while service providers can see the return on their investment in network infrastructure by somehow "monetizing the bandwidth".**

Keywords- Rating; Charging; IMS; OCS; Policy; PCRF; PCC; QoS; QoE

I. INTRODUCTION

The Charging and Policy topics in telecommunications networks cannot be considered as two distinct topics anymore. The days where policy management was considered as a pure network internal mechanism are over. Determining the right Quality of Service (QoS) is not only a network management topic like congestion control or call gapping. Main reason is that policy decisions do not depend only on the network traffic or load at a certain point in time.

Of course, policy decisions depend also on the kind of contents that is being transmitted: high-definition videos obviously require a better QoS than poor-quality videos. Similarly, progressive downloads do not require the same policy as live streaming.

Furthermore, policy decisions depend on the type of device as well: sessions triggered by older handsets do not require the same quality as sessions triggered by latest smart phones. In addition, they depend on the underlying technology too: it might not be necessary to grant the same QoS for a data session running on a General Packet Radio Service (GPRS) network or Universal Mobile Telecommunication System (UMTS) network, than on a Long Term Evolution (LTE) network. The GPRS architecture, sometimes called 2.5G (intermediate stage between the second and third network generation) is described in [2], the UMTS or 3G architecture is described in [3], while the LTE architecture is described in [4]. The reader might refer as well to the Terminology section at the end of this paper to get the meaning of the various acronyms used.

Independently of these technical aspects, policy decisions should most importantly depend on subscriber's personal information, which encompasses business information including, but not limited to, the price plan. This is the aspect that we are going to tackle in this paper.

We shall present first the evolution of pricing schemes from fixed tariff plans to flexible offers taking into account QoS, emphasizing the importance of real-time policy and charging decisions. We shall then investigate which subscriber data is relevant in this context. Afterwards, we analyse the technical impacts in order to achieve real-time policy and charging control on an individual basis. We then design a solution, and describe its implementation. In the subsequent sections, we review other possible solutions and the position of standard bodies in this area. Finally, we address a framework aiming at changing policy in a more user-friendly way.

II. HIGH QOS AS A TARIFF OPTION ?

The old days of fixed price plans, i.e., "one size fits all", are definitely over. Nowadays, service providers tend to target specific market segments with dedicated offers in order to increase customer satisfaction and avoid subscribers' churn.

Subscribers are not expected to just accept generic tariff plans anymore, but instead they are invited to take actively part in the definition of their own "tailor-made" tariff. Often, subscribers can choose a base or default tariff, on top of which they can combine various options, each being applicable to a certain usage; for example a bucket of 100 roaming voice minutes valid 30 days, or a renewable monthly bucket of 1 Giga Byte (GB) for data traffic from the home network, etc. This is illustrated in Figure 1. Additionally, the reader who wishes to get more insight on the increased diversity of tariff options and their technological impact can refer to [5].

The left part of Figure 1 represents old tariff schemes, for example in Public Switched Telephony Networks (PSTN), where a fixed rate per time interval is usually defined, while the right part of Figure 1 depicts newer tariff schemes, where various pricing components can be combined freely.

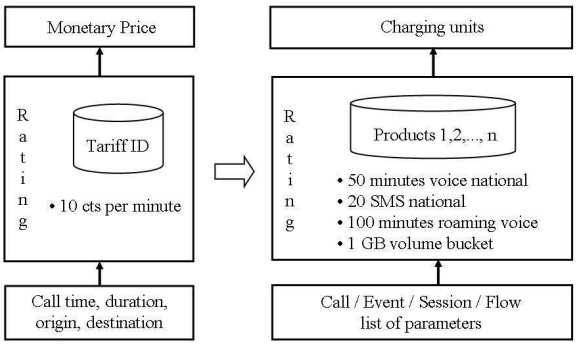


Figure 1. Evolution of tariff schemes

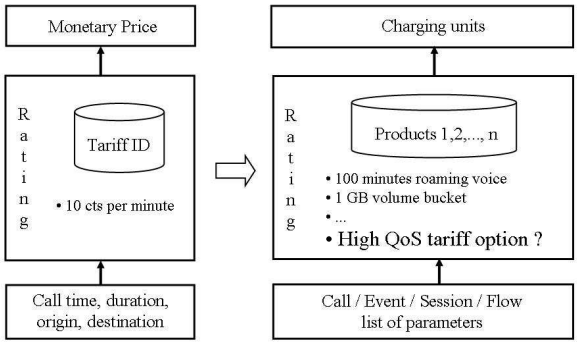


Figure 2. High QoS as a tariff option

These tariff options can be considered as different products, that the end-user may want to buy or not. For these tariff options, the charging unit is not necessarily money: once the user buys for example a bucket of data volume, he/she has a certain amount of Kilo Byte (KB) or Mega Byte (MB) on his/her account, so that a rating engine does not necessarily need to calculate a price at each session or event, but a volume amount.

Of course, these tariff options presented as products to end-customers should be easy to understand by the latter. If increased flexibility leads only to confusing complexity, there is no added-value! If a customer can easily represent for himself/herself what a number of minutes or Short Message Service (SMS) texts means, it might be more difficult to understand what it means for mega bytes! What can an end-user do with 100MB for example? How many pictures, how many mails can be retrieved? Not so easy to determine… So the options may be presented in a more user-friendly way, like an unlimited bucket applicable to Facebook or YouTube. Buckets could mix multiple traffic types too, like a Twitter bucket including Data and SMS.

In order to preserve the customer Quality of Experience (QoE), a data bucket should go hand in hand with a minimal QoS. Indeed, it would be frustrating for a subscriber to pay a certain fee to get 1GB for data traffic, and then be confronted to low speed and delays when surfing on the Internet from a mobile device! Here we see an initial correlation between the subscriber's charging profile and the subscriber's policy profile.

Furthermore, certain subscribers might be willing to pay, on top of their base tariff, which might include standard data traffic, a certain fee for having a high QoS guaranteed, independently of usage, whether cumulated or not. They just want to be sure that whenever they are going to access mobile Internet, high-speed will be guaranteed. Such a "High QoS" tariff option is represented in Figure 2.

On the right part of the picture, a "high-QoS tariff option" means that the subscriber pays a certain fee, and gets a guaranteed QoS in return. In fact, guaranteeing QoS in Internet Protocol (IP) networks, which typically work in "best-effort" mode, may not be technically achievable, but at least *prioritization* of premium users could be an option [6].

Of course, the customer should not have the feeling that in order to get a decent normal QoS, he/she has to pay more. If QoS is sold as a tariff option, it means that the obtained QoS will be beyond normal, or that this user will be prioritized over standard users.

III. IMPORTANCE OF REAL-TIME POLICY & CHARGING DECISIONS

In the previous section, we mentioned the possibility for a subscriber to buy a bucket or certain amount of units for a defined data usage. In a simple tariff offering, it might happen that the subscriber's default tariff does not cover data traffic, so that data traffic is allowed only when a data bucket is purchased by the subscriber on top of the default tariff.

In other words, when the data bucket is exhausted, data traffic should be blocked. However, the exhaustion event and thus the blocking effect might occur in the middle of an on-going data session. Such a behavior is not so user-friendly, even if notifications may be sent out for example when 80% and 90% of the bucket has been consumed already. This scenario illustrates though a basic interaction framework between charging and policy: if a data option is valid in the subscriber's profile, then traffic is allowed; if no data option is available, then traffic must be blocked.

Since a data option relates to a data usage, the charging system should track, preferably in real-time, the value of a corresponding usage counter for the subscriber. Consequently, the rule above could be expressed in the following way: if the counter value is lower than a pre-defined limit, e.g., 1GB, meaning that 1GB have not yet been consumed in the current period, then data traffic is allowed; if the counter value exceeds the limit, then data traffic must be blocked.

This is the kind of behavior that was required to be implemented as mandatory by European regulation authorities in order to control the cost of roaming data traffic and avoid "bill shocks" to subscribers. According to this regulation [7], from July 1st 2010 onward, the user should be notified when reaching 50€ of international data consumption. The rule is applicable both to prepaid and post-paid subscribers. The subscriber should be able to define a different limit if the possibility is offered by the service

provider, or opt out of this bill shock safeguard entirely. Here, we see the importance of real-time or online charging, i.e., "charging information can affect, in real-time, the service rendered and therefore a direct interaction of the charging mechanism with session/service control is required" as defined in [8].

Online charging is opposed to off-line charging, where charging takes place after usage is reported, with a certain delay, usually based on Call Detail Records (CDR) or Session Detail Records. During this delay, some chargeable traffic may occur. The cost might be quite high, even for a short time interval, in the case for example of roaming traffic. If a service provider had committed to block traffic when a certain limit is reached - eventually temporarily awaiting customer's willingness to continue - and the consumption is actually blocked when the consumption is already over this limit, then the delta cannot be legally charged to the end-customer, so it means a revenue leakage for the service provider in the end.

It should be noted that the distinction between online and off-line charging is not the same as the distinction between prepaid and post-paid. The second distinction refers to when the payment is made, whether prior to usage or afterward. However, online charging makes sense both to prepaid and post-paid subscribers, for example if end-users want to know exactly at a certain point in time how much they have spent in a billing cycle, or again, if traffic should be blocked exactly when a certain usage limit is reached.

IV. POLICY & CHARGING DECISIONS INFLUENCED BY USAGE COUNTERS

In the previous section, the decision on policy is only "allow" or "block", so actually a dedicated Policy Function is not mandatory as such in this scenario, because an online charging system is already able to cut-off a data session when a usage threshold is reached, in the same way as it can cut-off a data session when a prepaid subscriber's balance is exhausted.

Assuming now that the subscriber is entitled to make data traffic in his/her default tariff, and not only if he/she buys a data bucket on top of it, a smarter scenario would consist in throttling the data traffic when the limit is reached instead of blocking it. In other words, the subscriber would enjoy a high data speed as long as usage is charged from the data bucket, and a reduced speed when data usage is charged from the subscriber's main balance. In this case, the charging system is not the only one impacted; policy control is needed too, because QoS needs to be changed when some usage threshold is reached, and this change should happen again preferably in real-time.

Looking into the real-time aspect in more details, in fact, it would not be a big issue if QoS was not reduced in real-time, because it would be to subscriber's advantage: the subscriber could enjoy a higher QoS a bit longer than what he/she should. At the opposite, if QoS needs to be restored, or increased, it is important for the customer's quality of experience that it happens in real-time. Indeed, it would be frustrating for a user to book a new data option through an Interactive Voice Recognition (IVR) service menu, or by clicking on a pop-up window at the beginning of a download, and then have to wait some time till the QoS is actually increased.

For the sake of simplicity, we mention mainly traffic speed as attribute defining the Quality of Service in this paper, but of course QoS encompasses other attributes than speed, like delay, jitter, etc., so that a good QoS cannot be just reduced to high traffic speed. The reader, who wishes to have more insight on the definition, measurability and feasibility of a good QoS, QoE, etc., should refer to [9].

In this section, we presented use cases where the value of a volume counter should trigger a policy change. In fact, this is a good way to control heavy users, and make sure that their high usage is translated in terms of revenue for the service provider. However, we can think of other subscriber data which might trigger policy decisions too, independently of volume usage. Let us give a few examples in the next section.

V. OTHER SUBSCRIBER DATA INFLUENCING POLICY

Especially in the world of prepaid charging, subscribers are assigned a certain life cycle. A life cycle is a finite-state machine consisting of states, the transition from one state to another being triggered by the expiry of a certain time interval or by some action. For example, when a prepaid card is sent to a retailer's shop, its state might be "pre-active". When the subscriber is making the first call, after some welcome announcement is played, the card might move to a different state like "active", so that the subscriber will not hear again the welcome announcement at the next call. If the subscriber is not performing any recharges for six months, the state might move to "near expiry" and the subscriber can only enjoy limited functionality; for example, he/she might not be able to make international calls. If there is still not any recharge one month later, the card might become "inactive", etc.

In the context of policy control, we see that life cycle transitions could influence policy decisions too. For example, if a prepaid card is near expiry and the subscriber cannot make international calls anymore, maybe he/she should be throtted as well when doing mobile Internet? In this case, the transition from the life cycle's state "active" to the state "near expiry" should trigger a policy change in order to reduce the QoS. As soon as the subscriber performs the next recharge, the subscriber's state moves back to "active" and simultaneously the QoS should be set back to normal. In other words, the transition from the life cycle's state "near expiry" back to "active" should trigger another policy change, in order this time to restore the QoS.

Speaking about recharges, if a prepaid subscriber is performing lots of recharges in a short time frame, it means that he/she generates lots of revenue for the service provider. So maybe this subscriber should be paid special attention and be guaranteed a high QoS for any data session that he/she is attempting? Here again, we see that the criterion for the policy decision is not strictly usage, but the amount of recharges over a recent period.

Extending this framework, service providers can run some profiling tool on their subscribers' database or Data

Warehouse (DWH), and elaborate sophisticated policy rules based on the subscriber's charging history and behavior in order to grant the best QoS to what they consider the "best" customers. We see here a correlation between policy determination and loyalty management.

In this context, not only the network characteristics would decide how policies are granted, but subscriber profiles too. The approach would evolve from a *network-centric* approach to a *subscriber-centric* approach. The legal aspect of Network Neutrality should not be neglected here: in general, traffic should not be blocked if it originates from certain group of subscribers or from certain applications.

What are the technological impacts of this subscriber-centric approach in terms of network architecture? This is what we are going to study in the next section.

VI. IMPACTS IN TERMS OF NETWORK ARCHITECTURE

According to the 3rd Generation Partnership Project (3GPP), whatever the network access technology is, whether GPRS, UMTS, Wireless Fidelity (WiFi) [10] or LTE, data traffic transits through a packet gateway. This is represented in a simplified way in Figure 3.

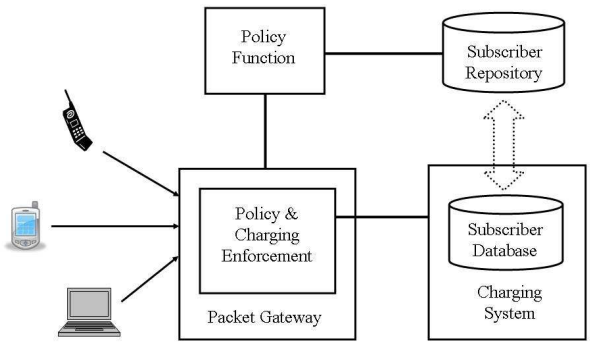


Figure 3. Data traffic from various access networks

In the case of GPRS or UMTS networks, the packet gateway might be a GPRS Gateway Support Node (GGSN); while in the case of an LTE network, it will be a Packet Data Network Gateway (PDN-GW). Besides, the latter acts as the anchor point between 3GPP and non-3GPP technologies such as WiFi or Worldwide Interoperability for Microwave Access (WiMAX) [11].

The PDN-GW provides to the User Equipment (UE) connectivity to external packet data networks by being the point of exit and entry of traffic for the UE. A single UE may have simultaneous connectivity with more than one PDN-GW for accessing multiple PDNs. The PDN-GW performs packet filtering, lawful interception and packet screening. Especially, the PDN-GW performs policy and charging enforcement, based on instruction from the policy function on one side, and from the charging system on the other side. The reader who wishes to have more insight on the Policy & Charging Control (PCC) architecture should refer to [12].

Studying in more details the reference architecture for data core network, i.e., the IP Multimedia Sub-system (IMS) standard architecture, and focusing on online charging [13], a so-called Online Charging System (OCS) relies on two databases:

- The database in the Rating Function (RF), which contains generic tariff information at service level;
- The database in the Account Balance Management Function (ABMF), which contains subscriber-specific information relevant for rating purposes.

Actually, searching the literature, an interaction between the policy decision function and external databases is mentioned in [14], but it does not relate specifically to the database of an OCS. And the dynamic mid-session interaction is not studied in detail either. A direct interaction between a so-called Policy & Control Resource Function (PCRF) and an OCS has already been studied in [15], but it restricts to an interaction of the PCRF with the Rating or Tariff Function of the OCS. It means that the policy decision might indeed depend on generic tariff rules, but it still does not depend on subscriber-specific information such as his/her current consumption or life cycle state. Moreover, reducing the subscriber's tariff information to a single tariff class ID might be restrictive given newer tariff schemes, where multiple charging options might be applied separately on top of a default tariff. The reader, who wishes to have more information about newer tariff schemes, might refer to [5]. Such charging options are amongst others usage-based discounts, subscriber bonus or individual buckets, e.g., free minutes, that the subscriber can book in addition to his/her default tariff, or that he/she gets as a reward for high consumption or recharge.

Basically, one of the functions of the OCS is to perform account balance management towards external systems through the ABMF. For this purpose, the OCS might store subscriber's pieces of information applicable for rating like usage counters. Furthermore, it might store additional information like his/her life-cycle state, e.g., validity dates, or the status of his/her valid tariff options.

According to [13], in order to support the online rating process, the Rating Function necessitates counters. The counters are maintained by the Rating Function through the Account Balance Management Function. Assuming that these counters are maintained at subscriber level, storing them together with other real-time subscriber information in the ABMF makes sense.

According to [16], in order to support the policy decision process, the PCRF may receive information about total allowed usage per user from a subscribers' repository called Subscription Profile Repository (SPR). Going further in this direction, some additional subscriber information might be relevant to the PCRF in order to determine the right policy: not only static data like an allowed usage threshold specific to a subscriber, but also subscriber's dynamic data like the value of specific counters at a certain point in time, his/her life-cycle state, or the status of his/her valid tariff options.

Storing such data in the SPR would be necessary to support scenarios like the following: as long as the subscriber consumption within one month does not exceed a

certain limit, he/she is eligible for a better QoS than once the threshold has been exceeded. Alternatively, a scenario might occur, in which a specific subscriber buys on top of his/her standard tariff an option for data traffic, so that he/she is eligible for a better policy than "normal" subscribers.

Consequently, the SPR would have to store such information as well. However, this information is still mandatory in the subscribers' database of the charging system because it might influence ratings. For example, having subscribed to a certain data option might lead to a reduced or negligible price for data traffic. Or taking the example mentioned earlier, once the subscriber consumption within one month exceeds a certain limit (not necessarily the same limit as for policy decision, but possibly tracked by the same counter!), the subscriber might enjoy cheaper rates for data traffic.

This shows that some subscriber data is meaningful both for the subscribers' repository (SPR) and the subscribers' database of the Charging System (ABMF). There could be here a kind of overlapping between the SPR and the ABMF, as the dotted arrow in the right part of Figure 3 suggests.

Replicating the information both in the SPR and in the ABMF would be an option. But this would assume efficient synchronization mechanisms between the two databases, since the number of subscribers respectively their data traffic in today's telecommunication networks might be substantial. Furthermore, the involved pieces of information consist of real-time data. If the policy should change when the subscriber's consumption reaches a certain limit, the change should happen in real-time and without delay as we saw earlier. In the same way, if the rating should change when a certain limit is reached, the change should happen in real-time too.

Duplication of databases, which store a great deal of real-time data, could increase the complexity of the implementation. If the relevant subscriber information is already present in the OCS, why should not the PCRF retrieve it directly from the OCS? This is represented by the dotted arrow in Figure 4.

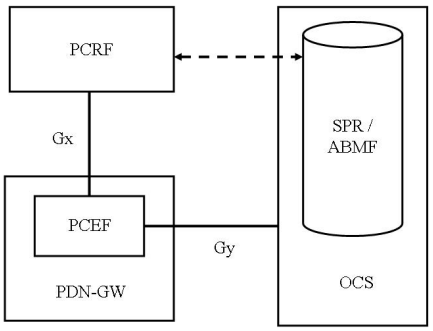


Figure 4. OCS acting as an SPR

VII. PROPOSED APPROACH

The proposed approach described for the first time in [1] consists of a framework where the PCRF and the OCS exchange in real-time subscriber information, which is necessary not only for charging, but also in order to determine the right policy. The goal is to support such scenarios where the policy might be changed in the middle of a session based on the value of some subscriber data volume counter.

The latter is stored in the OCS as master copy in any case because it is relevant for charging, in order to support offers like the following: after a subscriber has consumed 500MB within one week, he/she gets 10 free SMS, or he/she is eventually granted free-of-charge data traffic till the end of the week. Furthermore, these counters are relevant to the PCRF in order to support similar offers where, for example, the data speed is throttled once the subscriber has reached 1GB consumption within one month. In the context of the present contribution, we shall focus on volume counters. However, as mentioned earlier, it could be another piece of subscriber data, which would be relevant for the policy server, for example, the life-cycle state of the subscriber. For example, if a prepaid data card is near expiry, the surfing speed may diminish.

In the context of the implementation, which will be described in the next section, these are the values of subscriber volume counters, which should be reported in real-time from the OCS to the PCRF. More precisely, the counter values will be reported when they exceed some predefined threshold. The latter might be defined either for a certain subscribers' marketing category, or for all the subscribers in the same tariff, or individually at subscriber level. Since these thresholds might be reached in the middle of a session, the OCS might have to notify the PCRF in the middle of a data session too.

Nevertheless, the PCRF should retrieve latest subscriber information like the tariff plan information and the values of the volume counters at the beginning of the session as well, in order to determine correctly the initial policy. Alternatively, the PCRF could replicate this subscriber information, meaning again that some synchronization mechanism would have to be implemented.

In general, the message flow when a data session is established would resemble Figure 5.

In (1), the Policy & Control Enforcement Function (PCEF) asks the PCRF about the policy that should apply to the session, which is about to start for this subscriber. For this purpose, the PCRF retrieves latest subscriber information from the OCS in (2) and (3). Consequently, the PCRF can notify the initial policy to the PCEF in (4). This would happen through the Gx interface in accordance with [16].

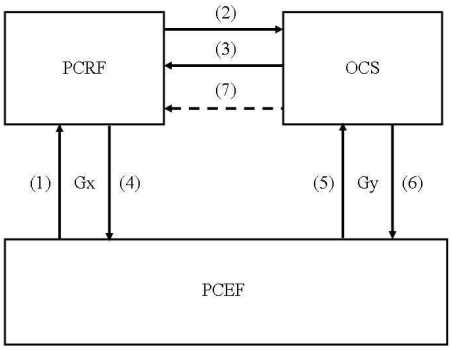


Figure 5. Message flow with PCRF/OCS interaction

Once the policy has been determined, the PCEF requests the OCS for a volume slice in (5). After checking the current subscriber's consumption, the subscriber's default tariff respectively his/her available options and current balance, the OCS allocates a slice in (6). This would happen through the Gy interface in accordance with [12]. In order to allocate the proper slice, the OCS takes into account charging-relevant thresholds, but it should take into account policy-relevant thresholds as well, because this will ensure a timely charging or policy change: as soon as the volume quota leading to the threshold will be consumed, the OCS is able to notify the PCRF. Depending on the duration of the session, there might be several volume slices requested, i.e., several messages like (5) and (6).

The arrow in (7) is represented in a dotted line because it may or may not occur during a session: the OCS would notify the PCRF only in the case that a policy-relevant threshold is exceeded during the on-going data session.

As stated above, the protocol for (1) & (4) respectively (5) & (6) is Gx respectively Gy. The protocol for (2) & (3) respectively (7) will be discussed in the next section. Since (2) & (3) respectively (7) were not fully covered by standard bodies at the time of the implementation, the most convenient protocol had to be assessed.

VIII. IMPLEMENTATION

Regarding the protocol for (7) in Figure 5, since Gx and Gy rely on Diameter [17], and Gy on Diameter Credit Control Application [18], it was decided to use Diameter Credit Control Request (CCR) Event. The reader might have noted that in (5) & (6), the OCS acts as a Diameter Server towards its client, i.e., the PCEF, while in (7) the OCS acts as a Diameter Client toward the Diameter Server, which is the PCRF in this case. As there might be several PCRF nodes, the OCS should support an N+K PCRF architecture in order to ensure a good scalability. The OCS should be able to send CCR Event messages to the PCRF nodes in round-robin way in order to ensure high-availability, meaning that the functionality can still be supported, even if one PCRF node is down.

Regarding (2) and (3), it is about the PCRF's retrieving subscriber profile data from the OCS database at the beginning of a session. Therefore, it is not really about Credit Control, nor Authentication/Accounting. Consequently, Diameter was not chosen, but Simple Object Access Protocol/eXtended Markup Language (SOAP/XML) instead, because it is a simple protocol to let applications exchange information over HyperText Transfer Protocol (HTTP) [19] in a platform-independent manner. For more information on SOAP/XML, the reader might refer to [20] and [21].

Within this framework, the following scenario has been implemented: let us assume that a subscriber is entitled a downlink/uplink speed of 768/384 Kilo bit per second (Kbps) as long as he/she has not exceeded 10MB within a month. Once he/she reaches 10MB, he/she should be throttled to 128/64 Kbps. Let us assume that at the beginning of a session, the subscriber has a consumption of 9.9MB in the current month.

Consequently, when the session is established, the PCRF communicates a QoS corresponding to 768/384 Kbps to the PCEF. In addition, the OCS allocates a quota of only 0.1MB (10-9.9) in the initial Credit Control Answer (CCA) message. That way, when the threshold of 10MB is reached, the PCRF can be notified in real-time. This is represented in Figure 6.

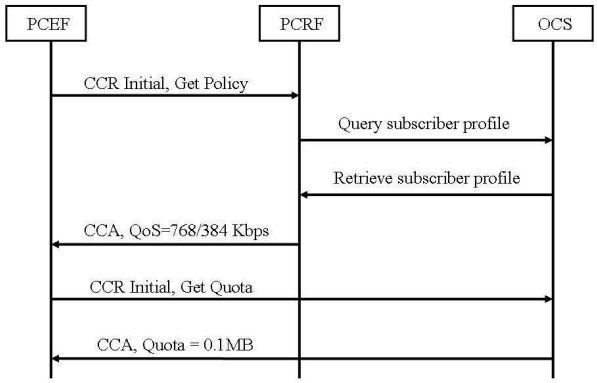


Figure 6. Initial slice granted by the OCS at session start

In case the PCRF has a local database duplicating the OCS database, and containing subscriber information that is not outdated, the query of the subscriber profile from the PCRF to the OCS may be skipped.

When the allocated quota of 0.1MB has been used up, the PCEF should request another volume quota. If the subscriber balance is sufficient, the OCS will allocate another quota so that the data session can carry on. The allocated quota might be bigger than 0.1MB this time, for example 0.5MB. Simultaneously, the OCS will notify through a Diameter CCR Event message as indicated previously that the volume threshold of 10MB has been reached for this subscriber, so that the PCRF can deduce the new QoS and notify it to the PCEF. This is represented in Figure 7.

In order to further notify the policy's change to the PCEF, the PCRF uses Diameter Re-Authentication Request / Answer messages (RAR/RAA) in accordance with [22].

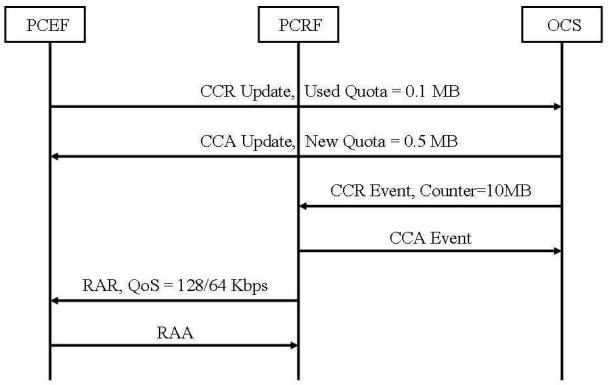


Figure 7. Mid-session notification from OCS to PCRF

In case of multiple parallel sessions, the policy change should apply to all on-going sessions. For example, let us assume that one session – Session 1 – starts when the counter value is 9.9MB. Given the threshold of 10MB, the OCS should allocate initially a slice of 0.1MB. Before the latter is used up, another session – Session 2 – starts. The OCS also allocates 0.1MB as initial slice because the counter value is still 9.9MB in the OCS database. This is represented in Figure 8.

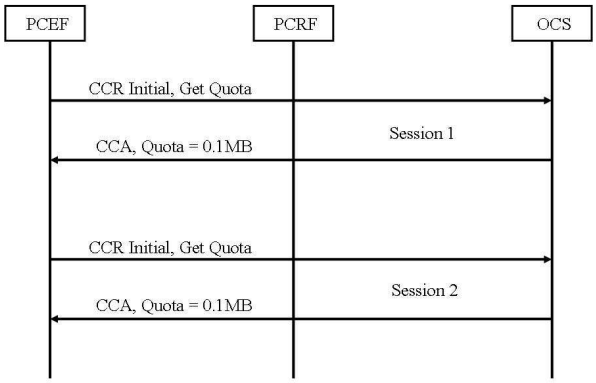


Figure 8. Initial slice for parallel sessions

As soon as the initial slice of 0.1MB of Session 1 or Session 2 is used up, the PCEF will request another slice. The OCS will grant a new slice, but it will update the volume counter value to 10MB, which should trigger the notification to the PCRF. This is represented in Figure 9, where the first session using up the 0.1MB quota is Session 1.

Consequently, the PCRF should notify the PCEF to change the QoS obviously for Session 1, but for Session 2 as well, because the volume threshold is applicable to both Session 1 and Session 2, even if the QoS change was triggered by Session 1 only.

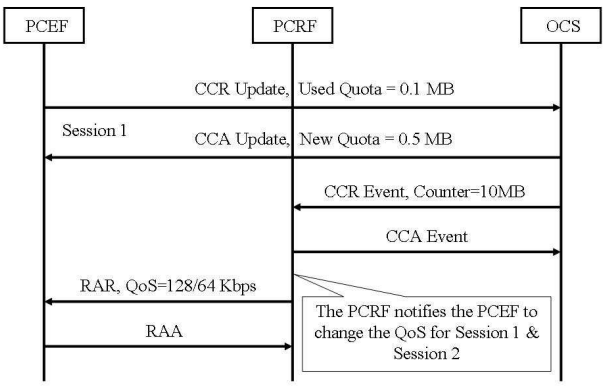


Figure 9. Mid-session notifications for parallel sessions

IX. ALTERNATIVE SOLUTIONS

We understand why PCRF and OCS should interact with each other, and we proposed a framework where they can exchange messages directly. However, interaction does not necessarily mean a direct interface between both components. The existing interfaces Gx [16] and Gy [12] could be extended to support this interaction. This is illustrated in Figure 10 (for one single session, not for two parallel sessions).

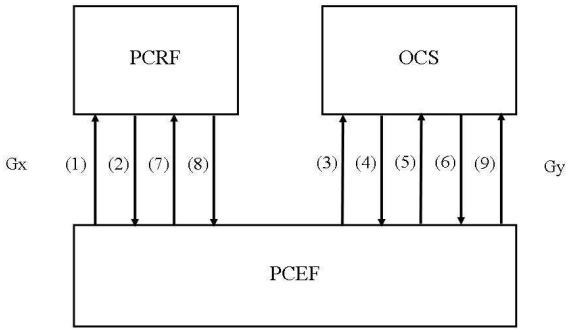


Figure 10. Flow without PCRF/OCS direct interface

The two different levels of QoS could be mapped to two different charging keys or *Rating Groups*. Assuming that the limit has not been reached yet for the subscriber in the current period, the PCRF would apply initially the first rating group together with the high QoS to the session being established in (1) & (2). In the Credit Control Request (CCR) Initial message in (3), the OCS would receive the first rating group and grant in (4) a volume quota equal or lower than the volume delta till the limit.

When the quota is used up, the PCEF notifies the OCS in a CCR Update message in (5). The Credit Control Answer (CCA) message from the OCS in (6) could indicate graceful service termination, and the Final-Unit-Action would be set to "Redirect". This way, the PCEF could forward in (7) the service termination message back to the PCRF, which could

react in (8) by returning the second charging key or Rating Group, in addition to the normal QoS information, instead of the high QoS. Upon receipt of this new Rating Group in (9), the OCS would continue granting credit to the service. Consequently, the session could continue, but not with the same QoS.

However, this dummy service termination and redirect action would have implied an extension of the existing protocols Gx and Gy, in order to support the exchange of limit-reached information from the OCS to the PCRF through Diameter Redirect.

Another alternative solution would have been to have one single system for the PCRF and OCS, as represented in Figure 11.

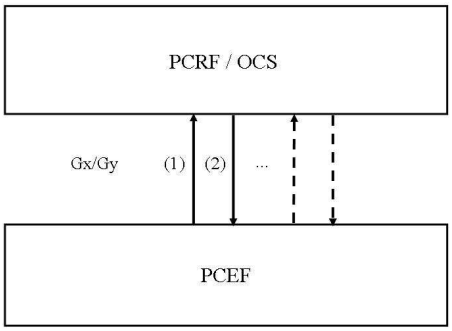


Figure 11. Integrated Policy & Charging function

As can be seen on the picture, the message flow would look quite simple. There would be successive messages like in (1) & (2), over a combined Gx and Gy interface, controlling both the policy rule to be applied and the volume quota to be granted. Such a solution may reduce the traffic between the PCEF and the integrated PCRF/OCS, and may reduce hardware and maintenance costs as well.

However, service providers, even if they are not managing policies yet on an individual basis, are already billing subscribers individually, thus they have a legacy charging system. It might be less risky to introduce a new policy function as a separate project than replacing completely the legacy charging system on top of adding a new policy control resource function.

Furthermore, keeping two separate systems for two different purposes can bring more flexibility in designing evolved policy and charging rules. Finally, it is also easier in terms of hardware and software upgrade, regarding maintenance windows, downtime, etc., which should be as short as possible, at least for a real-time charging system. A service provider might grant a high QoS temporarily for free to the complete subscriber base during a maintenance window in the night at low traffic hours, but if it grants free calls to everyone, the revenue impact is immediate, even if the number of calls is not huge.

Actually, it is a bit like having a TV and DVD player integrated in the same device. Some people may like it, but if the device is down, it means that both systems are down.

X. STANDARDIZATION

The time, when the prototype described in the 'Implementation' section was designed, goes back to the beginning of the year 2009. At this time, there was no standard regarding PCRF – OCS interaction.

Actually, IMS Release 7 introduced the concept of integrated Policy Charging & Control (PCC) architecture, with separate components for the policy function and the charging system. However, OCS – PCRF interaction was not covered at that point. Some discussion started during the course of 2009 in the context of 3GPP about "QoS and gating control based on spending limits".

A document issued end of 2009 discussed various options [23]. For the first time, a direct interface between the PCRF and the OCS was named: the 'Sy' interface. However, no conclusion was drawn at that time about which alternative would become the recommended solution.

Since then, a new version of the document [24] has been issued mid 2011, and the recommendation is now the following: "The Sy based solution where PCRF initiates Sy interaction shall be used". Main reason is that "it has the advantage of causing no increase in signaling load at the PCEF". Finally, a new technical specification dedicated to this Sy reference point was issued end of 2011 entitled "Spending Limit Reporting over Sy reference point" [25].

As the name of the specification suggests, this interface currently focuses on the exchange of counter information related to spending limits. It may be worth extending this interface in the future, in order to be able to exchange other subscriber's pieces of information, which might affect policy decisions too, like the subscriber's life cycle state, or his/her tariff options as we mentioned previously.

XI. NOTIFICATION FRAMEWORK

In the previous sections, we focused on a framework with the aim of real-time mid-session policy and charging control. In the context of the European regulation for roaming data traffic, we mentioned the possibility to notify the end-user when a certain threshold, or alternatively when a percentage of this threshold, is reached. The notification text would then explain when and why the policy is going to be changed. The possible message flow for an SMS notification is described in Figure 12.

When a session is initiated by the end-user in (1), the PDN-GW requests first from the PCRF in (2) & (3) information about the policy to be applied, and then it requests from the OCS in (4) & (5) information about the charging scheme to be applied. If it is just about notification, and not about actual policy change, the OCS can notify instead of the PCRF an SMS Center (SMS-C) like in (6), in order that the latter sends in (7) a notification to the end-user, e.g., "at this point in time, you have consumed 40€ in roaming data traffic, you are approaching the limit of 50€". The OCS provides all the information regarding the

subscriber identification and the text message that the SMS-C may need.

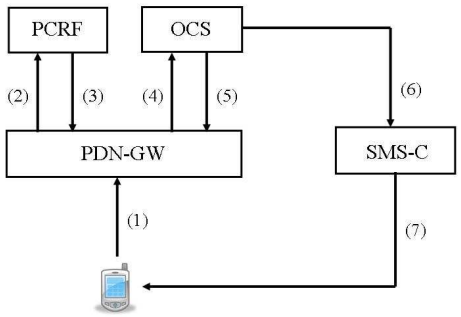


Figure 12. SMS notification in the context of Policy control

Of course, if the end-user is in the middle of a data session on his/her mobile phone, it might not be so convenient to read an SMS that is just incoming. Consequently, the user might instead be redirected to a landing page displaying notification contents. This scenario is described in Figure 13.

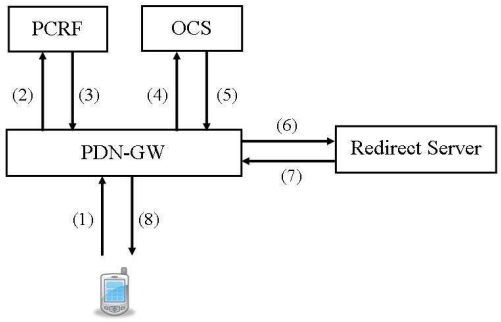


Figure 13. Redirection in the context of Policy Control

In comparison with the SMS notification, the OCS does not notify the SMS-C, but instead it sends back to the PDN-GW a Diameter Re-direct message in (5), providing the address of a Redirect Server, that the PDN-GW is able to contact in (6), in order to retrieve the landing page contents in (7), and send it back to the end-user in (8). The landing page might contain some links to invite the end-user to upgrade his price plan or his/her QoS, or to opt for a new attracting offer dedicated to data services.

Such a dialogue is definitely more user-friendly than a brutal QoS change, especially in the case of throttling. Before any policy change, and even if the ability to modify the QoS in real-time has been mutually agreed in advance between the service provider and the end-customer according to contractual terms, it might enhance the end-customer's experience to have a kind of interactive dialogue within a Web-based application.

Taking the example of throttled traffic, instead of just throttling the traffic, it might be more user-friendly to put temporarily the session on hold, and to trigger simultaneously via push mechanism the display of a pop-up window on the end-user's terminal, in order to ask him/her whether he/she agrees with additional expense or with booking a new data option. Such an interaction is illustrated in Figure 14.

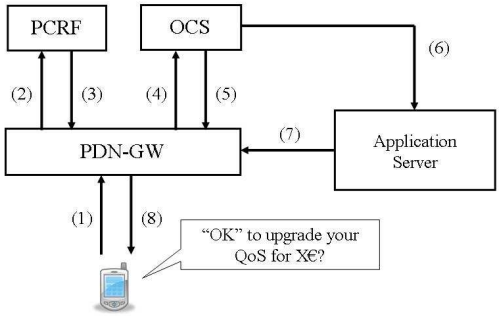


Figure 14. Pop-up window in the context of Policy Control

An application could be triggered by an Application Server (AS) on the end-user's mobile equipment. To make best use of the terminal specificities, it makes sense to have a dialogue between the AS and a specific client application running on the terminal; meaning that the AS might have to identify the type of end-user's device first, and subsequently trigger a corresponding notification interactive application on the end-user's terminal. The application on a Blackberry and on an iPhone might be different.

This application could conduct an intelligent dialogue with the end-user, asking him/her whether he/she agrees to upgrade his price and QoS plan, or accept a trial offer. Such an offer could depend on the session's context like URL. For example, if the subscriber tries to download a video from a certain video portal, he/she can be proposed a special bundle combining volume and bandwidth applicable to the video portal that the subscriber is just visiting.

In Figure 14, a user starts for example a video download in (1), and gets the standard QoS in (2) & (3). After the video download content type has been identified and authorized by the OCS in (4) & (5), the OCS notifies in parallel an AS in (6), which triggers a pop-up window on the end-user's terminal application in (7) and (8).

If the end-user answers positively, e.g., to a QoS upgrade against a certain fee of X€, eventually combined to a new "free" volume bucket, a message flow like the one represented in Figure 15 might occur. This would make an example of "in-application" smart charging.

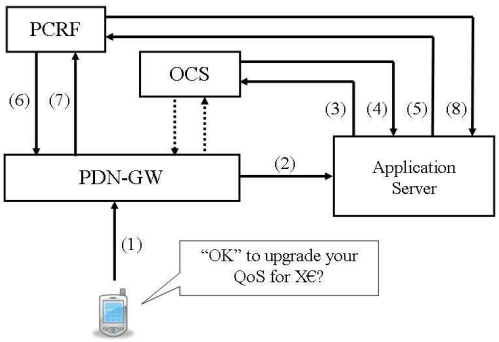


Figure 15. Application Server notifying directly a policy change

The positive answer is forwarded from the end-user's terminal to the AS in (1) & (2). Then the AS sends a debit request of X€ to the OCS in (3), and if the user's account has enough credit, the OCS will answer positively in (4). Consequently, the AS can trigger the QoS upgrade toward the PCRF in (5). The PCRF will enforce a new policy toward the PDN-GW in (6), and once the change is acknowledged by the PDN-GW in (7), the PCRF can notify the AS in (8). Since the balance of the subscriber's account has decreased, and the data tariff might have changed too, the OCS might grant a different volume quota than initially. This is represented in the dotted arrows in Figure 15. (5) and (8) would be implemented using the Rx interface in order to comply with [26].

Alternatively, the subscription for X€ to the QoS tariff option in (3) & (4) could lead the OCS's notifying itself the PCRF about the QoS change. This is represented in Figure 16.

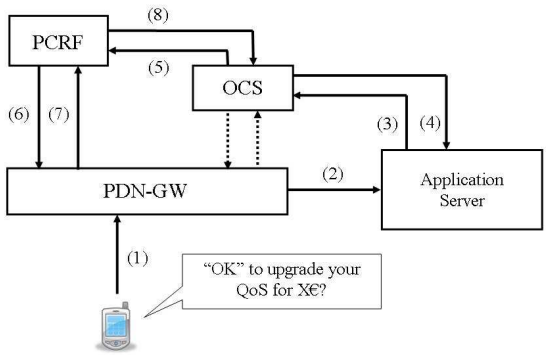


Figure 16. Application Server triggering a policy change through the OCS

The difference with Figure 15 is that the OCS is making use in (5) of the newly standardized Sy interface [25], in order to notify itself the PCRF.

XII. CONCLUSION

We studied in this paper several frameworks enabling better correlation between charging and policy control in today's telecommunication networks. This becomes a must given the tremendous increase of data traffic.

In this context, service providers face multiple challenges: the challenge to find the right balance between flexibility and complexity when proposing new tariff offers, which should combine competitive price and sufficient bandwidth for the end-user; the challenge to launch modern services consuming more network resources with the necessity to generate revenue from these services according to resource consumption; the challenge to empower new as well as existing customers to use innovative applications still preserving the overall end-user's quality of experience for the whole subscriber base; the challenge to design differentiated solutions for policy control, taking into account both standard architectures and the diversity of end-user terminals, especially smart phones and tablets.

TERMINOLOGY

3GPP	3rd Generation Partnership Project
ABMF	Account & Balance Management Function
AS	Application Server
CCA	Credit Control Answer
CCR	Credit Control Request
CDR	Call Detail Record
DVD	Digital Versatile Disc
DWH	Data Warehouse
GB	Giga Byte
GW	Gateway
GGSN	GPRS Gateway Support Node
GPRS	General Packet Radio Service
GW	Gateway
Gx	IMS reference point between PCEF & PCRF
Gy	IMS reference point between PCEF & OCS
HTTP	Hyper Text Transfer Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IVR	Interactive Voice Recognition
KB	Kilo Byte
Kbps	Kilo bit per second
LTE	Long Term Evolution
MB	Mega Byte
NN	Network Neutrality
OCS	Online Charging System
PCC	Policy & Charging Control
PCEF	Policy & Control Enforcement Function
PCRF	Policy & Control Resource Function
PDN	Packet Data Network
PSTN	Public Switched Telephony Network
QoE	Quality of Experience
QoS	Quality of Service
RAA	Re-Authentication Answer
RAR	Re-Authentication Request
RF	Rating Function
Rx	IMS reference point between AS & PCRF
SMS	Short Message Service
SMS-C	SMS Center
SOAP	Simple Object Access Protocol
SPR	Subscription Profile Repository

Sy IMS reference point between OCS & PCRF
TV Television
UE User Equipment
UMTS Universal Mobile Telecommunications System
WiFi Wireless Fidelity
WiMAX Worldwide Interoperability of Microwave Access
XML eXtended Markup Language

ACKNOWLEDGMENT

The author would like to thank Hongwei Li, Renée Fang, Jessica Han, Angelo Lattuada, Justin Bayley, Andy Wood, and Mark Bryant from Alcatel-Lucent, Michael Leahy from Openet, Dion Pirnaij, Guido Gillissen, and Ton van Boheemen from Vodafone, Steven Cotton from the TM Forum, Val Korolev from OpenCloud.

REFERENCES

[1] M. Cheboldaeff, "Interaction between an Online Charging System and a Policy Server", Tenth International Conference on Network (ICN), January 23-28, 2011. France. IARIA.

[2] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "General Packet Radio Service (GPRS); Service description", 3GPP TS TS 23.060, Release 10, June 2011.

[3] Y.-B. Lin, A.-C. Pang, Y.-R. Haung, I. Chlamtac, "An All-IP Approach for UMTS Third-Generation Mobile Networks", IEEE Network Magazine, September/October 2002

[4] 3rd Generation Partnership Project, 36 series, "LTE (Evolved UTRA) and LTE-Advanced radio technolgy", http://www.3gpp.org/ftp/Specs/html-info/36-series.htm [retrieved: January 3rd, 2012]

[5] M. Cheboldaeff, "Service Charging Challenges in Converged Networks", IEEE Communications Magazine, January 2011

[6] H. Zhou, K. Sparks, N. Gopalakrishnan, P. Monogioudis, F. Dominique, P. Busschbach, and J. Seymour, "Deprioritization of Heavy Users in Wireless Networks", IEEE Communications Magazine, October 2011

[7] Europe's information society, "Countering data roaming bill shocks", http://ec.europa.eu/information_society/activities/roaming/regulation/index_en.htm [retrieved: January 3rd, 2012]

[8] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "Charging management; Charging data description for the IP Multimedia Subsystem (IMS)", 3GPP TS TS 32.225, Release 5, March 2006.

[9] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What is It Really?", IEEE Communications Magazine, April 2011

[10] Institute of Electrical and Electronics Engineers (IEEE), Information technology, Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications, IEEE Std 802.11

[11] Institute of Electrical and Electronics Engineers (IEEE), Information technology, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, IEEE Std 802.16

[12] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "Policy and Charging Control (PCC) architecture", 3GPP TS 23.203 v11.2.0, Release 11, June 2011.

[13] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "Online Charging System (OCS): Applications and Interfaces", 3GPP TS 32.296 v11.0.0, Release 11, June 2011

[14] R. Good, and N. Ventura, "Application driven Policy Based Resource Management for IP multimedia subsystems", 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom), 2009

[15] T. Grgic, K. Ivesic, M. Grbac, and M. Matijasevic, "Policy-based Charging in IMS for Multimedia Services with Negotiable QoS Requirements", 10th International Conference on Telecommunications (ConTEL), 2009

[16] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, "Policy and Charging over Gx/Sd Reference Point", 3GPP TS 29.212 v11.1.0, Release 11, June 2011

[17] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol", IETF RFC 3588

[18] H. Hakala, L. Mattila, J-P. Koskinen, M. Stura, and J. Loughney, "Diameter Credit Control Application", IETF RFC 4006.

[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", IETF RFC 2616

[20] SOAP Version 1.2, World Wide Web Consortium (W3C) Recommendation, http://www.w3.org/TR/soap12-part1/ [retrieved: January 3rd, 2012]

[21] World Wide Web Consortium (W3C), http://www.w3.org/standards/xml/ [retrieved: January 3rd, 2012]

[22] P. Calhoun, G. Zorn, D. Spence, and D. Mitton, "Diameter Network Access Server Application", IETF RFC 4005

[23] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "Study on Policy solutions and enhancements", 3GPP TR 23.813 v0.1.0, Release 10, November 2009

[24] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "Study on Policy solutions and enhancements", 3GPP TR 23.813 v11.0.0, Release 11, June 2011

[25] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, "Policy and Charging Control: Spending Limit Reporting over Sy reference point", 3GPP TS 29.219 v1.0.0, Release 11, November 2011

[26] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, "Policy and Charging Control over Rx reference point", 3GPP TS 29.214 v11.1.0, Release 11, June 2011

The Secure Access Node Project: A Hardware-Based Large-Scale Security Solution for Access Networks

Jens Rohrbeck, Vlado Altmann, Stefan Pfeiffer,
Peter Danielis, Jan Skodzik, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
Rostock, Germany
{*jens.rohrbeck;dirk.timmermann*}*@uni-rostock.de*

Matthias Ninnemann, Maik Rönnau
Nokia Siemens Networks GmbH & Co. KG,
Broadband Access Division
Greifswald, Germany
{*matthias.ninnemann;maik.ronnau*}*@nsn.com*

***Abstract*—Providing network security is one of the most important tasks in todays Internet. Unfortunately, many users are not able to protect themselves and their networks. Therefore, a novel security concept is presented to protect users by providing security measures at the Internet Service Provider level. Already now, Internet Service Providers are using different security measures, e.g., Virtual Local Area Network tags, MAC limitation, or MAC address translation. The presented approach extends these security measures by three hardware-based security subsystems. A firewall engine controls the header of Ethernet frames, Internet packets, and the next following protocols. Furthermore, a Web filter module disables access to violent and child pornography Web content. The third subsystem is a Bloom filter-based deep packet inspection engine to observe the payload after the protocol header. Based on deep packet inspection, it is possible to detect network intruder. A firewall, a Web filter as well as a network intrusion detection system, at the ingress of the network, offer security measures to all connected users, especially to users with limited IT expert knowledge. Each of the mentioned systems has a powerful packet classification engine and a high speed rule set engine used by the firewall to find specific rules for each frame. The rule set engine does not need expensive content addressable memory. All described filter modules as well as the packet classification engine and the rule set engine are developed in reconfigurable hardware. Thus, rule updates and adjustments to the hardware are easy to realize. Adjustments can be made only by the Internet Service Provider administrator. Consequently, the security system itself is secured against attacks from users and from the network side. This novel security approach allows for the protection of up to 32,000 Internet users in wire speed. Furthermore, the prototype system is able to process network traffic at wire speed.**

Keywords-Internet Security; Access Network; Hardware Firewall; Hardware Web Filter; Hardware Intrusion Detection.

I. INTRODUCTION

Firewalls and anti-virus programs provide basic protection for Internet-enabled devices. Normally, these security measures are installed on computers of users. But installing security measures at the users' side has two serious drawbacks. Firstly, the threat detection is done on the target machine. Secondly, the users must install, upgrade, and maintain these security measures without professional support. Other measures such as a Web filter and a deep packet inspection engine like snort are often not installed and require additional maintenance. In addition, the majority of Internet users is missing the necessary expertise to configure their security software so that it provides optimal protection. Furthermore, because of negative experiences like phishing attacks targeting online banking, many users have lost their confidence in online services and the Internet itself. Therefore, it is mandatory to disburden respectively to support users in issues of Internet security.

A trustworthy place for the placement of security measures is the ingress of the network — the access network. Each user, referred to as subscriber by Internet Service Providers (ISPs), is connected to the Internet through the access network. The access network itself consists of access nodes (AN). As ANs are transparent for subscribers, these components are safe from, e.g., Denial of Service Attacks. To reestablish the subscribers' confidence into the Internet and moreover, to even protect the Internet itself, it is useful to establish additional security services at ANs. With these additional security features, two objectives can be achieved. On the one hand, the subscriber is offered a higher security service without the need to care about security measures himself. On the other hand, outgoing traffic from subscribers can be verified. Thus, the network is protected as well.

Additionally, if the subscribers and the global network are protected, services inside access network like Dynamic Host Configuration Protocol (DHCP) and Domain Name Service (DNS) are protected as well. Although an antivirus program is mandatory to protect a network, the presented system excludes this protection measure. Firstly, antivirus programs are for free. They can offer a good protection with default settings and the maintenance of such programs is very simple because they update themselves autonomously. Secondly, the resources to monitor antivirus signatures and other maleware signatures exceed the available resources of an AN. The access network, as the ingress of the network, aggregates the traffic of many single Internet connections. So, traffic

rates from some Mbit/s up to more than 100 Gbit/s have to be processed — both in up- and downstream.

Furthermore, the goal is to support rules for up to 32,000 connections. Due to hardware restrictions, a renouncement of connection tracking and the control of protocols' communication sequences is necessary. The prototype referred to as Secure Access Node (SecAN) presented in [1] extends the currently available security measures on an AN by a packet filtering firewall, Web filtering, and intrusion detection system (IDS). Thereby, this functionality moves from the subscriber to the ISP.

To fulfill these tasks under the conditions described, a very powerful packet classification [2] and packet processing are required. Due to these requirements, pure software solutions are not applicable. Therefore, SecAN is a hardware solution on a XILINX evaluation board with a FX70T Field Programmable Gate Array (FPGA). This solution do not use CAM memory. Already for 224 connections (these approximates ca. 0.7 % of all connections), over 90 % of available block ram ressources or 23 % of slice register would be needed. Without using CAM, the solution is able to control traffic at wire speed.

Through the development and deployment of the security measures, the operators will have financial effort. But at the same time, the ISP will have a benefit against ISP which do not use this security solution. He can take up new security measures in its portfolio and so he can offer network protection for subscribers without the need to care about it. This creates a new service, whereby the funding is guaranteed through subscribers.

Briefly summarized, the main contributions of this paper are the following:

- The presented prototype is a novel hardware solution of a packet filter, Web filter, and an intrusion detection system placed onto an access network. All filter subsystems can be used independently or together as a closed system.
- This solution is able to control traffic in up- and downstream direction simultaneously without packet loss. Thus, it can protect the connected subscribers and the network itself.
- SecAN uses a set of 10 frame parameter to classify connections individually at which the number of configurable rules is not limited as described in [2].
- As target platform, a XILINX evaluation board with an FX70T FPGA is used. Although no CAM is used, SecAN is able to control traffic in wire speed, at least with 1 GBit/s. Because the speed is module dependent the system is able to achieve up to 4.8 GBit/s.
- The structure and functionality of all developed modules as well as the used resources and the reached speed are described in detail.

The remainder of this paper is organized as follows: Section II describes security measures available in the access network today. In Section III, the SecAN's hardware architecture is presented. Here the various modules and their functions are explained. Following, in Section IV, the generation of different clocking domains is described. Before the paper concludes in Section VI, the developed software solution for flexible configuration of the hardware is presented in Section V.

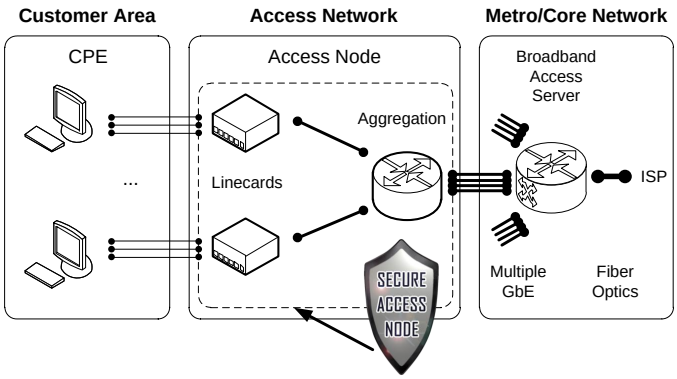


Figure 1. Access Network containing ANs. The Secure Access Node is an extension of an AN.

II. Security Measures In The Access Network

Each subscriber achieves access to the Internet through the access network. Access networks comprise subscriber premise equipments (CPEs) and access nodes such as Digital Subscriber Line Access Multiplexers (DSLAMs). The latter usually consists of linecards and aggregation cards as shown in Figure 1. While aggregation cards provide high-bandwidth interfaces towards metro or core networks, linecards aggregate various subscriber lines.

Although the network ingress is transparent for traffic from and to subscribers, ISPs have to protect the access network. Today, security measures mainly include passive measures on OSI layers II and III [3], [4]. For example, ISPs are using security measures like:

- Port isolation - subscriber may not communicate via an AN
- MAC antispoofing - a Source MAC address is allowed only at one port at a time
- MAC address limitation - to limit the number of MAC addresses per port
- MAC address translation - subscribers MAC address is translated to an ISP MAC address
- VLAN tags - to separate subscriber and services
- IP antispoofing - only the IP address - assigned by the ISP - in combination with the requested MAC address, is allowed Source IP and Source MAC pair at a special port

To ensure a minimum necessary level of security when connecting to the Internet, the already introduced security

measures must be integrated into the access area by means of the Secure Access Node.

III. SecAN - Architecture

A. Hardware Overview

To emulate the SecAN on an AN, a XILINX ML507 evaluation board with an FX70T FPGA [5] is utilized. Thereby the FPGA is the main component and is typically used on linecards. Furthermore, the 1 MB Static Random Access Memory (SRAM) and the 512 MB large Double Data Rate Synchronous Dynamic Random Access Memory (DDR2-SDRAM) is utilized. To control traffic in upstream and downstream direction, two 1 Gigabit Ethernet transceivers are used as well.

B. The System In General

Each Ethernet transceiver of the evaluation board is able to process data with 1 Gbit/s. That corresponds 16 bits per clock cycle which have to be processed. To avoid the discarding of any uncontrolled data frame and due to the internal delay during frame processing, an increasing of the internal bandwidth to 32 Bit/cycle is necessary.

Generally, the SecAN system is divided into two hardware groups. The inner hardware group is the actual filter core and consists of packet classification engine (PCE), rule set engine (RSE), and packet processing engine (PPE) and is used for processing of network traffic. Furthermore, the outer hardware group consists of two Ethernet transceiver, a frame multiplexer, a frame demultiplexer, and a configurator and is used for receiving and sending Ethernet data as well as the receiving of configuration information. These components consume the resources and achieve the speed as shown in Table I.

Modul	Flip Flop / LUT Slices	BRAM	Speed
SecAN firewall	1993/1921 ($\widehat{=}$5 %/($\widehat{=}$5 %)	8 ($\widehat{=}$6 %)	173.273 MHz ($\widehat{=}$5.54 Gbit/s)

Table I
Resources and Speed for all SecAN hardware modules which receive and send external data

The used resources are very low. Thus many resources remain for the main task - the packet processing. Nevertheless, the outer hardware group limits the maximum attainable speed to 5.5 GBit/s.

C. Configuration and Frame Processing In General

- Before the system can process traffic, it must be configured. The components that need to be configured are the PCE, RSE, Web filter, and the DPI control stage. All configuration data is solely written to the hardware and read from it by the ISP. The configuration flow is shown by dashed arrows in Figure 2.
- After configuration, frames reach the inner system. The frame multiplexer receives and buffers Ethernet frames from the evaluation board's interfaces and chooses the next frame to be processed by the PCE. The PCE separates flow data from the frame and requests the particular rule set from the RSE. Each rule set is a particular collection of rules, which are necessary to evaluate a frame. After identifying the right rule set, it has to be forwarded to the PCE. When the rule set reaches the PCE, the rule set, the data frame, and collected frame parameters have to be sent in the direction of the PPE - to the control stages. In the control stages (CS), the rules from the rule set are applied. The CSs are able to discard or forward frames or replace frame values like IP addresses. If a frame is not discarded it leaves the PPE towards to the frame demultiplexer. The frame demultiplexer discards the rest of the rule set and the frame parameter set and forwards the frame to the right output interface.

D. Hardware Configuration

An initial configuration as well as an on-the-fly configuration can be done but before the hardware components are not configured, no frames traverse the SecAN. The configurator module, shown in Figure 2, receives the writing configuration data from the configuration software. For control reasons, configuration data can be read via the configurator.

During configuration, no new frames can be process. For that reason, two possibilities can be applied. Either, all data traverses the system uncontrolled or all data is blocked. Because a security system should not be bypassed, it is better to block all data during the configuration phase.

The configuration starts with a 'start of configuration' information. All internal processes are stopped and all new frames are rejected. Following, the configuration data are received and distributed by the configurator. The last configuration data is a 'end of configuration' information. After that, the configuration process is finished and data frames can be processed by the SecAN.

Configuration data is provided to the appropriate modules by the configurator. This data has a type-length-value layout.

- Type is an 8 bit field and determines the component of the hardware to be addressed. Each component has two valid type values: one for writing configuration data and one for reading configured data.
- Length is an 8 bit field and represents the number of configuration data bytes. A maximum of 256 bytes plus configuration header can be configured.
- The actual configuration data is contained in the value part. All components are assigned specific configuration values.

Resources and Speed: Table II shows the required resources as well as the speed of the configurator module based on an Virtex 5 FX70T FPGA. Just as the modules

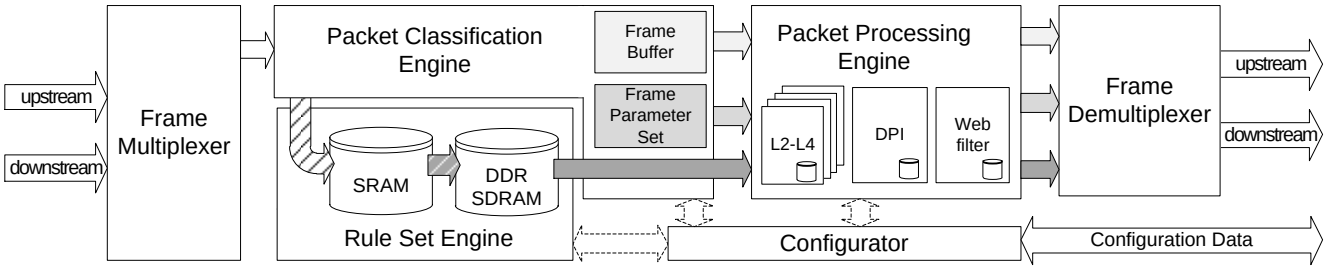


Figure 2. Block diagram of the Secure Access Node

for receiving and sending data the configurator modul is not among the core components. The hardware requirements are very low for this module. However, the maximum processing speed cannot be achieved because the transceiver modules limit the processing speed.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Configurator	851/1300 ($\widehat{=}$2 %/$\widehat{=}$3 %)	1 ($\widehat{=}$1 %)	195.236 MHz ($\widehat{=}$6.25 Gbit/s)

Table II
RESOURCES AND SPEED FOR THE CONFIGURATOR

E. The Frame Processing Flow

The frame processing flow is shown by shaded arrows in Figure 2. If the PCE is not busy it has to receive and classify the frame. The frame multiplexer selects a frame from the internal buffer with the highest fill level. After frame classification is finished, the RSE searches for an individual rule set for each frame. Rules of the rule set are applied to the frame. If the frame is not discarded by the PPE due to the rules it is sent to the correct output interface by the frame demultiplexer.

1) Packet Classification Engine: The PCE fulfills several tasks. Briefly summarized, it buffers the whole frame and creates an special parameter set. In this parameter set, different frame parameter and special information for the Web filter and the IDS control stage are saved. Later, all control stages use this special parameter set to accelerate the decision whether the frame is accepted or dropped. Furthermore, it separates a distinct key from the frame - the Flow ID. The creation of the Flow ID depends on the receiving direction of the frame. Therefore, the PCE has got additional information from the frame multiplexer. This flow ID is the basis for a connection-specific rule set. Moreover, for a quick search for the rule set, the PCE determines the memory address.

Creation of the Flow IDs: During the configuration, the PCE has got two so called Flow ID trigger. These triggers describe, which of the frame parameters are necessary to classify a frame. It is possible to set a new trigger by reconfiguration on the fly. One trigger is used for upstream frames and the other trigger for downstream frames. With one of these trigger in combination with the received information of the frame multiplexer, the PCE is able to create a directional unique Flow ID. Additionally, a Flow ID valid flag signals the validation of the unique Flow ID. That means, if at least one of the necessary frame parameter is not be available the Flow ID is invalid. As a result, a default rule set must be ordered. Often packets are classified by five packet header fields: Both IP addresses and port numbers, and transport layer protocol [6]–[8]. To achieve a higher degree of flexibility during frame classification both MAC addresses, up to 2 VLAN tags, and the Ether type field are added to the frame parameter set.

Calculation of a memory address: Similar projects like [6]–[8] have very short Flow IDs and use CAM or bloom filter approaches to increase the lookup performance as suggested in [2]. So, they do not need address calculation for memories. If CAM is used this either results in high acquisition costs or a disproportionately high consumption of hardware resources. Furthermore, bloom filter approaches are not able to calculate a memory address for a rule set. Other hardware friendly approaches are tree based. Although a logarithmic time complexity promises good results, but a quicker working solution is needed. Therefore, a hardware-friendly CRC hash algorithm is used. This kind of algorithms compresses an input vector to an output vector, at which the length of the input vector is usually higher than the length of the output vector. In this case, the output vector - the hash value - is used as memory address. Although CRC is not a perfect hash function, the big advantage is that CRC provides uniformly distributed binary vectors. The aspect of collision resolution is discussed in Section III-E2.

After the Flow ID is completely composed and the hash value is calculated, a request for the rule set is performed by the RSE. Together with the Flow ID, the PCE has been set a ”Flow ID valid” flag. The RSE decides with the help of the ”Flow ID valid” flag, which rule set should be requested - the individual rule set from the calculated address or the default rule set. If the individual rule set is received from the RSE, the frame, rule set, and parameter set is sent towards the PPE. Because only the parameter set is available in the

PPE with the first cycle, a comparison with rule parameter can be done before the proper frame data reaches at the PPE.

Generation of log data: Particularly in the access network, hardware solutions are very expensive. Therefore, the optimal use of existing resources is required. Thus, the generation of log data is extremely important. For that reason, the PCE is able to capture log data. Thereby, all frames and all bytes are counted. In combination with the captured log data of the filter modules statistics can be created about the traffic. Thus, new rules can be created and existing rules can be optimized. In this way, existing hardware resources are saved.

Resources and Speed: Table III shows the required resources as well as the speed of the PCE based on an Virtex 5 FX70T FPGA.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Packet Classification Engine	593/865 (≙2 %/≙2 %)	5 (≙4 %)	165.893 MHz (≙5.31 Gbit/s)

Table III
RESOURCES AND SPEED FOR THE PACKET CLASSIFICATION ENGINE

2) Rule Set Engine: For the rule set search, a two-stage approach with a hardware-gentle compression method is used. Firstly, the mapping between Flow ID and the rule set is done in a sufficiently large SRAM memory. Two clock cycles after the application of an address to the SRAM, the date is available. Secondly, very large rule sets have to be stored in DDR2-SDRAM. To increase speed when reading and writing memory information, the given memory controller [9], which does not use the evaluation board's internal bus systems, has been extended. The rule set is sent towards the PCE and forwarded together with the frame and frame parameter set to the PPE.

Flow ID Mapping in SRAM: When the packet classification is finished, the RSE gets the Flow ID, the CRC hash value of the Flow ID, and the validation information of the Flow ID from the PCE. Should the "Flow ID valid" bit indicate an invalid Flow ID, the memory address for the default rule set is selected and a request to the DDR2-SDRAM is sent. Normally, a specific SRAM entry has to be searched. As described before, CRC is not a perfect hash function and so the aspect of collision resolution is extremely important. For this reason we have developed two strategies. Firstly, a linear collision resolution has been tested. Therefore, thirty runs have been made, each with 32,000 valid Flow IDs and we have calculated the CRC hash values of all Flow IDs and ordered it after collisions. In the 1 MB of SRAM, 43,690 of the linking entries can be saved. All Flow IDs, which do not cause collisions, are stored. After that, all entries, which cause one collision, are stored as well. Whenever a memory location is found, which is occupied, a linear search for a free memory location begins.

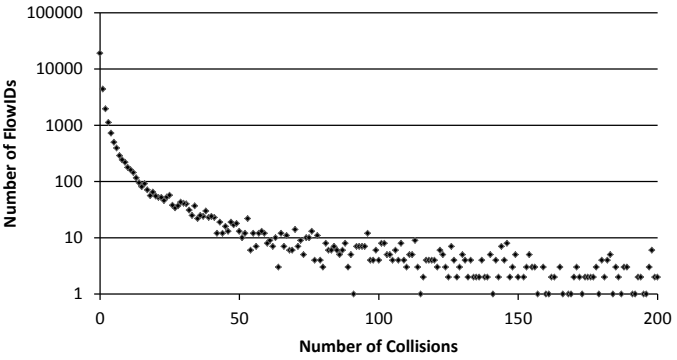


Figure 3. Linear Collision Resolution for Flow IDs. For the y-Axis, a Base 10 Logarithmic Scale is Used.

Figure 4. Indexed Collision Resolution for Flow IDs.

Although the SRAM is large enough, in the worst case, up to 1617 collisions are reached (as shown in Figure 3). This solution undoes the benefits of CRC.

The second strategy uses indexed memory entries. That means, after all entries without collision have been saved, all remaining entries are stored in free memory slots. Now, each memory entry contains a collision information and a SRAM address information. If the collision information indicates another entry on the same SRAM position the following SRAM address information is used. This procedure is repeated until the correct entry is found. If the desired Flow ID has not been found but the collision information indicates no further collision, the address of the default rule set is used. This case should not occur as all of the correctly collected Flow IDs are known in advance.

In approximately 29 % of all test cases, the calculated CRC value corresponds directly with the searched memory address and in about another 49 % the searched address is achieved after one collision. In the worst case, 6 collisions occur. The result is significantly better than a logarithmic search tree, which needs up to 16 steps to search all 43,690 possible memory entries. Figure 4 shows the test results.

Rule Set Order in DDR2-SDRAM: The DDR2 memory stores all available rule sets. A rule set is a collection of individual rules, which are used by the firewall filter stages.

By definition, a rule set has a maximum length of up to 1,024 bytes. Thus, up to 262,144 rule sets can be stored in the available DDR2 memory. After the DDR2 controller has got the DDR2 address information from the SRAM controller, approximately 27 clock cycles have to be waited until the first data arrives at the DDR controller. The rule set is preceded by a head, sent towards the PCE, and forwarded together with the frame and frame parameter set to the PPE. The self-developed SRAM controller and the extended DDR controller as well as increasing the internal bandwidth will guarantee maximum throughput for the entire SecAN system. Table IV shows the required resources as well as the speed for the SRAM controller module and the DDR2 controller modul based on an Virtex 5 FX70T FPGA.

Modul	Flip Flop / LUT Slices	BRAM	Speed
SRAM controller	470/1048 (≙2 %/≙3 %)	6 (≙4 %)	179.727 MHz (≙5.75 Gbit/s)
DDR2 controller	2807/1842 (≙6 %/≙4 %)	6 (≙4 %)	179.340 MHz (≙5.74 Gbit/s)

Table IV
RESOURCES AND SPEED FOR THE SRAM CONTROLLER AND DDR2 CONTROLLER

3) Packet Processing Engine: The PPE is responsible the for control and evaluation of the data stream and consists of three central components. In addition to a classic packet filtering, a Web filter and a signature recognition engine have been implemented. Each of the three components aims at protecting subscribers from unauthorized access from the network side and suppresses attacks from subscribers on the network.

Packet Filtering: The packet filter is divided into 12 control stages, so each CS has only a marginal role to fulfill. On OSI layer 2, a source MAC and a destination MAC CS has been developed as well as separate CS's for both possible VLAN tags and the ethertype. Furthermore, to control IPv4 parameter, a CS for the OSI layer 4 protocol and 2 separate CS's for both IP addresses have been designed. Moreover, 2 CSs control OSI layer 4 port information. Last but not least, a MAC address translation as well as IP antispoofing CS has been developed. Due to the modularity of the PPE, the whole system is very flexible and efficiently to extend. Figure 5 shows on the one hand the outer structure of all CS's and on the other hand the design structure for the OSI layer 4 port CS's.

The design of Figure 5 is separated into two parts. In the upper part, the Ethernet frame is processed and in the lower part the logic for the log data is shown.

The Ethernet frames pass all CS's one after the other. Simultaneously with a frame, the belonging parameter set as well as the rule set reaches at the CS. Rule sets can have one or more rules whereby each rule has a type-length-value composition similar to configuration data. To speed up the

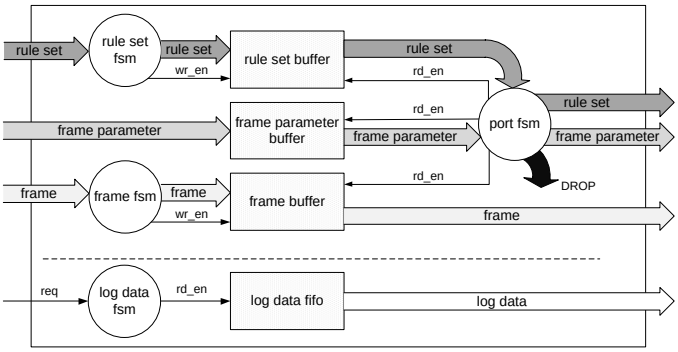


Figure 5. Outer and inner design of the port control stages

processing, the rules in the rule set have been configured in the same order as the CS's are arranged. Because each applied rule is removed, each CS has to look only at the first rule. Moreover, each CS has a unique identifier. If the type of the first rule does not equal the CS's ID, the frame, rule set, and parameter set are forwarded to the next stage. Otherwise, the rule is processed by the CS. Each CS compares the data from the rule with the data of the parameter set. Because the whole parameter set is available in the first cycle of a new frame, the lookup increases the processing speed, especially for OSI layer 4 values. In case of a match, the rule action has to be executed. That is, the frame, rule set, and parameter set can be discarded or forwarded as well as frame parameter can be changed. After processing, the applied rule will be removed and the next CS is able to look at the first position of the rule set. According to the principle of divide and conquer, the rule set finit state machine (FSM) and the frame FSM receives and buffers incoming data. The port FSM analyzes the rule set, applies the rule and forwards or discards the buffered data.

If the rule action requires the discarding of the received data, there are two counter values, which have to be increased. The drop frame counter counts the discarded Ethernet frames and the drop byte counter counts all discarded bytes. Both values are stored in a 32 bit register. Additionally, the reason for discarding is stored in a FIFO buffer, e.g., the existing and the required MAC addresses. If the buffer reaches the maximum fill level, the oldest date is replaced by the current. Prospectively, a log data collector will request the stored data from all CS's. The log data FSM will get a request signal and send all captured log data in the direction of the log data collector.

Table V shows the required resources as well as the speed for all described CSs based on an Virtex 5 FX70T FPGA.

Throughput and Resources of the SecAN Subsystems: The SecAN project consists of three subsystems. The SecAN packet filter firewall subsystem, consists of two Ethernet interfaces, two receiving and sending synchronization frame buffers, a frame multiplexer and a frame demultiplexer, the

Modul	Flip Flop / LUT Slices	BRAM	Speed
Source MAC and Destination MAC CS	701/657 (≙2 %/≙2 %)	1 (≙1 %)	173.322 MHz (≙5.55 Gbit/s)
Inner VLAN and Outer VLAN CS	623/689 (≙2 %/≙2 %)	1 (≙1 %)	176.444 MHz (≙5.65 Gbit/s)
Ethertype CS	688/710 (≙2 %/≙2 %)	1 (≙1 %)	180,101 MHz (≙5.67 Gbit/s)
OSI L4 Protocol CS	695/722 (≙2 %/≙2 %)	1 (≙1 %)	179.556 MHz (≙5.75 Gbit/s)
Source IP and Destination IP CS	699/695 (≙2 %/≙2 %)	1 (≙1 %)	172.311 MHz (≙5.51 Gbit/s)
OSI L4 Source and Destination Port CS	593/865 (≙2 %/≙2 %)	5 (≙3 %)	182.815 MHz (≙5.85 Gbit/s)
IP Antispoofing CS	675/665 (≙2 %/≙2 %)	1 (≙0 1%)	184.101 MHz (≙5.89 Gbit/s)
MAC Address Translation CS	723/625 (≙2 %/≙2 %)	1 (≙1 %)	178.230 MHz (≙5.70 Gbit/s)

Table V
RESOURCES AND SPEED FOR ALL DESCRIBED CONTROL STAGES

configurator as well as the packet classification engine, the rule set engine and the described 12 control stages. Required resources and reached speed are shown in Table VI. Thereby, the speed is sufficient to process traffic in wire speed so that no packets have to be dropped. In order to avoid packet loss, the overall delay time must be less than or equal to a time needed for the internal processing of 248 bytes. Consequently, the following formula must be satisfied:

$$\frac{D_i}{T_i} \leq \frac{D_e}{T_e}, \tag{1}$$

where D_i is internal data volume, T_i is internal throughput, D_e is external data volume and T_e is external throughput. As the internal throughput is 4.8 Gbit/s and the external throughput is 2 Gbit/s, the formula is satisfied.

Subsystem	Slices Flip Flop pairs	BRAM	Speed
SecAN firewall	14.267/18.456 (≙ 31.9%/≙ 41.2%)	45 (≙ 30.4%)	150.3 MHz (≙4.8 Gbit/s)

Table VI
RESOURCES AND SPEED FOR THE SECAN FIREWALL SUBSYSTEM

Web Filtering: Web filters are a very sensitive issue and have been poorly discussed in the research community. Some countries such as China, the United States, and Great Britain [10] already use Web filtering. The British system "Cleanfeed" has a two stage structure [10]. In the first stage, the system filters IP addresses. If the IP address matches a request is sent to the external data base to verify the domain. The data base is managed by the Internet Watch Foundation (IWF), which collects reports about criminal online content. "Cleanfeed" grants an efficient domain filtering. However, it suffers from high latency due to its structure, which constrains the Web surfing experience of users. The US Web filtering system achieves better latency. However, overblocking was substantiated, i.e., the Web sites were blocked although they were not blacklisted [11]. The China Internet filtering system inspects Web traffic for specified keywords [12]. If the keyword is found the Web filter resets the connection by setting the TCP reset flag. The frame that contains the keyword is still forwarded to the recipient. If the endpoints ignore the reset flag the connection persists. Thus, the content can be transported to the requester.

The developed Web filter avoids the drawbacks of the mentioned Web filtering systems. The suggested solution solely utilizes local resources ensuring high processing speed. Moreover, it cannot produce false positives as each domain is exactly verified in the blacklist. Thereby, overblocking is avoided. The packet with malicious content is dropped and though the communication is interrupted.

SecAN Web filter module filters HTTP traffic. It inspects HTTP-GET requests for domain name of the Web server. Afterwards, the domain name is checked in the local blacklist. HTTP-based filtering as opposed to DNS filtering grants immediate effect, which is an essential issue in order to block a malicious Web content. Moreover, HTTP requests to proxies are checked as well. As the authors' goal is to monitor Web traffic, other protocols should not be blocked. By using HTTP monitoring, the Web filter cannot be simply bypassed by adding the IP address of the Web server into the local hosts file.

High throughput requires fast search algorithms. Therefore, the suggested Web filtering approach has two level search architecture. In the first level search, the domain name is hashed with CRC64 hash function. The calculated hash value is checked in the hash table, which is stored in a cache memory. The rree structure of the hash table grants logarithmic complexity. Moreover, cache access time ensures high search speed. As hashing produces false positives, a second level search is required. The full domain names is stored in the on-board DDR2 SDRAM. If the first level search was successful, start search address for the DDR2 SDRAM is provided. The blacklist has a bucket structure, i.e., all domain names, which generate the same hash value belong to one bucket. The domain names in one bucket are linearly searched. If the requested domain name matches in the blacklist, the HTTP-GET frame is dropped. Otherwise, it is passed through. The described structure is depicted in Figure 6.

In order to test the Web filter module, a data bank with real world domains provided from domain name registrar VeriSign is used [13]. 23 million domains were hashed with the CRC64 hash function and thereby 159 collisions were detected with a maximum of two domains per collision. The collision ratio is $6.9 \cdot 10^{-6}$. As a result, one bucket would normally have only one domain and thus only one DDR2 SDRAM access is required in the second level searching. According to that, the possibility to get a false positive in

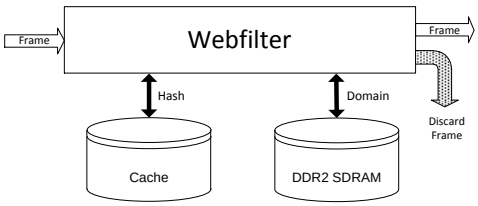


Figure 6. Webfilter structure

the first level searching is under 1‰.

Resource consumption of the Web filter module for the test platform (XILINX FX70T) is depicted in Table VII. The blacklist size was limited to 4096 entries. Growing blacklists result in increasing BRAM and DDR2 SDRAM consumption. However, slice consumption remains constantly low.

Modul	Flip Flop / LUT Slices	BRAM	SDRAM	Speed
Webfilter	897/2319 (≙2 %/≙5 %)	13 (≙9 %)	1.1 MB (≙0.2 %)	112 MHz (≙3.58 Gbit/s)

Table VII
WEB FILTER MODULE RESOURCE CONSUMPTION

The processing of the Web filter's blacklist induce indispensable computation and waiting cycles. In the worst case, 12 cycles cache computation and 54 waiting cycles for DDR2 SDRAM collision resolution are necessary. As the system processes 4 bytes per cycle, the overall delay time corresponds to a time needed for processing 264 bytes. Thereby, the shortest domain name (e.g., *"g.cn"*) can be found after 62 processed bytes. In order to avoid packet loss, the overall delay time must be less than or equal to a time needed for the internal processing of 326 bytes. Consequently, the Formula 1 must be satisfied. In the case of the Web filter, the internal throughput is 3.58 Gbit/s. The external throughput is 1 Gbit/s as stated above. Therefore, a frame has to be at least 92 bytes long to avoid packet loss. However, the length of the shortest possible HTTP-GET frame inclusive interframe gap is 89 bytes. As a 2 KB buffer is used to store incoming frames, packet loss could occur if 683 HTTP-GET requests with minimum length followed each other. This scenario is not realistic on a DSLAM because in practice, the average HTTP-GET frame length is about 400 bytes due to additional HTTP headers. Moreover, HTTP-GET requests represent a fractional part of the overall Internet traffic.

Signature Recognition: To detect malicious signatures at wire speed, Bloom filter-based deep packet inspection technologies are used. The signature detection starts after the header of the transport layer. As there is no clear definition, where to find the signature in the payload, compared with searching for specific header information, signature detection is a problem of massive parallel pattern matching. This problem is solved by concerning pattern matching at wire speed, using a Bloom filter cluster approach (see Figure 7). A Bloom filter is a space-efficient data structure for checking set affinity. The checked element is compressed using several hash functions and depicted to a bitmap-like structure. The hash values serve as bitmap addresses. For programming a Bloom filter, the bitmap is first initialized with zeros. Afterwards, each element of the set is hashed and the bitmap is set to one at the corresponding addresses. For checking set affinity, the alleged element is compressed and looked up in the bitmap. If each address points to a set bitmap element the element is an element of the set with a certain probability. Elements that are recognized as elements of the set but are not due to all their bitmap-elements set by other elements, are called false-positives. The rate of detecting false-positives at a single Bloom filter is called false-positive rate.

For the realization of a Bloom filter based pattern matcher, the Bloom filter set is the signature database whereas the single set element is a specific string. In the implementation, the incoming data stream passes an n byte long monitoring window where each signature length is analyzed by a separate Bloom filter. This is acceptable because every Bloom filter can only hold elements of the same length. Otherwise, the false-positive rate would increase dramatically and no conclusion of set affinity would be possible. The result of the Bloom filter-based analysis is coordinated by an arbiter due to possible simultaneous matches at the same time. One possible algorithm for the analyzer could be longest match first referring to the smaller false-positive-probability for larger signatures. Finally, the match analyzer eliminates all false-positives and generates alarm signals on malicious signature. In this constellation, the Bloom filter cluster plays the role of an optimal pre-filter for the match analyzer reducing the number of possible signature matches found by conventional signature detection algorithms.

As basis for our signatures, the database of the free intrusion detection system SNORT is used, which is parsed and prepared by some external tools for the use in the Secure Access Node. Furthermore, ISP administrator rules are supported, which can be defined in a SNORT-like syntax. To improve the quality of malicious signature detection, additional attack correlated information like protocol type, input port and output port were integrated in the Bloom filter-based pattern matcher to realize a lightweight, hardware-based intrusion detection system working at full wire speed.

The prototype achieves a signature filter-rate of 1.056 GBit/s with a real false-positive rate of less than 0.001 according to the current SNORT database. Building parallel clusters of Bloom filter-based signature matchers, even more throughput can be achieved due to the linear scaling of the filter-rate with the number of instances. The data stream itself is then analyzed by each instance with a particular offset. The resource consumption of the Bloom filter cluster

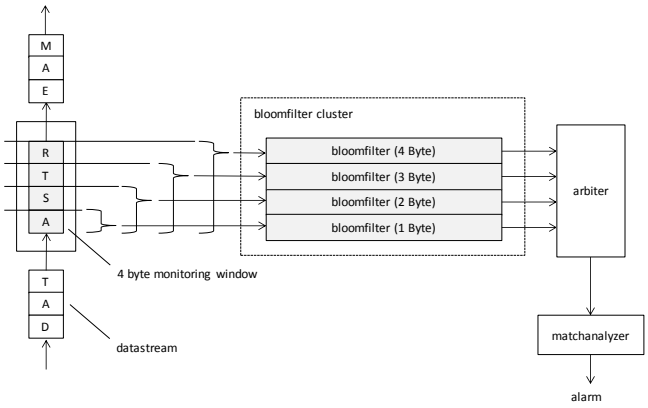


Figure 7. Bloomfilter-based IDS filter cluster for exemplary 4 byte monitoring window

matching signatures with a maximal length of 30 bytes is given in table VIII.

Modul	Flip Flop / LUT Slices	BRAM	Speed
Bloomfilter	16,234/17,150 (≙36 %/≙38 %)	134 (≙90 %)	33 MHz (≙1.06 Gbit/s)

Table VIII
BLOOMFILTER MODULE RESOURCE CONSUMPTION

IV. GENERATION AND DISTRIBUTION OF SECAN'S CLOCK SIGNALS

SecAN has an outer and at least one inner clock domains. The clocks for all clocking domains are generated by clocking moduls using a single digital clock manager (DCM) and global buffers.

In the outer clock domain, the ML507 evaluation board receives and sends data over SecAN's Ethernet interfaces. Both interfaces are able to process data at a speed of 1 Gbit/s. That means, each interface has to process data with 125 MHz and 1 byte per clock cycle. Following the Ethernet interfaces, FIFOs with separated read and write clocks are used to synchronize the outer and the inner clocking domain.

Furthermore, the inner clocking domain depends on SecAN's subsystem: the deep packet inspection, Web filter, and packet filter module. Although the functionality of the DPI module is very complex, the generation of the clock is very easy. Because the DPI subsystem solely uses FPGA internal resources, only one inner clock domain is existent. Hence, the clock for the DPI module as well as the clock for the outer clock domain is generated by the same DCM module.

The generation of the clock for the Web filter subsystem is slightly more complex. Because this control stage uses the DDR2 memory to verify domain matches, there are two inner clock domains - one for the DDR2 memory and one for Web filter's internal logic. The used DDR controller is based on Micron sources. A system internal DCM module uses the 100 MHz board clock as input clock and generates a 200 MHz clock for DDR2 internal processes as well as a 125 MHz clock for all other Web filter modules.

The last of the three subsystems is SecAN's packet filter firewall. This system has the most complex clocking scheme because it uses SRAM as well as DDR2 memory. Both memories have different clocks. Hence, the challenge is to synchronize both memory and the packet filter modules optimally. Figure 8 shows the generation as well as the clock arrangement of the SecAN's firewall subsystem.

Because the packet filter firewall and the Web filter module uses the same DDR controller, the clock generation is exactly identical. DMC 1 generates the 200 MHz and the 125 MHz input clocks of the DDR controller. The controller itself has an internal DCM and generates a new 125 MHz clock. This clock is phase shifted relative to the 125 MHz input clock and should be used by the connected hardware. In case of the packet filter firewall, the SRAM and all firewall internal modules achieves a higher speed of 150.3 MHz. For that reason, the DCM 2 generates from the 125 MHz DDR controller clock three new clock signals. The 125 MHz clock is used for receiving and sending Ethernet data and both 150 MHz clock are used for the internal firewall hardware modules.

Although the SRAM and all firewall internal modules use the same frequency, there is a phase shift between both clocks. The phase shift depends on the distance between FPGA and SRAM memory on the evaluation board. This delay is compensated by a SRAM feedback signal (shown in Figure 8). The phase shift depends on the achievable speed and is 180 °. For example, if the speed is reduced to 80 MHz the phase shift between both clocks is compensated.

Through the use of multiple DCMs, different clock domains inside the hardware firewall can be generated. The data delivery between these clock domains is realized with clock domain crossings. That means, synchronization FIFOs, which have separate read and write clock input ports, are used. If, e.g., request data is sent to the DDR controller, this data is written with a speed of 150 MHz in the DDR controller request synchronization FIFO. After that, the 200 MHz clock reads the request data from the same FIFO. The answer from the DDR controller is written with 200 MHz into the answer synchronization FIFO and read with 150 MHz. Multiple use of DCMs together with clock domain crossings allow to achieve the maximum speed for each subsystem.

V. CONFIGURATION SOFTWARE

Via a web interface, customers can set their own filtering rules. Before these rules are applied, they are verified by the ISP. The configuration of the hardware is done by platform

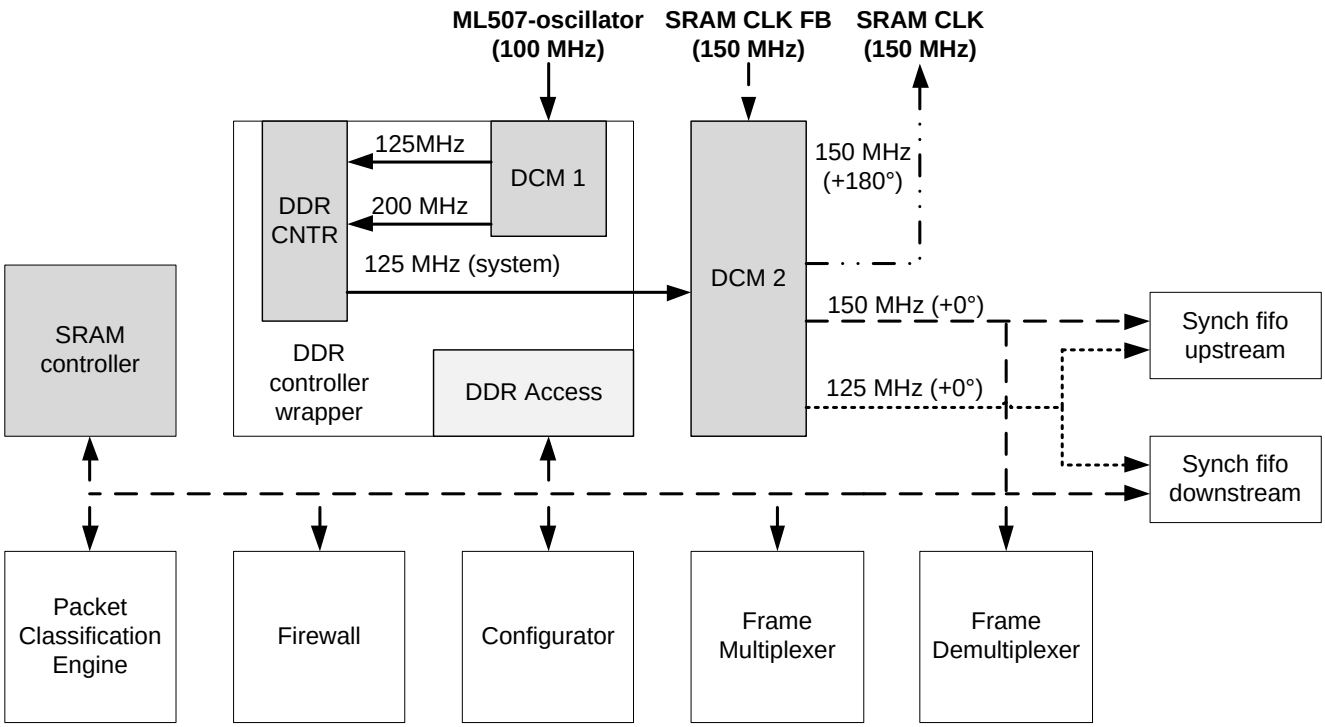


Figure 8. Clocking scheme of the Secure Access Node packet filter

independent software developed with QT. The graphical user interface (GUI) consists of a framework, which is able to include so called plugins. Each plugin offers a GUI to configure a separate hardware component of the Secure Access Node. When starting the GUI, the software searches in a special directory for available plugins. All plugins are loaded and appear in the software as a tab. By means of the plugins, ISP provided rule can be generated and customer rules are applied. Furthermore, the configuration software is able to interrupt the hardware processing flow for updating the hardware configuration.

VI. Conclusion

Because many subscribers do not have the necessary knowledge to maintain their own security measures, it is important to include security features at the ingress of the network. Therefore, we have designed a software/hardware co-design consisting of a packet filter firewall, a signature detection, and a Web filter module. The implementation results show a reachable speed of 150.3 MHz corresponding to 4.81 Gbit/s. Furthermore, subscribers are protected by the Secure Access Node and do not need to care about their own security. Especially for the large number of customers with minor technical knowledge, this is an important feature. Because of the applied methods, the bandwidth of customers is not influenced. Furthermore, no attacker has access to the hardware. Only an ISP administrator is able to update the security mechanism. Moreover, it is possible to update the system during operation. Prospectively, a functional test with real traffic data is intended.

As many subscribers do not have the necessary knowledge to maintain their own security measures, it is important to include security features at the ingress of the network. Therefore, a hardware-based approach consisting of a packet filter firewall, a Web filter module, and a signature detection engine is presented. As a hardware solution, it offers more advantages in terms of security and robustness. The implementation results show a reachable throughput of 4.81 Gbit/s for the packet filter firewall, 3.58 Gbit/s for the Web filter module as well as 1 Gbit/s for the intrusion detection engine. The throughput is only limited by the FPGA type and can be even multiplied by using application-specific integrated circuits.

Furthermore, subscribers are protected by the Secure Access Node and do not need to care about their own security. Especially for the large number of customers with minor technical knowledge, this is an important feature. Because of the applied methods, the bandwidth of customers is not influenced. The configuration of the suggested security system can be done only by the network administrator. Since the Secure Access Node is fully transparent for all network participants, it is safe from attacks. Moreover, it is possible to update the system during operation.

Prospectively, a functional test with real traffic data is intended.

ACKNOWLEDGMENT

We would like to thank the Broadband Access Division of Nokia Siemens Networks in Greifswald, Germany for their inspiration and continued support in this project. This work is partly granted by Nokia Siemens Networks.

REFERENCES

[1] J. Rohrbeck, V. Altmann, S. Pfeiffer, D. Timmermann, M. Ninnemann, and M. Roennau, "Secure Access Node: an FPGA-based Security Architecture for Access Networks," *The Sixth International Conference on Internet Monitoring and Protection (ICIMP 2011)*, pp. 54–57, 2011.

[2] D. Taylor and J. Turner, "Scalable packet classification using distributed crossproducting of field labels," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pp. 269–280, 2004.

[3] A. Guruprasad, P. Pandey, and B. Prashant, "Security features in ethernet switches for access networks, TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region ," pp. 1211–1214, 2003.

[4] Nokia Siemens Networks GmbH & Co KG, "Multi-Service IP-DSLAM SURPASS hiX 5622/5625/5630/5635 Release 3.8M," January 2012. [Online]. Available: http://www.itm-group.com/web/fileadmin/itm/datenblaetter/NSN/hiX-56xx.pdf

[5] Xilinx, "Platform User Guide rev3.1.2," January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf

[6] G. S. Jedhe, A. Ramamoorthy, and K. Varghese, "A Scalable High Throughput Firewall in FPGA," *16th International Symposium on Field-Programmable Custom Computing Machines*, pp. 43–52, Apr. 2008.

[7] W. Jiang and V. K. Prasanna, "A FPGA-based Parallel Architecture for Scalable High-Speed Packet Classification," *20th IEEE International Conference on Application-specific Systems, Architectures and Processors*, pp. 24–31, Jul. 2009.

[8] M. Dixit, B. V. Barbadekar, and A. B. Barbadekar, "Packet classification algorithms," *IEEE International Symposium on Industrial Electronics*, no. ISlE, pp. 1407–1412, Jul. 2009.

[9] K. Palanisamy and R. Chiu, "High-Performance DDR2 SDRAM Interface in Virtex-5 Devices, rev. 2.2," January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp858.pdf

[10] R. Clayton, "Anonymity and traceability in cyberspace," *ACM SIGACT News*, vol. 36, no. 653, pp. 115–148, Nov. 2005.

[11] US District Court for the Eastern District of Pennsylvania, "CDT, ACLU, Plantagenet Inc. v Pappert," *337 F.Supp. 2d 606*, September 2004.

[12] R. Clayton, S. J. Murdoch, and R. N. M. Watson, "Ignoring the great firewall of china," *6th Workshop on Privacy Enhancing Technologies*, no. 16, June 2006.

[13] Verisign, Inc., January 2012. [Online]. Available: www.verisign.com

CincoSecurity:
Automating the Security of Java EE Applications with Fine-Grained Roles and Security Profiles

María Consuelo Franky
Department of Systems Engineering
Pontificia Universidad Javeriana
Bogotá, Colombia
lfranky@javeriana.edu.co

Victor Manuel Toro C.
Department of Systems and Computing Engineering
Universidad de los Andes
Bogotá, Colombia
vm.toro815@uniandes.edu.co

***Abstract*— Almost every software system must include a security module to authenticate users and to authorize what elements of the system can be accessed by each user. This paper describes a security model called "CincoSecurity" that follows the Role Based Access Control model (RBAC), but implementing fine-grained roles that can be grouped into "security profiles". This leads to a great flexibility to configure the security of an application by selecting the operations allowed to each security profile, and later, by registering the users in one or several of these profiles. We describe also a security software module (that implements the CincoSecurity model) that we propose to be the initial code baseline for the development of any Use Cases oriented Java EE system, offering from the beginning a flexible, extensible and administrable access control to the elements of the application that is to be developed. Moreover, CincoSecurity allows automating the generation of the additional code required to protect the use cases and its elements of the Java EE application being developed, with tools that add the required security restriction code accordingly with the proposed security model.**

Keywords- Security; Access control; RBAC; Framework; Java EE; Seam; Security automation.

I. INTRODUCTION

This paper summarizes the experience of the authors designing and developing a reusable security module, called CincoSecurity, that has been used for several years to control access to the elements of web applications written in Java Enterprise Edition (J2EE initially [9] and later Java EE 5 [10]). Currently, the module CincoSecurity is available [18] under the GPL license, and is used by some important software houses in Colombia.

The security model underlying CincoSecurity implements a RBAC (Role-Based Access Control) [7], providing high flexibility to control access to the various elements of a Web application, such as the invocation of an operation of a business component, the access to a web page, or the access to elements within that page. The innovation of CincoSecurity is the use of very fine-grained roles, each role having a single permission associated with the invocation of an operation (method) of a business component. From these fine roles —whose fulfillment the Application Server can directly control at run-time— CincoSecurity allows to define "security profiles" as sets of fine-grained roles. This facility of security profiles gives a great flexibility for configuring the security of an application by selecting the operations allowed to each profile (i.e., selecting a set of fine-grained roles for each profile), and later, by registering the users in one or several of these profiles.

A Java EE web application that is to be constructed with the Seam framework [12] gets several benefits by integrating the CincoSecurity module. When a user authentication is performed, the Application Server is informed about the fine roles derived from the security profiles the user belongs to, and a personalized menu is dynamically built containing only the entries leading to the use cases allowed for the user. Additionally, CincoSecurity contributes to the application being constructed with several use cases to administer the security profiles, to manage user registration in these security profiles and to administer passwords. Additionally, CincoSecurity comes with use cases to register new modules, new use cases and new services, as they become available during the development project, for their security to be administrable.

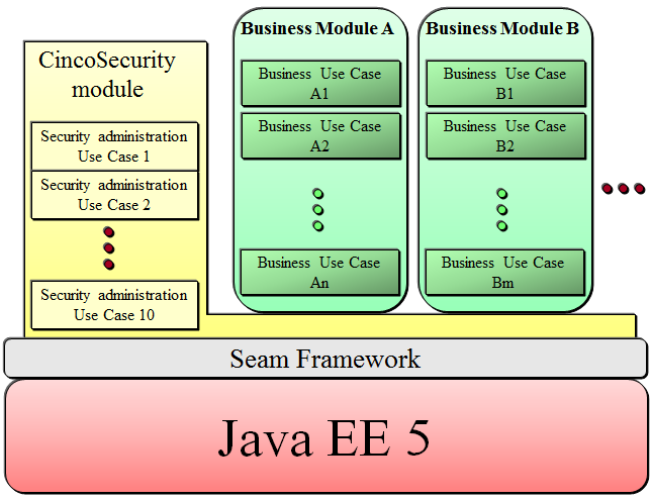


Figure 1: What is the CincoSecurity module?

Figure 1 illustrates the CincoSecurity module as a platform to embed and support security (and its administration) into a new Java EE application. This means that, after generating the very initial codebase of the new application with the Seam framework [12], CincoSecurity shall be the first module to be coupled into the new application, in order to be able to include and administer security of the forthcoming business modules.

Once embedded into the new application that begins to be developed, the CincoSecurity module facilitates to automate the incorporation of security into the new modules as they are built, by the means of tools that add security restrictions to each new use case and its elements, and with administrative use cases (coming with CincoSecurity) that configure the security of the whole application. With respect to previous work of the authors [1], this paper is an extension that explains in detail how to automate the incorporation of security into a new Java EE application by applying the CincoSecurity model.

In the following section, this paper presents the RBAC security model on which CincoSecurity is based, and more specifically, the RBAC model applied in the context of Java Application Servers. Then, additional concepts provided by the CincoSecurity module are introduced, as well as its entities model. Later, there is a description of the use cases coming with the CincoSecurity module (e.g., create a new user, create/edit a security profile, add/delete users from a security profile, etc.). Then, the paper provides a short summary and references to the detailed guidelines [19] for integrating the CincoSecurity module to a Java EE application built with the Seam framework [12]. At the end, the security automation of an application that uses CincoSecurity is explained. Finally, there is a comparison with other works, followed by the conclusion and a short description of our future work.

II. EVOLUTION OF THE RBAC SECURITY MODEL

The RBAC model introduced the concept of "role" to control the access to computing resources. The RBAC term was first proposed by Ferraiolo and Kuhn [3], based on previous works of Baldwin [2]. The initial proposal of this model creates a role for each type of job within an organization (cashier, customer service person, office director, ...). Then, each role is assigned with the set of access permissions that are required for this type of job. Finally, each user is enrolled into one or more roles (rather than to specific permissions). This model simplifies security management because the roles (with their associated permissions) tend to be stable, and users can be added or retired easily from roles. The RBAC model allows reinforcing the "least privilege" principle by giving each user the minimum set of permissions required to perform his work, by enrolling him only in the appropriate roles [7].

From the initial RBAC model (called Core RBAC) the work of Sandhu and colleagues [4] defined extended models, such as the hierarchical RBAC (to include role hierarchy with inheritance of permissions), and constrained RBAC (to prevent, for example, to assign a user to two conflicting roles, or to restrict the time interval in which a user can use the permissions of one of its roles).

The main applications of the RBAC model have been in Data Base management Systems, Enterprise Security Management Systems, and Web applications that run on Application Server s [6] [7] [8].

The wide spread of RBAC models, implemented in numerous products from many providers, led to define an ANSI standard [5] in 2004, aiming to standardize terminology, promote its adoption and improve productivity. However, the current RBAC ANSI standard (consisting of a reference model and a functional specification) has some limitations and gaps as indicated in the work of Bertino and colleagues [8].

III. THE RBAC MODEL APPLIED TO JAVA EE APPLICATION SERVERS

Since the late 90's, the emergence of Application Servers brought a new way to build web applications (both in the enterprise Java platform and in Microsoft .NET), with business components managed by containers that provide added services for security, transaction management, parallelism, pool of connections, logging, etc. [9].

Regarding security, Java EE Application Servers [10] implement the Core RBAC model [7] to control the access to resources based on the roles the user belongs to. In order to take advantage of these security services (and not to write additional code in the application to internally control the access to resources), it is necessary to specify the roles of the application, the association of resources to roles, and the association of users to roles.

A. Enrolling users in roles

In a Java EE application that uses a database to store the authentication and authorization information, the following entities EJB3 (Enterprise Java Beans - version 3) are required [11]:

- An entity "User" shall be implemented (with its corresponding support table in the database), to store users and passwords.
- Entities shall be implemented (with its support tables in the database) to specify the association of each user with one or more roles.
- A "User management" use case shall be implemented to enroll a user in one or more roles.

These facilities are included in CincoSecurity. Similarly, it is also possible to store users, passwords and roles in a LDAP (Lightweight Directory Access Protocol) server.

B. Controlling access to resources

Seam is a framework to develop Java EE applications, that is being developed by JBoss since 2005, whose principal author is Gavin King [12] [14]. Seam allows to directly expose and use in the Web layer the entities and business components of the application. This simplifies enormously the development by eliminating the intermediaries and conversions between the layers of the application. Seam has been widely accepted and has been incorporated in the recent

Java EE 6 standard, under the name of "CDI" (Contexts Dependency Injection).

To control access to resources in a Java EE application that uses the Seam framework, the following strategies are required [13]:

- An annotation is used to protect each method of the session EJB3. This annotation indicates what roles are authorized to invoke the method.
- The url of each JSF (JavaServer Faces) web page [10] can be protected in the navigation flow descriptor (pages.xml) so that it can be accessed only by users belonging to one of the specified roles.
- Each button or element of a JSF web page can be protected so that it is rendered only to users belonging to one of the specified roles.

Notice that annotations must be scattered along the code —in the declaration of methods, in the section of a page in the navigation flow descriptor, in the buttons tags and elements of JSF pages— to indicate what roles can access these elements.

C. User authentication

In a Java EE application that uses Seam, the authentication service must be specified in the descriptor components.xml. This service shall be a method of a class of the application, and must implement a query in JPQL (Java Persistence Query Language) [11] to verify the user's password (alternatively this process can also be performed with a LDAP server).

Additionally, the authentication service must also obtain the roles of the authenticated user. With the Seam component called "Identity" these roles can be added to the session and informed to the Application Server.

D. Controlling access to a JSF page

When an http access request is received, the Application Server verifies if the user belongs to a role allowed to access the requested JSF page, and if so, the requested page is displayed.

For example, in the following piece of the navigation flow descriptor it is specified that the access to myPage.xhtml is granted only to users having the 'tourist' role:

```
<page view-id="/myPage.xhtml" login-required="true">
   <restrict> #{s:hasRole('tourist')} </restrict>
</page>
```

Similarly, inside the page only the elements that the user is authorized to see are shown (elements such as buttons and text boxes can specify, with the attribute "rendered", what roles can see them). For example, in the following piece of page it is specified that the button "View hotel" is visible only to users with the 'tourist' role:

```
<s:button id="viewHotel" value="View hotel"
   action="#{hotelBooking.viewHotel(hot)}"
   styleClass="buttonSmall"
   rendered="#{s:hasRole('tourist')}"
/>
```

E. Authorizing an action from a JSF page

In a JSF page a button's action is typically associated with the invocation of a method of a session EJB3. The server verifies that the user roles allow him to invoke the associated method, assuming that the method is protected by an annotation indicating the roles that can invoke it.

For example, in the following piece of a session EJB, the method viewHotel is allowed only to users with the 'tourist' role:

```
@Restrict ("#{s:hasRole('tourist')}")
public void viewHotel(Hotel hot) {...}
```

IV. ADDITIONAL CONCEPTS IMPLEMENTED BY THE CINCOSECURITY MODULE

In addition to the security concepts for a Java EE application that uses the Seam framework [12] [13] explained above, the CincoSecurity module implements additional concepts to provide greater flexibility to define the permissions for users.

A. Use case and services

Definition: A use case is a system's capacity to deliver a useful and indivisible functionality to the user.

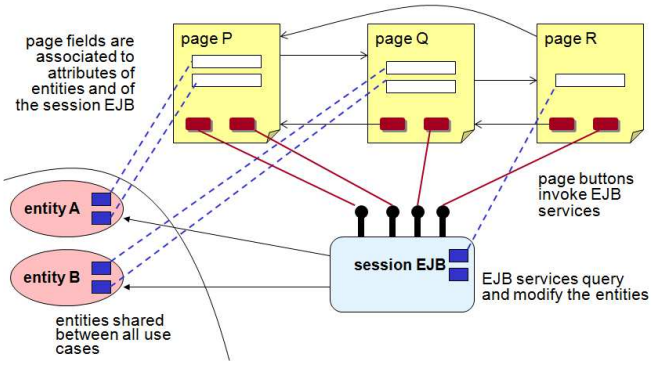


Figure 2: Elements of a Use Case implemented in Java EE with Seam

Definition of a use case in terms of its implementation in Java EE with Seam (see Figure 2): a use case consists of one (or more) business entities and a group of services that act upon them. These services are implemented as methods of a session EJB3 (see "use case controller" pattern). One or more JSF pages display attributes of the entities involved in the use case, and attributes of the session EJB3 controlling it. In those JSF pages there are actions that invoke the services of the session EJB3. These EJB3 services (methods) are programmed in terms of queries and modifications to the persistent business entities. The navigation flow descriptor contains rules to decide the next page to display.

B. Module

Definition: A module is a set of related use cases. The CincoSecurity module comes with the following use cases, that will be explained below: security profiles management, users management, change of password, basic security reports, registration of menu entries, and registration of modules, use cases and services.

C. Fine grained roles

The CincoSecurity module works with fine-grained security roles:

- A role for entering to each use case. The name assigned to this role is the same name of the use case (which is also the Seam name of the session EJB3 that supports the use case).
- A role to invoke each service within a use case (i.e., each of the methods of the session EJB3 that supports the use case). The name assigned to this role is "use case name"_ "method name".

D. Protection of resources

- The session EJB3 that supports a use case is protected with an annotation indicating the role for entering to the use case. For example, the profileGestion use case is supported by the session EJB3 ProfileGestionAction.java (which implements the interface ProfileGestion.java); this EJB3 has the Seam name "profileGestion". Consequently, the role for entering to this use case is called "**profileGestion**" and the EJB3 class will have the following annotation:

```
@Restrict("#{s:hasRole('profileGestion')}")
```

- Each service (method) of the session EJB3 is protected by an annotation indicating the role associated with the service. The name of this role is the concatenation of the use case name with the name of the service (with "_" between). For example, the **update** method of the session EJB3 that supports the use case **profileGestion** will have the following annotation:

```
@Restrict("#{s:hasRole('profileGestion_update')}")
```

- Methods get and set do not require any annotation: they are protected with the role of entering to the use case.
- Access to each JSF page of a use case is protected in the navigation flow descriptor by the role for entering to the use case. For example, the page **profiles.xhtml** of the use case **profileGestion** has a navigation flow descriptor called **profiles.page.xml** that contains the following restriction:

```
<page view-id="/profileManagementInit.xhtml"
        login-required="true">
   <restrict>
      #{s:hasRole('profileManagement')}
   </restrict>
</page>
```

- Each button in a JSF page should be displayed only to users having the role associated to invoke the action of the button, which corresponds to an EJB3 service (method). For example, the **profiles.xhtml** page of **profileGestion** use case contains a button whose associated action is to invoke the **update** method of the session EJB3 that supports the use case. Consequently the button tag indicates that it is showed only to the role **profileGestion_update**:

```
<h:commandButton id="update"
   value="Update"  styleClass="button"
   action="#{profileGestion.update}"
   rendered="#{s:hasRole('profileGestion_update')}"
/>
```

E. Security profile

A security profile is a set of fine roles, each fine role expressing the right to invoke a service belonging to a use case. Unlike the role, the concept of security profile is not supported directly by Application Servers and must be implemented with additional entities.

The use cases of the CincoSecurity module allow the association of users to roles via security profiles:

- A user can be enrolled in one or more security profiles, so he/she will have the set of fine roles allowed by the union of these profiles.
- There is a *many-to-many* relationship between users and security profiles.
- There is a *many-to-many* relationship between security profiles and fine roles.

F. Actions after a user authentication

After a user is authenticated, CincoSecurity calculates all the fine grained roles from the security profiles the user belongs to, and informs them to the Application Server (by assigning these roles to a Seam component called "Identity"). Additionally, the EJB3 Login performs the following actions:

- The session timeout is set, according to the parameters stored in the database.
- The user's menu is built, containing only the entries leading to use cases allowed to the user.
- The security information of the user is added to the session context, should the application logic needs it.

It is important to remark that the access to use cases not authorized to a user by any profile is prevented in two ways. From one side, not authorized use cases do not appear in the user's menu. From the other side –even if the user types in the url of a not authorized use case– the Application Server throws a security exception. This happens because the fine role for entering to this use case was not included in the list of fine roles that was informed to the Application Server.

Figure 3: User menu allowing access to all use cases of CincoSecurity

The screen snapshot of Figure 3 shows the menu of a user that is enrolled in security profiles allowing access to all the use cases of the CincoSecurity module.

Figure 4: User menu allowing access to fewer use cases of CincoSecurity

The screen snapshot of Figure 4 shows the menu of another user that is enrolled in security profiles allowing access to just a few use cases of the CincoSecurity module.

V. ENTITIES MODEL OF THE CINCOSECURITY MODULE

The entities model shown in Figure 5 illustrates the relationship one-to-many from Module to Usecase, and from Usecase to Service.

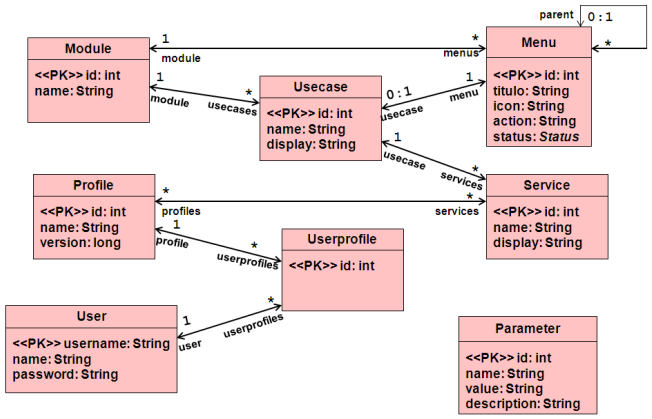


Figure 5: Model of entities of the CincoSecurity module

Figure 5 also illustrates the relationship many-to-many between Profile (security profile) and Service, as well as between Profile and User (via the intermediate entity Userprofile). Each menu entry may have submenus (only terminal menu entries have an action for going to the entry page of a use case).

The system parameters are arbitrary. They can be used, for example, to record the session timeout, the path of the directory to store reports, the address of the printer, etc.

In addition to this entity model, the CincoSecurity module also contains a view that directly associates a user with fine roles. The fine roles of a user are the union of the roles associated with the profiles the user belongs to.

VI. USE CASES OFFERED BY THE CINCOSECURITY MODULE

The following are the use cases offered by the CincoSecurity module:

A. *CRUD Use cases*

The CincoSecurity module offers:

- A use case to list/add/edit and remove **parameters** of the application.
- A use case to list/add/edit and remove **modules** of the application:

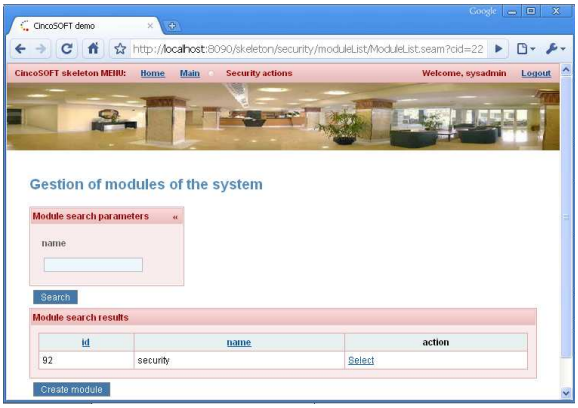


Figure 6: Use case to list/add/edit or remove modules.

- A use case to list/add/edit and remove the **use cases** of a module:

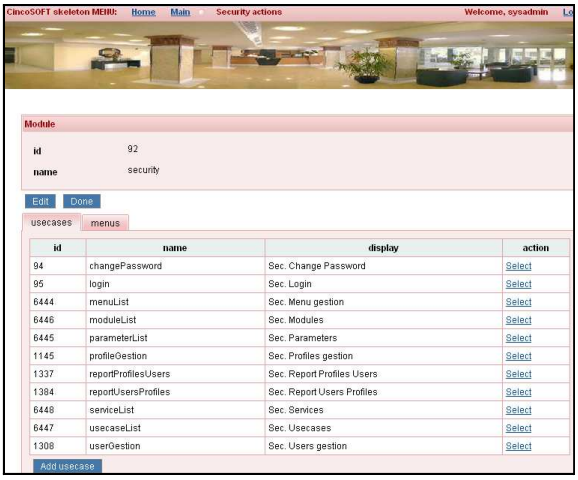


Figure 7: Use case to list/add/edit or remove use case.

- A use case to list/add/edit and remove the **services** of an application's use case.

- A use case to list/add/edit and remove **menu entries**:

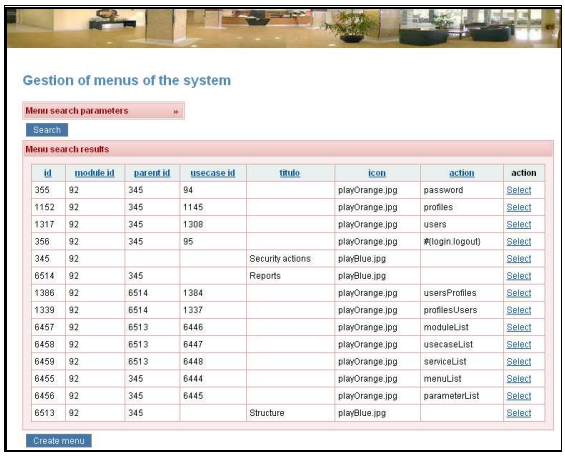


Figure 8: Use case to list/add/edit or remove menu entries.

It can be easily specified what menu entries have a submenu, as well as the use case associated with a terminal menu entry (see Figure 8).

B. *Management of security profiles*

This use case allows to add/edit/remove security profiles. Initially, the existing security profiles are listed. When a security profile is selected, the modules, use cases and allowed services are shown, so that the user can check or uncheck services (see Figure 9).

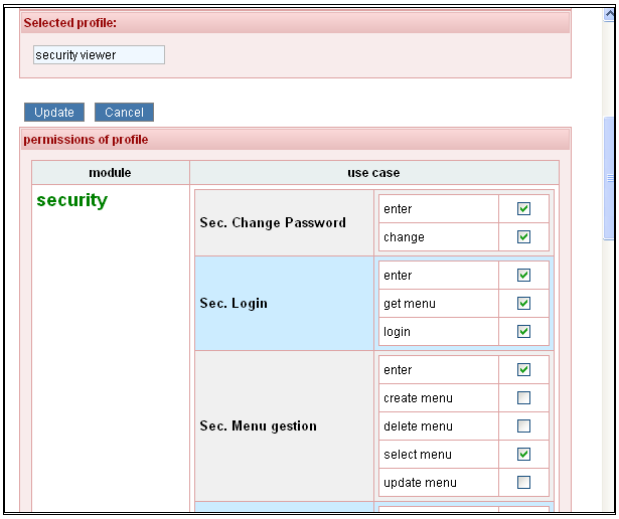


Figure 9: Use case to manage security profiles.

Similarly, the user can create a new security profile. In this case, the system displays all modules, and within it, the use cases and services, for the user to select those allowed by the new profile.

C. *Management of users*

This use case allows adding users of the application, indicating its name, login and password. It also allows enrolling the new user in one or more security profiles.

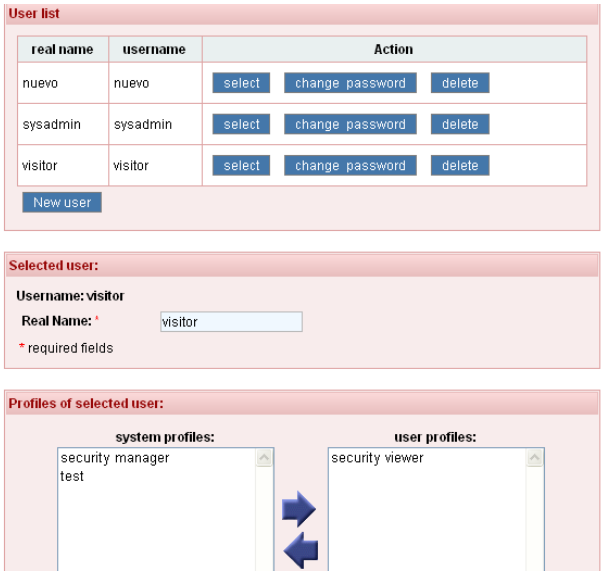


Figure 10. Use case to manage users.

D. *Password change*

This use case allows a user to change his password. Passwords are stored encrypted. See figure 11:

Figure 11. Use case to change password.

E. *Report of security profiles vs users*

This use case shows, for each security profile, what users are enrolled.

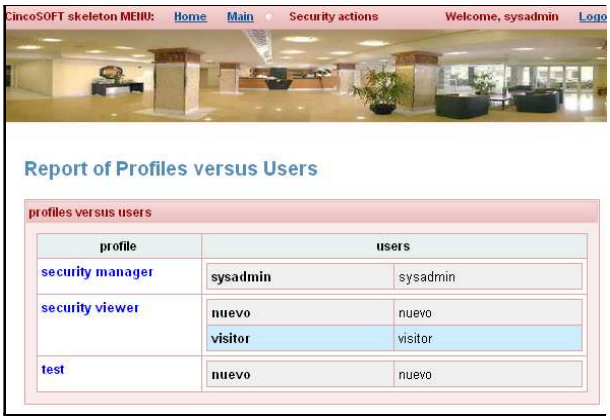


Figure 12. Use case to show a report of profiles vs. users.

F. *Report of users vs security profiles*

This use case reports, for each user, in what security profiles he/she is enrolled.

VII. How to integrate the CincoSecurity module to a Java EE Seam application

The CincoSecurity module is open source with GPL License [17]. It can be downloaded from SourceForge [18] in the form of an Eclipse project [16]. It comes ready to be deployed on the Application Server JBoss [15], but can be installed in any other Java EE Application Server by following the guidelines provided in the Seam manual [14].

The documentation accompanying the CincoSecurity module explains in detail how to deploy and execute the module, and how to integrate it with a Java EE application built with Seam. In particular, detailed explanations are included for registering application's use cases and services in the security module. An earlier publication about the CincoSecurity module, oriented to programmers, focused on these technical details [19].

It is important to emphasize that to incorporate and manage the security of an application, the modules of the application, the use cases contained in such modules, and the services offered by these use cases must be registered into the CincoSecurity module (by using CincoSecurity's use cases provided for this). This way, the fine roles associated with these services can be included in the security profiles, and the access to these use cases will appears in the menu of authorized users.

VIII. How to automate the protection of an application that uses the CincoSecurity module

The fine roles and the naming discipline proposed above to protect the services of the session EJBs and the web elements that invoke these services, allow to think in automating the protection of resources of an application that uses the CincoSecurity module. Indeed, from the names of the services to be protected, a tool could insert in the code the annotations (in the java sources) and the tag elements (in the JSF pages) required to achieve such protection.

With traditional coarse roles it is not possible to make such automation of the protection, because each time a role needs to be added or changed, it is necessary to review the whole code to decide what EJB services and web elements must be protected with the new role, thus requiring to manually write the appropriate annotations and tag elements.

The following section describes the main ideas of a generation framework based on Regular Expressions techniques [24], currently being developed by the authors, that automates the protection of a Java EE application that uses the CincoSecurity module.

A. *Techniques of Regular Expressions to build code generators*

The tool required to process Regular Expressions must be capable of detecting in a text file the strings –either in a single line or multi-line– that match a given tagged regular expression, and then, transforming the detected strings using the tags. These features are the basis for building a framework for code generation.

Example of tools that process tagged regular expressions are the java.util.regex library [24] [25] for Java programs and the replaceregexp command provided by the tool **ant** [26]. Below is an example of using this command to add the prefix **new_** to the name of each property in a set of files that contain properties (for example, a line with **aa = some string** becomes **new_aa = some string**):

```
<replaceregexp
  match="([^\s]+)=([^\s]+)"
  replace="new_\1=\2"
  byline="true"
>
    <fileset dir=".">
        <include name="*.properties"/>
    </fileset>
</replaceregexp>
```

In the previous example the regular expressions **[^\s]+** represents a word (*one or more non-space character*). The **match** attribute contains a regular expression that describes a property, enclosing in a first parenthesis the property name (tag \1) and enclosing in a second parenthesis the value of the property (tag \2). These tags are used in the **replace** attribute, that indicates that the **new_** string must be added before the name of the property. The command also indicates that the transformation must be done for each file of the form *.properties in the current directory, by analyzing each line separately.

The following example adds an annotation before the declaration of each Java class, which specifies the restriction that the user must be authenticated:

```
<replaceregexp
  match="(public class)"
  replace="@Restrict("#{identity.loggedIn}")  \1"
  byline="true"
>
    <fileset dir=".">
        <include name="*.java"/>
    </fileset>
</replaceregexp>
```

With techniques of Regular Expressions is also possible to extend source files. For example, the commands **replaceregex** and **loadfile,** provided by the tool **ant**, allow to assign to a property the text that matches a Regular Expression in a first source file, and then, inserting such text into a second source file, in the place matching a second Expression Regular.

B. *Framework of code generation based on on techniques of Regular Expressions*

The JBoss Seam generator [13] generates the initial skeleton of a Java EE 5 application with the following elements:

- Several control elements, based on the infrastructure frameworks JSF [10], Seam [13] and EJB3 [11].
- Several descriptor files correctly configured.
- An **ant** script [26] with tasks for compiling, packaging and deployment.
- Inclusion of many libraries providing infrastructure frameworks.

However, the code generated by Seam is not enough to be the basis of a serious business application. It does not have important features, such as: the organization of the source code into separated modules (each one with a set of related use cases) to facilitate maintenance; the inclusion of a module with use cases for managing the security; the inclusion of security constraints by fine roles in order to protect both the web pages and the business components.

Consequently, once the skeleton of a new application is generated with Seam, it is necessary to reorganize this initial skeleton and to couple it with the security module. An appropriate tool could incorporate the source code of the CincoSecurity module into the application. We have developed such tool based on techniques of Regular Expressions. The tool takes a copy of code pieces of CincoSecurity module, and then it adds or transforms the text, and incorporates the result into the software project under construction.

Other tools, also based on techniques of Regular Expressions, can add to the project use cases skeletons that facilitate to use JMS message queues [27], or to use report generator capabilities, or to produce pdf files, etc..

The set of all these tools is what we call a Code Generation Framework based on techniques of Regular Expressions. Each tool works from proven source code, which is copied, transformed and incorporated into the project that is being constructed.

C. *Generator to couple the CincoSecurity module*

The current version of the CincoSecurity module can be incorporated only to a Java EE 5 application that was initially created with JBoss Seam generator [13].

The CincoSecurity module can be directly taken as the generation model for the new application. The generator tool will look for certain strings in the source files of the module and will replace these strings with the appropriate strings for the new application. Examples of the strings that must be replaced are: the name of the project; the package name of the application; the name of the database, and so on. The descriptors of the new application must also be extended to incorporate the CincoSecurity module.

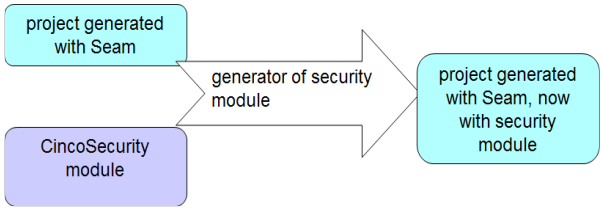


Figure 13. Generator of the security module, based on techniques of Regular expressions.

Figure 13 shows the process followed by the generator of the security module for an application previously generated with Seam.

Summarizing, the security module generator is composed by the tasks of text replacement, the tasks of copying the resulting files into the new application, and the tasks of extending the descriptors. These tasks must be expressed following the conventions of the tool used to process regular expressions.

D. *Automatic protection of new use cases by using the elements of CincoSecurity*

After incorporating the CincoSecurity module to an application, the Seam generator of CRUDs [14] can be used to generate the skeleton of new use cases. To facilitate the maintenance of the resulting application, this generation must be complement with a refactoring tool that reorganizes the generated code into subdirectories for modules, containing subdirectories for its use cases. This refactoring can be done with (Regular Expression) tasks that move the files to the appropriate directory, and (Regular Expression) tasks that fix the references in the web pages.

To take advantage of the security facilities provided by the CincoSecurity module (i.e., fine-grained roles and security profiles), the new use cases must be registered by the means of the use cases provided by CincoSecurity. After that, both the services of business components (EJBs) as the pages elements can be easily protected. A task, also based on Regular Expressions, provides:

- Registration of the new use case and each one of its services, associating each one with a fine-grained role.
- Registration of the menu item associated with the new use case.
- For each method of the EJB business component that controls the use case, an annotation with a security constraint is added, associating it with a fine-grained role.
- In the descriptors of the application, a security restriction for accessing the corresponding web pages is added, with the role of entering into the use case.

- A condition is added to each button on each web page of the use case, to make them visible only for users with the fine-grained role associated with the invoked service.

Thanks to the concept of fine-grained role and to the discipline of names proposed by CincoSecurity, the protection of the resources of the new application (use cases, services and page elements) can be added automatically. Subsequently, the administrator user can update the security profiles by selecting the services of old and new use cases (by the means of the use case of Management of security profiles, provided by CincoSecurity).

IX. COMPARISON WITH OTHER WORKS

There are other security modules proposed for the Java EE technology. Currently, in the SourceForge portal there is a dozen of software projects related with security for Java EE 5 applications, but most of them are proposals without implementation (i.e., they are in planning status). Two relevant projects with implementation and good acceptance from users are the following:

- *JPA Security* [20] is an Access Control Solution for the Java Persistence API (JPA) [11] with support for role-based access control, access control lists (ACLs) and domain-driven access control. Compared with CincoSecurity, this project does not offer access control to web pages as CincoSecurity does. JPA Security uses access rules in terms of database operations, which provides less flexibility than CincoSecurity, where security profiles are defined in terms of the services offered by the use cases of the application.
- *[fleXive]* [21] is a Java EE 5 open-source framework for the development of complex and evolving web applications. It offers an administration module that manages users and security. Some fleXive characteristics are a reliable data store backed by a relational database, native JavaEE 5+ support, complex data structures, fine-grained security, WebDAV and CMIS interfaces, and fully open source. It implements an access control list based approach, combined with roles; user accounts can be assigned to any number of user groups. Access control lists –which are assigned to user groups– define a list (Read, Edit, Create, etc.) of permissions attached to an arbitrary object. Compared with CincoSecurity, this project uses 10 coarse roles with predefined permissions related to the administration module, while CincoSecurity lets to define any number of security profiles, each one as a set of fine roles related to the services of the application (not only to the security services). We believe it is more intuitive to associate the users to these security profiles and not to the [fleXive] Access control lists (ACL), that define lists of permissions attached to arbitrary objects. [fleXive] does not offer access control to elements of web pages, as CincoSecurity does.

With respect to recent proposals for extending the RBAC model, some research works as [22] [23] try to statically validate the correctness of roles usage in an application, for solving what they call the fragility of traditional dynamic checks. In [23], Fisher et al. argument that traditional RBAC does not easily express application-level security requirements. For instance, in a medical records system it is difficult to express that doctors should only update the records of their own patients. Further, traditional RBAC frameworks rely solely on dynamic checks, which makes application code fragile and difficult to ensure correct. They introduce ORBAC, a generalized RBAC model allowing roles to be parameterized by properties of the business objects being manipulated, with static validation of a program's conformance to an ORBAC policy. Centoze et al. [22] present a theoretical foundation for correlating an operation-based RBAC policy with a data-based RBAC policy. They have built a static analysis tool for Java EE that analyzes bytecode to determine if the associated RBAC policy is location consistent, and reports potential security problems.

CincoSecurity does not implement static checks, but its strategy of fine grained roles enables to automate the correct incorporation of security in a web application. In effect, by following the names discipline explained in this paper, it is possible to automatically add annotations to each method of the session EJB3 controlling a use case, in order to permit its access only to users having the associated fine role; it is also possible to automatically modify button tags of JSF pages for rendering them only to users having the corresponding fine role. This automation has been explained in section VIII.

On the other hand, it is important to note that Seam offers a complete security module [14], that is based on (coarse) roles, permissions and rules, that achieves a very flexible control of resources. The CincoSecurity module takes advantage of the Seam security by using some of its facilities related with authentication, restriction annotations for roles, and tags of JSF pages for rendering only for the appropriate role. However, we believe that for an administrator it is more difficult to write rules for granting fine permissions to roles (as is done in the Seam module), than to configure security profiles by checking the services of the application to be granted (as is done in the CincoSecurity module). Also, given that CincoSecurity does not use permissions nor rules (only fine grained roles), the incorporation of security to a web application can be automated, as it was explained above; with the Seam module it seems more difficult to automate the incorporation of security.

X. CONCLUSION AND FUTURE WORK

Needless to say that developing a secure Java EE application is a difficult job, where dozens of subtle details must be handled coherently. The CincoSecurity module provides a complete code baseline to develop a Java EE application with the Seam framework, incorporating, from the beginning of the development process, a full and flexible access control to the use cases and services of the application being developed. The CincoSecurity module also provides the use cases required to administer the users and their access

permissions to the use cases and services of the application being developed.

With respect to the Core RBAC model [7], an access permission is materialized in CincoSecurity as the right to invoke a method (service) of a business component. Fine grained roles are defined and implemented, each one having just one permission to invoke a single method (service) of a business component (session EJB3). There is also a fine grained role to allow entrance to each use case, as well as a fine grained role to grant access to each one of the services provided by the use case. The concept of "security profile" is defined and implemented as a set of fine grained roles.

The CincoSecurity module takes advantage of the low level access control principle implemented by any Application Server, by feeding the Application Server with the fine grained roles included in the security profiles the authenticated user belongs to. Additionally, the CincoSecurity module dynamically builds a customized menu containing only the entries leading to the application's use cases authorized for the user.

The CincoSecurity module is a Java EE 5 application built using the Seam framework [12] [13]. It is distributed under the GPL license and can be freely downloaded from [18]. CincoSecurity is used by several software houses in Colombia.

CincoSecurity module may be modified or extended to incorporate more complex security policies (e.g., elaborated policies for password handling), permissions to access the different attributes of a persistent entity, or to implement extended RBAC models (hierarchical roles, constraints).

As future work, CincoSecurity will be extended to further automate and simplify the incorporation and administration of the security of a web application, as well as to include other capabilities of the Seam security module, like Identity management, in a compatible way with our approach.

REFERENCES

[1] M. C. Franky, V. M. Toro, "CincoSecurity: A Reusable Security Module Based on Fine Grained Roles and Security Profiles for Java EE Applications", ICIW 2011: The Sixth International Conference on Internet and Web Applications and Services, St. Maarten-The Netherlands Antilles, March 2011, pp. 118-123, ISBN: 978-1-61208-004-8

[2] R. L. Baldwin, "Naming and Grouping Privileges to Simplify Security Management in Large Databases", Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy (Oakland, CA), IEEE Computer Society Press, pp. 116-132, 1990.

[3] D. F. Ferraiolo and D. R. Kuhn, "Role-Based Access Control". Proc. 15th Nat'l Information Systems Security Conf., Diane Publishing Company, pp. 554–563, 1992.

[4] R. Sandhu, C. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models". IEEE Computer Magazine, pp. 38-47, 1996.

[5] American National Standard for Information Technology – Role Based Access Control, ANSI INCITS 359-2004, 2004.

[6] B. Messaoud, "Access Control Systems: Security, Identity Management and Trust Models", Springer Science+Business Media, Inc., 2006.

[7] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, "Role Based Access Control", Artech House 2003, 2nd Edition 2007.

[8] N. Li, J. W. Byun, and E. Bertino, "A Critique of the ANSI Standard on Role-Based Access Control", IEEE Security and Privacy, Volume 5, Issue 6, pp. 41-49, 2007.

[9] D. Alur, J. Crupi, and D. Malks, "Core J2EE Patterns: best practices and Design Strategies", Sun Microsystems - Prentice Hall, 2001.

[10] Oracle, "The Java EE 5 Tutorial", http://docs.oracle.com/javaee/5/tutorial/doc/ 01.23.2012

[11] M. Keith and M. Schincariol, "Pro EJB 3: Java Persistence API", Apress, 2006.

[12] M. Yuan and T. Heute, "JBoss Seam: Simplicity and Power Beyond Java EE", Prentice Hall, 2007.

[13] D. Allen, "Seam in Action", Manning Publications Co., 2009.

[14] JBoss Seam Group, "Reference manuals of JBoss Seam", http://seamframework.org 01.23.2012

[15] JBoss Community, "JBoss Application Server", http://www.jboss.org/jbossas 01.23.2012

[16] Eclipse Open source development platform comprised of extensible frameworks, tools and runtimes, http://www.eclipse.org 01.23.2012

[17] General Public License, http://www.gnu.org/licenses/gpl.html 01.23.2012

[18] M. C. Franky, V. M. Toro, and R. López, "CincoSecurity Module", http://sourceforge.net/projects/cincosecurity 01.23.2012

[19] M. C. Franky, V. M. Toro, and R. López, "CincoModule: Módulo de seguridad basado en roles finos y en perfiles de seguridad para aplicaciones Java EE 5". Quinto Congreso Colombiano de Computación (5CCC), Cartagena-Colombia, Abril 2010. ISBN: 978-958-8387-40-6.

[20] "JPA Security", http://jpasecurity.sourceforge.net 01.18.2011

[21] D. Lichtenberger, M. Plesser, G. Glos, J. Wernig-Pichler, H. Bacher, A. Zrzavy, and C. Blasnik, "[fleXive]™ 3.1 Reference Documentation", Copyright © 1999-2010 UCS - unique computing solutions gmbh, http://www.flexive.org/docs/3.1/xhtml/index.xhtml 23.01.2012

[22] P. Centonze, G. Naumovich, S. J. Fink, and Marco Pistoia, "Role-Based access control consistency validation", Proceedings of the 2006 international symposium on Software testing and analysis (ISSTA'06). ACM, New York, NY, USA, pp. 121-132, 2006

[23] J. Fischer, D. Marino, R. Majumdar, and T. Millstein, "Fine-Grained Access Control with Object-Sensitive Roles", ECOOP 2009 – Object-Oriented Programming, Lecture Notes in Computer Science, Volume 5653/2009, pp. 173-194, 2009.

[24] J. Friedl, "Mastering Regular Expressions". O'Reilly, 2002.

[25] M. Habibi, "Java Regular Expressions: Taming the java.util.regex Engine". Apress Publishing, 2004.

[26] The Apache Software Foundation, "Apache Ant™ 1.8.2 Manual", http://ant.apache.org/manual 01.23.2012

[27] M. Richards, R. Monson-Haefel, and D. A. Chappell, "Java Message Service", O'Reilly, 2009.

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