JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

3/2012

(Contents Continued on Back Cover)

Editorial Board

Editorial Advisory Board

ISSN 1509-4553 on-line: ISSN 1899-8852

Circulation: 300 copies

Sowa – Druk na życzenie, www.sowadruk.pl, tel. 22 431-81-40

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

Trusted and secure computing properties of computer system are usually believed to behave in a well-defined way and enable the processed information to be properly protected. Nowadays, the need to protect information is increasing, particularly on the type of computers that we use directly for processing data with different levels of sensitivity. A computing system that supports multiple levels of security (MLS) provides the protections to guarantee that information which is assigned to different security levels is handled appropriately. The problem of processing information with different levels of sensitivity has been extensively studied since the early 70s of the twentieth century. To ensure the confidentiality and integrity of information, there are often used several models (B-LP, Biba, Clark-Wilson), which provide mandatory access control (MAC) entity (called subject) to the resource (called object).

The design of MLS systems guaranteed the correct performance with respect to security considerations, which is a daunting challenge. There are two main approaches to the construction of MLS systems: centralized and distributed one. Nowadays, the centralized (i.e., no distributed) approach to build computer system with multi-level security is mainly based on the virtualization technology for the separation of independent security domains as different virtual machines. The implementation of this type of MLS system design requires the integration of available virtualization technology (software and hardware), application of cryptographic protection and formal methods for both ensuring and control of the confidentiality and integrity of data, and advanced techniques for user authentication. Virtualization is now becoming more broadly available and is supported in off-the-shelf systems based on Intel and AMD architecture hardware. Virtualization can improve overall system security and reliability by isolating multiple software stacks in their own VMs. The application of some selected security concerned solutions for building a secure and trust environment called Secure Workstation for Special Applications (SWSA) is a key topic of the current issue. So, we have a developed method of secure MLS type system constructing an application of a RT-family trust management language for an access control model, a method of cryptographic protection of removable storage devices with USB interface, a design of hybrid cluster system for encryption and decryption of large amounts of data, an application of the Zak-Gabor-based iris coding to build a secure biometric verification station, designed a simple verification protocol for autonomous verification modules, and a solution that allows to combine hard drive encryption with a trustful boot process. Additional three papers present some results, applications and developments concerning the reduction of the traffic volume of user location data updating in a cellular

network, an experimental evaluation of YouTube video transmission examining the quality of experience of end user applications, and the method of optimal pump scheduling for large scale water transmission system by linear programming.

The problem of building a secure and trust specialized computer systems (SCS), which are processing data with different levels of sensitivity becomes particularly topical, especially in regard to the SCS applications in government institutions, military or financial. One of the essential questions concerning development of SCS involves a method of secure software designing. Z. Zieliński *et al*. present a component based approach to development secure SCS (SWSA) by means of integration of available software and hardware virtualization technology, application of formal methods, cryptographic protection of data stored on hard and removable disks and using biometrics techniques for user authentication. Also, an interesting approach to integration of security models with models of architecture of the system described in UML, which allows models simulation, has been proposed.

In the SWSA, multiple virtual machines simultaneously running (VMs) are used to process sensitive information from multiple security domains, providing strict separation of the domains. The users of SWSA may act in several different roles, with different access rights. The problem is how to control the access of SWSA users to particular VMs in the situation when users may have different periods of validity of different credentials. K. Lasota and A. Kozakiewicz have proposed an interesting solution of this problem which is based on an application of an RT-family trust management language, as a basis for an access control model to VMs. In prototype implementation of SWSA this model is mapped into a set of SELinux policy rules.

One of the necessary capabilities of Secure Workstation for Special Applications is cryptographic protection of hard and removable storage devices with USB interface. The use of solutions from ordinary systems with multilevel security (MLS), which include the SWSA, is insufficient. J. Chudzikiewicz and J. Furtak present a mechanisms to ensure an adequate level of protection of data stored on removable storage in MLS type workstation which is enabling to such a preparation of data stored in Flash RAM, so that the sender of data is assured that data will be available only for designated recipient, and the recipient is assured that the received data comes from the expected sender. In addition, the selected elements of these mechanisms implementation used in Windows operating system have been described.

Data encryption and decryption involve cumbersome calculations, especially when considering the processing of large amount of data. On the other hand, cryptography algorithms are natural candidates for massively parallel computations. E. Niewiadomska-Szynkiewicz *et al*. present a hybrid cluster system – a novel computing architecture with multi-core CPUs working together with many-core GPUs for encryption and decryption of large amounts of data. The experimental results presented in the paper demonstrate the effectiveness and scalability of such a cluster system.

In the next two papers the authors present the solutions for building a secure verification station acting as a server of biometric-based verification within secure workstation (SWSA) and consisting of a professional iris capture camera, a processing unit with specially designed iris recognition and a communication software. The iris recognition used in this work is based on the original methodology employing Zak-Gabor transformation. A. Czajka and K. Piech propose an automatic iris feature selection mechanism employing, among others, the minimum redundancy, maximum relevance (mRMR) methodology as one, yet more importantly, a step to assess the optimal set of wavelets used in this iris recognition application. The electronic communication between SWSA and the station is secured by a protocol that is specially designed to the purpose of such an application. A. Kozakiewicz and T. Pałka present the design and the rationale behind a simple verification protocol for autonomous verification modules.

A workstation may be regarded as secure only if it runs an original, unmodified software. The question is how we could assure that the workstation has not been modified in any way? M. Małowidzki *et al*. propose a solution that allows to combine hard drive encryption with a trustful boot process, preventing risk of software tampering. The logon process, which have been proposed, offers a reasonable level of security and could be increased by some additional mechanisms.

The key aspects in design of modern ‘ad hoc’ sensor networks are data security and energy aware communication. K. Daniluk and E. Niewiadomska-Szynkiewicz give in their paper an excellent survey of energy efficient security architectures and protocols for wireless sensor

networks. Also, the security requirements for wireless sensor networks are presented and the relationships between network security and network lifetime limited by often insufficient resources of network nodes are explained.

In mobile communications (e.g., GSM, UMTS, 3G, ...), the location of users may change in time. To make a communication between two users, the system must first find the location of destination user which must be extracted from databases. Thus, the most important criterion of a location tracking algorithm is to provide a small database access time. M. V. Dolama and A. G. Rahbar propose a new location tracking scheme, called Virtual Overlap Region with Forwarding Pointer, for cellular networks which reduces the updating information when a user frequently moves within the boundaries of several cells grouped into a Location Area.

Video sharing services like YouTube have become very popular recently. This situation results in a drastic growth of the Internet traffic statistic. On the other hand, when transmitting video content over packet based networks, stringent quality of service (QoS) constraints must be met in order to provide the comparable level of quality to a traditional broadcast television. A. Biernacki *et al*. conducted an experimental evaluation of YouTube video transmission (HTTP based) examining the quality of experience of end user applications expressed as a function of playback buffer occupancy.

The last paper of the issue is focused on the model for large scale potable water transmission system. J. Błaszczyk *et al*. describe in this paper a linear, the so-called Simplified Model (SM), based on mass-balance equations, which is solved on a week horizon and delivers boundary conditions for the so-called Full Model (FM) that is nonlinear and takes into account hydraulic phenomena and water quality.

Zbigniew Zieliński
Guest Editor

Paper

Secured Workstation to Process the Data of Different Classification Levels

Zbigniew Zieliński[a], Janusz Furtak[a], Jan Chudzikiewicz[a], Andrzej Stasiak[a], and Marek Brudka[b]

[a] *Military University of Technology, Warsaw, Poland*
[b] *FILBICO Ltd, Zielonka, Poland*

Abstract—The paper presents some of the results obtained within the ongoing project related with functional requirements and design models of secure workstation for special applications (SWSA). SWSA project is directed toward the combination of the existing hardware and software virtualization with cryptography and identification technologies to ensure the security of multilevel classified data by means of some formal methods. In the paper the requirements for SWSA, its hardware and software architecture, selected security solution for data processing and utilized approach to designing secure software are presented. The novel method for secure software design employs dedicated tools to verify the confidentiality and the integrity of data using Unified Modeling Language (UML) models. In general, the UML security models are embedded in and simulated with the system architecture models, thus the security problems in SWSA can be detected early during the software design. The application of UML topology models enables also to verify the fundamental requirement for MLS systems, namely the hardware isolation of subjects from different security domains.

Keywords—cryptographic protection, multilevel security, software design, UML, virtualization.

1. Introduction

The issue of building a reliable Specialized Computer Systems (SCS), which are processing data with different levels of sensitivity becomes particularly topical, especially in regard to the SCS applications in government institutions, military or financial. The problem of processing information with different levels of sensitivity has been extensively studied since the early 70s of the twentieth century [1]–[3]. Formal base of multilevel security (MLS) are presented in the work of Bell-LaPadula (B-LP) [2]. In the computerized system, which uses a multilevel security, it is necessary to determine users authorization (so-called security clearance) who work with classified information in accordance with the requirements of the missions' tasks (the rule of "necessary knowledge") and the classification of information, due to the required level of protection.

To ensure the confidentiality and integrity of information, there are often used models of B-LP, Biba [4], Clark-Wilson [5] which provide mandatory access control (MAC) entity (called subject) to the resource (called object). The mandatory access control of any entity (this can be a process) to resource (e.g., data file, the communication channel, etc.) is assigned to a security context. In order to determine entitlements in systems using MAC are designed labels, which contain the security context in particular pairs: *<sensitivity level, information category>*. On the set of labels of protected data partial order relation is defined, and to subjects and objects must be used invariable rules [3], [4], [6], among others, a rule prohibiting "writing down", a rule prohibiting "reading up". It should be noted that implementing the systems and networks' set of rules (that is, building a reliable system based solely on operating systems with multilevel protection of information) in the computer is difficult and expensive. This is mainly because of difficulties to build a reliable reference monitor and the difficulty of ensuring that in the system will not be the "leak" of sensitive information due to the possibility of the so-called covert communication channels in operating system [7].

Another approach to the construction of a centralized (ie, no distributed) computer system with multi-level security is to develop software in the virtualization technology [8]–[10] for the separation of independent security domains, called the Multiple Independent Levels of Security. Such software should allow for the simultaneous launch of several specific instances of operating systems on one PC (such as a workstation or server) designed to process data of different classification levels (e.g., public and proprietary), or to process the data in different systems for which there need for separation of data.

This approach has become today entirely possible thanks to the availability of solutions with virtualization hardware support in modern Intel and AMD processors, and developed software packages (COTS type) for virtualization. Now, widely used are the extension of the x86 architecture, designed to support hardware virtualization [8]–[10] as Intel Virtualization Technology in particular VTx, VTD for x86 processors, VTi for Intel IA-64 (Itanium), and AMD Virtualization (AMD-V) for 64-bit x86 processors from AMD. These technologies also allow (in addition to hardware support emulation of the virtual machine) for building a trusted environment in which a separate virtual machines (which are separate security domains) are performed in separated hardware partitions. The implementing of this type of design of MLS system requires the integration of available virtualization technology (software and hardware), application of formal methods for both ensuring and control of the confidentiality and integrity of data, and techniques for user authentication. A natural way to build such systems is component approach, which assumes the use of

ready and available hardware components and software, in particular virtualization packages available as open source like Xen [11] and KVM [12].
The paper presents the requirements for a Secure Workstation for Special Applications (SWSA), its hardware and software architecture, selected security solution for data processing and utilized approach to designing secure software. The developed method of manufacturing the type systems of MLS, which is defined as a Model Driven Multilevel Security (MDmls) method, organizes the process of producing the SCS of MLS type and is derived from the concept of Model Driven Architecture (MDA) [13], [14] and Model Driven Development (MDD) [15], [16]. A similar approach to build secure software is presented in [15], but it does not include multilevel protection issues. The integration of security models with models of systems described in UML enables the simulation which allows to verify the effectiveness of security the designed software of SCS type MLS at the stage of modeling.

2. SWSA Requirements

The technical solutions should be designed and prepared for strictly defined applications. This paradigm is particularly important when considering the high assurance equipment. The invalid definition of SWSA designation could lead to a significant increase in costs, realization time and the complexity of security functions, and as a result to reduce the chances of achieving the adequate quality, reliability and security assurance of the product. The first task in the project was therefore to consider the technology limits imposed by legal requirements and then to select the suitable usage scenarios.

2.1. SWSA Usage Scenarios

The most important discriminating factors for SWSA usage scenarios were adopted ad hoc, just to make the legal analysis of usage scenarios more thorough and systematic:

- differing classification levels of information which is processed within individual virtual machines: *single-level*, *multi-level*, and *international multi-level*, while the latter is not considered in the subsequent analysis;
- the number of users accessing SWSA categorized as *user-less* and *multi-user*; in user-less applications the Administrator (ADM) and the Security Officer (SO) only have access to the system, while in multi-user applications there are some additional users of the workstation with access rights other than ADM and SO;
- the connectivity of SWSA, which is recognized as: *stand-alone*, *local-area* and *wide-area*; the stand-alone SWSA has no access to any ICT networks; the locally connected SWSA accesses a local area network within a single security zone, while the wide-area connectivity implies that SWSA is connected to a wide area networks and may access multiple ICT networks located in different security zones.

The combinations of the aforementioned factors lead to different legal implications for related classes of SWSA usage scenarios. The analysis of these combinations allowed to arrange them in order of the increasing complexity of the most important implementations, or to be more specific, the anticipated complexity of obtaining the security approval during the accreditation:

- *single-level*, *multi-user* and *wide-area* class of usage scenarios. The workstation in these scenarios is connected to several networks processing information of the same security classification, but differing categories; an example of such application is the mutual access of SWSA to the networks of R&D and accounting departments;
- *multi-level*, *multi-user*, and *stand-alone* class of usage scenarios; example of such applications may be the stand-alone trusted workstation in classified information storage facility;
- *multi-level*, *user-less*, and *wide-area* class of usage scenarios, in which the SWSA could be the base platform for ICT security assurance solutions, e.g., to transfer the data between systems processing the information of differing levels of classification;
- *multi-level*, multi-user and *wide-area* class of usage scenarios; the security requirements and limitations for such applications are the most demanding when compared to any other class of applications; in general, these requirements and limitations regard the challenging threats for the confidentiality of information in multi-level, multi-user and distributed environments.

2.2. Legal Regulations and SWSA

The main conclusion of a survey on national regulations and guidelines in ICT security is the observation that these legal documents apply mainly to the "hardware" level of ICT, and do not contain any specific requirements for the virtualization. The lack of necessary regulations does not imply, however, the application of virtualization in the classified information systems is forbidden due to the well-known security principle of "what is not allowed – it is forbidden". While some changes of the legal status could help to implement such solutions, there is always the opportunity to obtain the approval for such applications within the system accreditation procedures. The next statements are the fundamental security terms and requirements for SWSA identified during the analysis of the Polish Classified Information Protection Act [17], regulations and security authorities guidelines.
The most important requirement is that the SWSA must ensure the protection adequate to the highest level of classification of the information processed within VMs. It is

assumed that due to the anticipated classification level such a security assurance of SWSA should to be evaluated on at least 4th level (EAL4) in accordance to Common Criteria methodology. It was also found that at least the compartmented security mode is adequate for applications demanding the security functions of SWSA. As a result, SWSA should provide the technical support for strict mandatory access control (MAC) policies.

In the *wide-area* applications, in which the distinct VMs are connected to local area networks of cooperating entities, SWSA should comply with the security requirements for system interconnection. In particular, SWSA must provide security measures to absolutely protect the confidentiality of the information. These security functions should be accompanied by proof of a controlled separation of SOS including the covert channels analysis.

The *wide-area* application imposes some specific requirements on the formal labeling and registration of each workstation and individual VMs. These terms are in a sense analogous to the deployment of the remote IT terminal within the security zone, which is not controlled by the system owner. The host system of SWSA comprising of the computer, hyper-visor and the host operating system should be labeled and registered as a part of ICT system in the deployment place. VMs, which are remotely attached to other ICT systems via wide area connections, should be labeled and registered as agreed between the cooperating entities and the system owners. The agreements may vary with respect to the ownership, administration duties and responsibilities, and even liabilities regarding VM. SWSA should therefore provide some technical and operational support regarding the registration and labeling of virtual machines as well as their backups.

The Classified Information Protection Act imposes also the obligations to implement the security measures to protect ICT equipments against the compromising electromagnetic emanations. In general, these obligations apply to the hardware part of the workstation, namely chassis, signal and power supply lines. In particular, there are some specific requirements on the separation of the signal and supply lines belonging to either the unclassified (BLACK) or classified (RED) parts of ICT system. However, these conditions do not imply any separation requirements for the parts of SWSA which process the classified information of different security level. It is therefore assumed that SWSA should host only either the RED or the BLACK VMs.

2.3. Specification of Functional Requirements

In the SWSA environment can distinguish three types of actors: the system administrator, security officer (hereinafter SO), SOS user (hereinafter user) (Fig. 1).

Security officer with the administrator and others are developing special security requirements of the system (SSRS), and safe operation procedures (SOP). The SSRS is identifing levels of security virtual machines installed in SWSA and permissions for actors (users). SL involves clause of

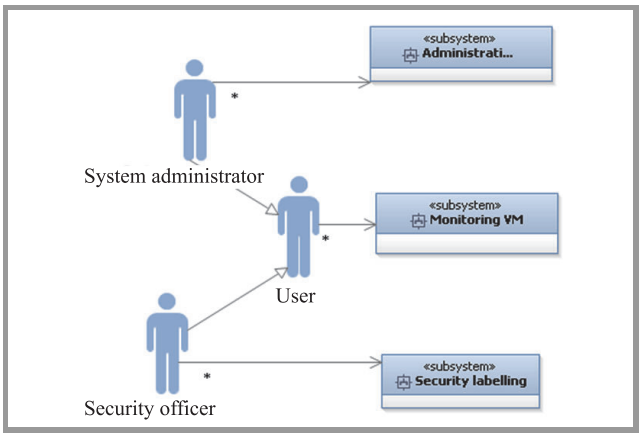


Fig. 1. General scheme of the basic elements of SWSA architecture.

information authorized to process and its set of information categories (range). Each level is described (according to Bell-LaPadula model) by a pair $<k, c>$ where $k \in K$ is the clause of information ($K = \{$public, proprietary, confidential, secret$\}$), and $c \in C$ is a subset of the categories of information $c = \{c_1, c_2, \ldots, c_L\}$. For example, $C = \{PD, GR, OS, DP, \ldots\}$, where the symbols PD, GR, OS, and DP denote the personal data, guidance resources, the operational situation, the data for purposes.

The clauses are ordered (from minor to major), $\forall_{i=(1,\ldots,4)}\, k_i \leq k_{i+1}$, but the categories are not. Security levels can be compared. For example, $SL_a = <k_a, C_a>$ and $SL_b = <k_b, C_b>$, if the following conditions: $k_b \leq k_a$ and $C_b \subseteq C_a$, then $SL_b \leq SL_a$ (SL_a level is higher or equal than the SL_b). Let $SL_b = <confidential, \{PD, GR\}>$, and $SL_a = <secret, \{PD, GR, OS\}>$, then we have a $SL_b \leq SL_a$, because according to satisfy the following: $confidential < secret$ and $\{PD, GR\} \subseteq \{PD, GR, OS\}$. Please note also that not all pairs of security levels are comparable. This leads to the use of the concept of lattice of security levels.

The security system according to Bell-LaPadula model is satisfied if the following axioms are preserved: security simple, stars, stability, security discretionary, non-availability of inactive object, the independence of the initial state. These axioms have been adopted in all models using mandatory access control to information. The fulfillment of these axioms ensures that classified information in the system will not be available for those who did not receive proper authorization. SO defines security attributes of SOS_k $(k = 1, 2, \ldots, n)$ existing in the SWSA, such as their clauses and classes of applications, manages the database of users and their security credentials, identifying opportunities to access resources for each of the domains. Access permissions to the domains are determined by security labels. Security officer gets access to the labels management and control of their allocation in the system through Virtual Machine Monitor (VMM).

The administrator performs backups of host machines and virtual machines, creates a new account for SSO users,

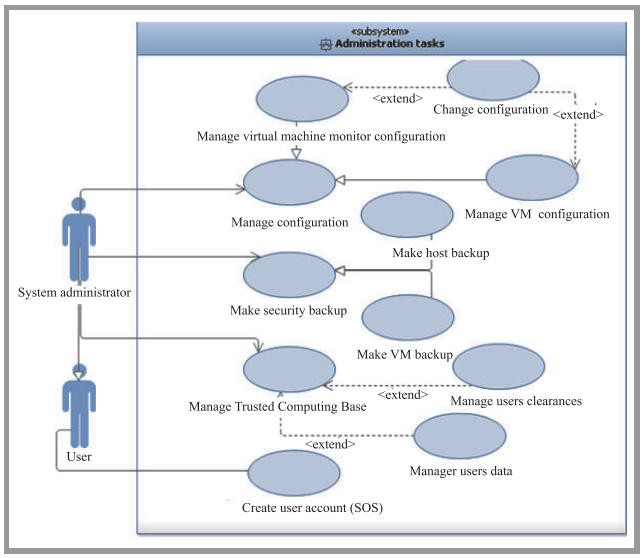


Fig. 2. Functional requirements for administration subsystem of SWSA.

creates and modifies the configuration of virtual machines (Fig. 2). In the model of requirements for VMM (Fig. 1), the user limits his task to run the SSO.

3. Architecture SWSA

It is essential for creating complex computer systems plays an architecture solution. With regard to the SWSA to be particularly important to recognize the hardware and software elements of the architecture due to their significant impact on the security of the system.

3.1. Hardware Architecture

It is assumed the use of components that enable hardware support for security technology and hardware virtualization support. To isolate the separate domains containing isolated environments implementing Trusted Execution Technology (TXT) will be used, while an important role in ensuring the integrity of the SWSA will play a Trusted Platform Module (TPM) that allows the secure creation and storage encryption keys.

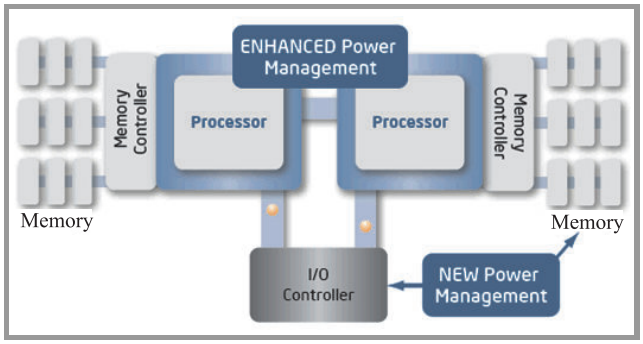


Fig. 3. Block diagram of a system based on the Xeon processor 5600 series.

Because of the applied approach (using ready components), hardware design limits itself to the configuration that describes the characteristics of the applied solution. Proposed hardware architecture of SWSA is based on a machine with an Intel Westmere with dual-processor Xeon E5630 model (Fig. 3), each of which contains the four processor cores. This architecture gives the possibility of sharing the responsibilities between the various hardware components, allows for the separation of the flow of information within the hardware and allows the separation of partitions with distinct security domains. Block diagram of a system based on the Xeon processor 5600 series is shown in Fig. 3. When designing the hardware architecture used UML with using topological models which construction is supported by a CASE environment – Rational Software Architect (RSA).

3.2. Software Architecture

In the architecture SWSA can be distinguished the following elements: the trusted system platform (TSP) and executable special versions of operating systems SOS. The general scheme of architecture SWSA is shown in Fig. 4.

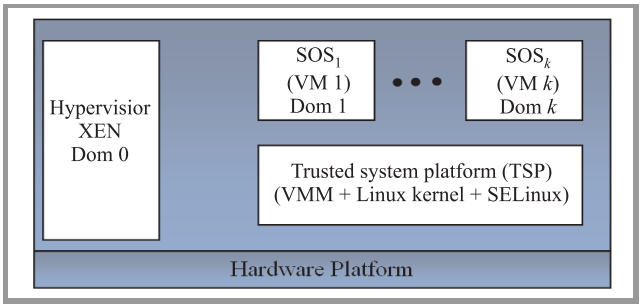


Fig. 4. General scheme of the basic elements of SWSA architecture.

The essential elements of an architecture SWSA are the SSP and virtual machines VM_i which we write in the form of the formula: $SWSA := SSP + \{VM_i\}$, where $i = \{1, \ldots n\}$. SSP consists of: kernel of Linux operating system, its extension in the form of a component (SELinux), and monitor virtual machines (*VMM*), *TSP* = *Linux_Kernel* + *SELinux* + + *VMM*. *TSP* allows you to run and supervise activities of specialized operating systems SOS_i $(i = 1, 2, \ldots n)$ and acting on their environment, application programs $\{AP\}$, forming a virtual machine, $VM_i := SOS_i + \{AP\}_i$.

The VMM is a key component of the SWSA, responsible for running virtual machines in accordance with defined security rules (using the hardware support) and their switching to ensure the separation of resources. It was assumed that the proposed VMM software should make it possible to simultaneously launch several (of many possible) instances of special versions of operating systems SOS_i on a single computer with the provision of: access control, separation of resources, cryptographic protection, and strict

control of data flow. The number of instances of virtual machines that is run, depends on the configuration of the station, in particular on the number of physical processors and cores. For example (Fig. 4), SSP supervises n virtual machines, among which operate both VM_1 and VM_2, and accordingly, on the VM_1 is active $\{AP\}_1$ (which is represented as $VM_1 \rightarrow \{AP\}_1$), on VM_2 is active $\{AP\}_2$ (which is represented as $VM_2 \rightarrow \{AP\}_2$). The VMM manages access to both virtual machines SOS_i, as well as to hardware resources (physical and virtualized).

The project also assumes that the instance of a special version of the operating system (SOS) working within a virtual machine (VM) is a separate Security domain (SD), $VM_i\ [SSO_i] = SD_i$. In each of the domains the processing of the data qualified to different security levels is allowed. Figure 4 shows two security domains, and each of them associated with one virtual machine.

It is worth noting that both the operating system kernel, as well as special versions of operating systems in terms of the project are ready components and their design will be limited only to the specifications of their interfaces and configuration descriptions. Interfaces were described in UML, and the configurations on the topological diagrams. In this area, the CASE tools (RSA) were used.

4. Cryptographic Protection of SWSA

Even in the simplest systems and applications, there are many places where the potential attack is possible, and their number is limited only by the inventiveness of the attacker. There are three basic areas in which information is exposed to capture:

- when you enter (e.g., keyboard);
- during transmission (e.g, via a local network or the Internet);
- when writing (e.g., on fixed and removable media).

In the area of interest of presented data protection solution, the SWSA belongs to the third area, including security of data stored on fixed and removable storage media.

The elaborated solution is assumed, among other things, the exchange of data between the internal, to the operating system, medium (ie hard drive), and external media (e.g., hard drive or Flash RAM) connected to the system via USB. However, the data exchanging between external media (e.g., Flash RAM) which are connected to the SWSA by USB is not possible. The process of securing the exchange of data using removable media should satisfy the following functional requirements:

- should be implemented in a manner transparent to the user;
- should not cause any noticeable to the user loads of the operating system;
- should not have a significant impact on the speed of read and write data onto data media;
- should allow to perform any operation allowed for data storage media, such as volume, surface checking for errors, and defragment the disk-based data.

These requirements force the use of the process of data security, a dedicated module (driver) for the operating system running at the kernel level. Schematic diagram of the developed solution is shown in Fig. 5. A detailed description of the method of securing removable media is given in [18].

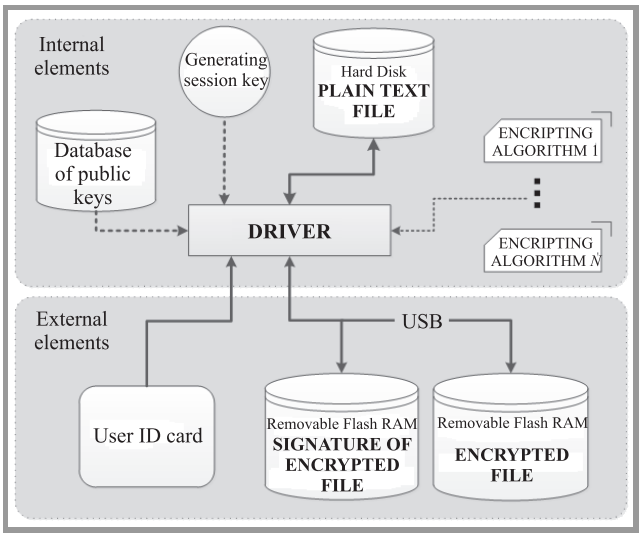


Fig. 5. Schematic diagram of securing of data stored on an external drive.

It is assumed that the elements involved in the process of securing stored data are divided into two groups: the internal and external components. The division has been completed, taking as its criterion, the relationship between the elements and the operating system installed on computer. The internal components (Fig. 5) includes both the hardware in a computer's hard drive, and software modules: the driver, `.DLL` library that provides the functionality of the implemented encryption algorithms, module of session key generation, and a database of public keys of users. External components (Fig. 5) connected to a computer via USB include hardware components in the form of Flash RAM, and the user identification card in the form of a smart card.

The process of writing (encryption) of data transferred to external media requires the user to identify the recipient of such a data. The process of data recording is initialized by the user (operator) currently logged on the system. The logged user may also be the recipient of the data. During the reading operations (decryption) the file, the operator is the recipient. For each of the saved file, the signature

is created that contains the information needed to decrypt the file. A signature is cryptographically protected, and could be read only by the recipient of the file.

5. The Method of SWSA Software Design

The key problem of SWSA software design boils down to building the trusted system platform (TSP) which includes: operating system kernel, virtual machines manager (VMM), and running special versions of operating systems. It is worth noting that both the operating system kernel, and special versions of operating systems in terms of the carried out project are ready components, and the project will be limited only to the specifications of their interfaces and configuration descriptions. Interfaces would be described in UML and the configurations on topological diagrams.

The essential complexity of software design is thus reduced to the construction of VMM virtual machines manager (Fig. 4), which will be responsible for its own implementation of the Xen hypervisor virtualization component, but it is worth noting that the work includes the implementation of its own unique solutions in this scope, in particular, it provides multilevel security policies.

In the process of development of VMM software, a new method of software design of MLS-type systems called MDmls was proposed [19] which is based on MDD (Model Driven Development) approach [13]. The method intended for designing specialized MLS-type systems contains, in particular, its basic processes, domain languages used, stages and development environment. The essence of the MDmls method is the integration of MLS security models with system design models expressed in UML-based language. Such integrated models with both a concrete notation and abstract syntax are called security design models [20].

In the MDmls method, the activities concerning of domain modeling languages DSM are essential. The construction of a new class of Domain Specific Language requires a metamodel created, because only on this basis the profile can be defined. The metamodel formalizes the structure of models, as well as scenarios, which represent possible instances of SWSA.

From the perspective of project management, the transition to the implementation stage takes place after completing modeling, which we build the next release of "tested models" in an incremental and agile way. Implementation, however, is carried out, but only after delivering the final version of the system model, and considerably makes up the result of automatic transformation of models into system code and descriptions of the required configuration. Therefore, the method assumes that all developed, in accordance with MDA, models are combined with transformations: speed manual or automatic (model-to-model [M2M], model-to-text [M2T]).

In this scope of design process, it is proposed to use the extended UML language with the so-called topological models [16], [21]. The creation of these models is supported by the CASE environment – Rational Software Architect (RSA). The IBM RSA ver. 8.0 extended environment was used with the Rational Software Architect Simulation Toolkit, with support for UML Action Language (UAL), which provides a subset of the specifications described in OMG with technologies fUML and ALF (Fig. 6).

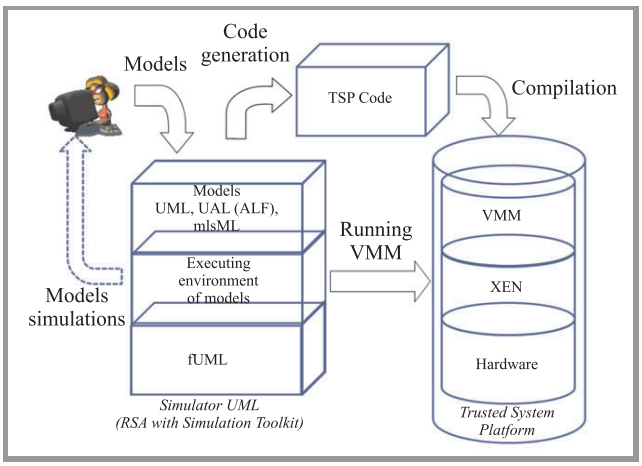


Fig. 6. The outline of environment for development of SWSA software [19].

The description of methods on how to validate the design solutions, specification of threat scenarios, and verification of the models in the RSA environment the subject of another publication.

Thanks to the integration of security models with the MLS-type SCS models (described in UML), in the proposed method of designing MDmls, the possibility of simulating models is obtained, which allows verification of many security problems of the designed system at the modeling stage. Using the model topology also allows the (physical) study of separating data processing processes belonging to different domains of security, which is one of the elements of verifying MLS-type systems. Additionally, the topology model enables defining the SWSA configuration, and then validate the rules concerning the exhaustion of station physical resources (resulting from its current configuration).

6. Summary

The main goal of the project that is to develop a secure environment for processing data of different classification levels on the same physical machine is achieved by integration of existing hardware and software virtualization, cryptography and identification technologies to ensure the security of multilevel classified data by means of some formal methods and components approach to provide different virtual machines with either Linux or Windows systems for each security level. The SWSA project is currently in the validation phase and its results are quite promising.

Acknowledgements

This work was supported by The National Center for Research and Development, Project OR00014011.

References

[1] J. P. Anderson, "Computer Security Technology Planning Study", vol. II, ESD-TR-73-51. Electronic System Division, Air Force System Command, L. G. Hanscom Field, Bedford, MA 01730, USA, Oct. 1972.

[2] D. E. Bell and L. J. La Padula, "Secure computer system: unified exposition and multics interpretation", ESD-TR-75-306, Bedford, MA: ESD/AFSC, Hanscom AFB [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf

[3] D. E. Bell, "Looking back at the Bell-La Padula model", in *Proc. 21st Ann. Comp. Secur. Appl. Conf. ACSAC 2005*, Tucson, AZ, USA, 2005, pp. 337–351.

[4] K. J. Biba, "Integrity Considerations for secure computer systems", Tech. Rep. MTR-3153, MITRE Corporation, Bedford, Massachusetts, USA, 1975.

[5] D. Clark and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", in *Proc. IEEE Symp. Secur. Priv. S&P 1987*, Oakland, California, USA, 1987, pp. 184–194.

[6] M. Brudka and J. Furtak, "Ponad barierami – łączenie sieci o różnych klauzulach", *Biuletyn IAiR*, no. 26, 2009 (in Polish).

[7] R. Smith, "Cost profile of a highly assured, secure operating system", *ACM Trans. Inform. Sys. Secur.*, vol. 4, no. 1, 2001, pp. 72–101.

[8] J. S. Robin and C. E. Irvine, "Analysis of the Intel Pentium's ability to support a secure virtual machine monitor", in *Proc. 9th USENIX Secur. Symp.*, Denver, Colorado, USA, 2000.

[9] C. E. Irvine *et al.*, "Overview of a high assurance architecture for distributed multilevel security", in *Proc. IEEE Sys. Man, Cybernet. Inform. Assur. Worksh.*, West Point, NY, USA, 2004.

[10] D. Kleidermacher, "Methods and applications of system virtualization using Intel® virtualization technology(Intel® VT)", *Intel® Technol. J.*, vol. 13, iss. 01, March 2009.

[11] P. Barhamet *et al.*, "Xen and the art of virtualization", University of Cambridge Computer Laboratory, CGF Brussels, 2004.

[12] A. Kivity *et al.*, "KVM: the Linux virtual machine monitor", in *Proc. Linux Symp.*, Ottawa, Ontario, Canada, 2007, pp. 225–230.

[13] D. S. Frankel, *Model Driven Architecture: Applying MDA to Enterprise Computing.* New York: Wiley, 2003.

[14] W. Dąbrowski, A. Stasiak, and M. Wolski, *Modelowanie Systemów Informatycznych w Języku UML 2.1*. Warszawa: PWN, 2007 (in Polish).

[15] T. Lodderstedt, D. A. Basin, and J. Doser, "SecureUML: a UML-based modeling language for model-driven security", in *Proc. 5th Int. Conf.*, *LNCS*, vol. 2460, 2002, pp. 426–441.

[16] "Planning deployment with the topology editor", IBM Tutorial, 2008.

[17] "Ustawa o ochronie informacji niejawnych", z dnia 5 sierpnia 2010, Dz.U. nr 182, poz. 1228 (in Polish).

[18] J. Chudzikiewicz and J. Furtak, "Cryptographic protection of removable media with a USB interface for secure workstation for special applications", *J. Telecom. Inform. Technol.*, vol. 3, pp. 22–31, 2012.

[19] Z. Zieliński, A. Stasiak, and W. Dąbrowski, "A Model Driven Method for Multilevel Security Systems Design", *Przegląd Elektrotechniczny* (*Electrical Review*), No. 2, 2012, pp. 120–125.

[20] D. Basin, M. Clavel, J. Doser, and M. Egea, "Automated Analysis of Security-Design Models", Preprint submitted to Elsevier, 2007.

[21] N. Makin, "Anatomy of a topology model in Rational Software Architect Version 7.5: Part 1: Deployment modeling", IBM, 2008.

[22] "Modeling deployment topologies", IBM Tutorial, 2008.

[23] N. Makin, "Anatomy of a topology model used in IBM Rational Software Architect Version 7.5: Part 2: Advanced concepts", IBM, 2008.

[24] S. Willard, *General Topology*. Courier Dover Publications, 2004.

[25] N. Li and J. C. Mitchell, "RT: A role-based trust management framework", in *Proc. 3rd DARPA Inform. Surviv. Conf. Exposition DISCEX III*, Washington, DC, USA, 2003, pp. 201–212.

[26] S. T. King, P. M. Chen, Y. Wang, C. Verbowski, H. J. Wang, and J. R. Lorch, "SubVirt: Implementing Malware with Virtual Machines", in *Proc. IEEE Sym. Secur. Priv. S&P 2006*, Berkeley, CA, USA, 2006.

[27] P. Ferrie, "Attacks on Virtual Machine Emulators", in *Proc. Association of Anti Virus Asia Res. Conf.*, Auckland, New Zealand, 2006.

[28] S. Mellor and M. Balcer, *Executable UML: A Foundation for Model-Driven Architecture*. Boston: Addison Wesley, 2002.

[29] M. Fowler and R. Parsons, *Domain Specific Languages*. Boston¿: Addison Wesley, 2010.

[30] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston: Addison Wesley, 2002.

[31] J. Jürjens, *Secure Systems Development with UML*. Berlin: Springer, 2010.

Zbigniew Zieliński received the M.Sc. in Computer Sciences from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1978, and the Ph.D. degree in Computer Systems from Military University of Telecommunication (St. Petersburg) in 1988. He is currently an Assistant Professor of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. His current research interests are in the areas of computer systems dependability, processors network diagnosis methods, fault-tolerant systems, as well as virtualization and system security.
E-mail: zzielinski@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Jan Chudzikiewicz received the M.Sc. in Computer Sciences from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1993, and the Ph.D. degree in Diagnosis of Computer Networks from the Cybernetics Faculty of Military University of Technology in 2001. He is currently an Assistant Professor

of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. From 1994 to 1998 he was cooperated with Industrial Institute of Electronics in domain of design of diagnostic systems for digital circuits. His current research interests are in the areas of diagnosis methods for computer systems, computer networks, and fault-tolerant systems, as well as the low-level software for the Windows systems.

E-mail: jchudzikiewicz@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Janusz Furtak received his M.Sc. from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1982. For eight years he was a member of the design team which developed software for command systems. Since 1990 he has been a university teacher at the Cybernetics Faculty of Military University of Technology. In 1999, he received Ph.D. degree in the field of Computer Science. Currently, he is an Assistant Professor in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology and Director of this Institute. His main areas of expertise are computer networks, network security, cyber defense and administering of network operating systems.

E-mail: jfurtak@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Andrzej Stasiak is an expert in the field of design of information systems. From years 2004–2012 he is a member of program committees of conferences on Software Engineering and Real Time Systems. From 1987 he is an Assistant Professor of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. He gained his professional experience directing some complex IT projects.

E-mail: astasiak@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Marek Brudka graduated in 1994 the Faculty of Electronics and Information Technology of Warsaw University of Technology. In 2000 he was awarded with honors a Ph.D. title for the dissertation on the intelligent robots control using neural networks and ultrasonic measurements. Since 2001 he is working for Filbico Ltd., currently as R&D manager. His professional experience comprises of the research and development as well as software development projects on robotics, command and control systems, crisis managements systems and ICT security.

E-mail: mbrudka@filbico.pl
Filbico Ltd.
Prymasa S. Wyszyńskiego st 7
05-220 Zielonka, Poland

Paper

Model of User Access Control to Virtual Machines Based on *RT*-Family Trust Management Language with Temporal Validity Constraints – Practical Application

Krzysztof Lasota and Adam Kozakiewicz

Research and Academic Computer Network (NASK), Warsaw, Poland

Abstract—The paper presents an application of an *RT*-family trust management language as a basis for an access control model. The discussion concerns a secure workstation running multiple virtual machines used to process sensitive information from multiple security domains, providing strict separation of the domains. The users may act in several different roles, with different access rights. The inference mechanisms of the language are used to translate credentials allowing users to access different functional domains, and assigning virtual machines to these domains into clear rules, regulating the rights of a particular user to a particular machine, taking into account different periods of validity of different credentials. The paper also describes a prototype implementation of the model.

Keywords*—RT-*family languages, security model, user access control, virtual environment.

1. Introduction

The article presents issues related to granting secure user access to resources having different sensitivity levels. This subject is one of the most important aspects of the project called "Special workstation for special applications". The project is focused on building a secure system to work with documents from different security domains located in virtual environment. Separation of system resources (e.g., Processors, RAM Memory, etc.) between virtual machines is not part of presented security model and it will not be described in this article. However, it was included in the architecture of the prototype solution.

The rest of this section presents a short description of the "Secure workstation for special application" project and the RT-family trust management languages, one of which is used to describe the proposed model. Section 2 focuses on presentation of the most important functional requirements. The proposed security model is presented in Sections 3, 4, and 5. Section 6 describes the implemented prototype. The article concludes with a short summary in Section 7.

1.1. Secure Workstation for Special Application

The work presented in this paper is a part of the project called "Secure workstation for special applications" [1], which aim is to create a secure environment for processing of sensitive information based on virtualization technology. The documents belonging to different security domains (different sensitivity levels or functional domains) are processed in separate, isolated, virtual machines running special secure versions of guest systems. The expected result of the project, to be delivered later this year, is an advanced technology demonstrator. The products of the project will include:

- secure system platform, referenced as SSP – software component integrating a secure host operating system and virtual machine management tools, which is able to run several instances of guest operating system;
- special versions of guest operating systems – secure version of Linux and Windows systems prepared to run under control of the secure system platform;
- technical and operational documentation of the system, recommendations, procedures and templates;
- examples of cryptographic data protection and authentication mechanisms, e.g. biometrics.

The project consortium is led by the Military University of Technology in Warsaw and consists of Filbico Sp. zo.o., Military Telecommunications Institute and Research and Academic Computer Network (NASK).

1.2. RT-Family Trust Management Language

Role-based trust management (RT) languages were introduced in [2] and combine features of trust management [3] and Role Based Access Control [4]. They are used for representing security policies and credentials in centralized and distributed access control systems. A credential provides information about the user access privileges and the security policies issued by one or more trusted authorities. So far the family consists of: RT_0, RT_1, RT_2, RT^T, RT^D languages [2], [5]–[7] which are progressively increasing in expressive power and complexity. For language RT^T which is backwards compatible with RT_0, RT_1 and RT_2, in [8] time validity of credentials is proposed. The extended versions of these languages are referenced as RT^T_+,

RT_{2+}, RT_{1+}, and RT_{0+}. In [9] the general structure of proofs of soundness and completeness of inference systems for these languages are presented. The complete proof is still waiting for publication.

While legal regulations enforce only using mandatory access control (MAC [10], [11]), the proposed model of user access control to virtual machines will be based on a trust management language from the RT family with time validity, enabling more detailed specification of access rules. Language RT_{2+} fits our expectations by supporting parameterized roles and defining o-sets which are able to group objects representing resources in much the same way as roles group subjects. This enables completely abstract policies to be created where permissions are specified in terms of roles, and o-sets and actual relations between subjects and objects are established by assigning them to roles and o-sets.

2. Security Model – Requirements

The basic requirement for the defined model of access control to virtual machines is its validity. Additionally it has to comply with functional requirements of the project. The most important ones are described in this section.

2.1. Security Clauses

The security model must support various levels of security which are assigned to system users and documents located on virtual machines. Figure 1 shows a diagram of relations between security levels in Poland, European Union and NATO.

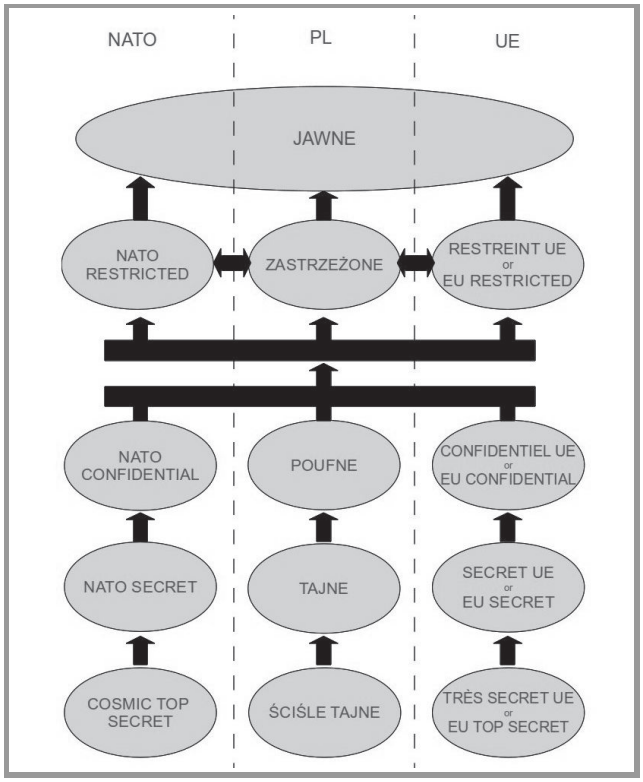


Fig. 1. Relations between security levels in Poland, European Union and NATO.

2.2. Protected Resources

Protected resources, e.g., documents, are located inside virtual systems. The security model for user access to virtual machines presumes that all protected resources are located on virtual machines. A permission is granted to access a particular virtual machine which holds documents with the same sensitivity level. A user in the system can have access to many virtual machines, as well as one virtual machine can be accessed by many users.

2.3. Functional Domains

The security model has to support various functional domains. Domain partitioning allows for more elastic management of system resources and user permissions. Examples of security domains might include: finance – this domain holds information related to payrolls, incomes of employees, etc., or projects – the domain holds information about running projects.

Additionally, the security model should support separation of resources located in the same functional domain but with different sensitivity levels. Security domain can hold resources with different sensitivity levels assigned to them, but a user can only access those with sensitivity level not exceeding his clearance.

2.4. Users of the SSP System

The security model has to differentiate users based on the potential system utilization by them. A basic system user can only use resources which he is allowed to access. From the system security point of view, the people responsible for granting other users rights to use the system are the most important group. Additionally, due to the accountability requirement for the secure workstation, it is required to include users responsible for auditing the system.

2.5. Time Validity

The security model should include information about validity period of the issued credentials. Implementing time frames of validity for granted permissions is necessary in the system. Security clearance and certificates of received trainings considering data protection are valid only for specified periods of time. The procedures for assigning access rights to protected resources for a user need to enforce a definition of their validity period, so that the model will be similar to a real system.

3. Security Model – Subjects and Objects

Security model is based on two types of entities – a secure workstation and system users – and the same number of objects corresponding to the resources of the system – virtual machines and shared storage areas.

Secure workstation is a central point of authority. Only roles and sets defined by the Secure Workstation and credentials issued by it are supported by the deductive system. The most important attribute of an entity is a *unique identifier* that is referenced as B later in the article.

System User is a key entity. Based on issued credentials, protected system resources are made available to it. User entity possesses a number of attributes that are derived from requirements for security model:

- *Unique identifier* – referenced as u_{id} – used to differentiate between users;
- *Security levels* – referenced as u_{lev} – contains a list of all sensitivity levels which the user has access to.

Additionally each clause has a defined time period of its validity. The validity depends on the clearance held by the user, e.g., in Polish law validity is determined by owned security credentials and completed trainings on protection of classified information. The time period of certificate validity is referenced as v_l, where l corresponds to the identifier of security level.

Virtual machine and **shared storage area**. All resources of the secure workstation to which access is determined by the defined model are contained in the described objects. The main purpose of virtual machines is to allow for working with documents of different security level. Shared storage areas provide access to resources that are located directly on the secure workstation. They enable different access to be assigned to directories or hard drives for different users. In particular, a separate hard drive destined for collecting logs is intended and available only for the safety auditor. The most important attributes of an object are:

- *Unique identifier* – referenced as mv_{id} and ss_{id} – used to differentiate between objects.
- *Validation period* – referenced as v_{mv} and v_{ss} – period of time when particular object is available for systems users.
- *Security level* – referenced as mv_{lev} and ss_{lev} – attribute specifies security level of all resources contained in a particular object.

4. Security Model – Roles and O-Sets

The RT_1 is an extention of RT_0 and introduces parameterized roles, while the RT_2 language extends RT_1 with o-sets. In the following section all roles and o-sets used by the access control model are described.

4.1. Parameters

All defined roles for grouping subjects, and also all object sets can depend only on the following parameters:

User role parameter – referenced as *rol* – has a slightly different meaning depending on the context where it is used, either referencing security model roles or o-sets. In case of security model roles, it defines a role for the system user. Second case is when it describes a user's role necessary to access protected resources (either a virtual machine or shared storage area). The parameter can be assigned the following values:

- *USER* – references users with basic rights in the system which can get access only to resources assigned to them;
- *SPEC* – references Chief Information Security Officer (security specialist) with function of granting users access rights to resources, but does not have an access to these resources himself. This is the only role which can add new credentials interpreted by the system. In fact, a user with this role can be viewed as the host system's administrator;
- *AUDIT* – references Chief Audit Executive (security auditor) whose role is assuring system accountability by validating system event log files;
- *ADMIN* – additional role for virtual machine administrators responsible for the configuration of virtual machines.

Functional domain parameter – referenced as *dom* – corresponds to system partitioning into separate domains allowing an easy management of users and resources. A user gains access to resources by acquiring proper credentials to access a domain. A particular resource is accessible only if it is assigned to a domain. The parameter does not have a defined set of allowed values. The only restriction is a unique identifier of a newly added domain.

Type of access parameter – referenced as *rig* – the security model provides means to determine the access type granted for a user to a resource. Two basic types of access rights are distinguished: a right to read referenced as *R* and a right to read and write referenced as *RW*. The first type is applicable in case of shared storage areas containing log files of system events. A system administrator should have an access to it, but should not have a right to modify it. An auditor has the ability to create a backup of secured data and delete the old one by using an additional software. In case of resources on virtual machines, only the second type of access rights is used.

Security level parameter – referenced as *lev* – The security model requires specifying a sensitivity level for each

resource. The resource may not be accessed by users without a security clearance for this – or higher – level of security. The following security level identifiers are used to reference proper security clearances used in Poland, EU and NATO:

- *J* – public level, common for security clearance in Poland, EU and NATO;
- *Z* – restricted level, references: Polish *'ZASTRZEZONE'*, and *'EU RESTREINT'* or *'EU RESTRICTED'* and *'NATO RESTRICTED'*;
- *P-PL* – confidential level, references Polish *'POUFNE'* clearance;
- *P-EU* – confidential level, references *'EU CONFIDENTIAL'* or *'CONFIDENTIEL UE'* security clearance;
- *P-NA* – confidential level, references *'NATO CONFIDENTIAL'* security clearance;
- *T-PL* – secret level, references Polish *'TAJNE'* clearance;
- *T-EU* – secret level, references *'SECRET UE'* or *'EU SECRET'* clearance;
- *T-NA* – secret level, references *'NATO SECRET'* clearance;
- *S-PL* – top secret level, references Polish *'SCISLE TAJNE'* clearance;
- *S-EU* – top secret level, references European *'EU TOP SECRET'* or *'TRES SECRET UE'* clearance;
- *S-NA* – top secret level, references *'NATO COSMIC TOP SECRET'* clearance.

4.2. User Roles

The security model allows for grouping of the entities related to users in three roles. Role *B.ide* of the security model contains information about all system roles that can be assigned to users, and is defined as follows:

$$B.ide(?rol) \text{ in } V, \tag{1}$$

where:
rol – may take all defined values.

It is very important in the context of real system operations. Taking into account that a user can be assigned to different roles, identifying a particular one is done by user 'Identity' (denoted as ide in role names). The Identity is defined as a combination of user and assigned role, and typically corresponds to an account in the system – a user may have more than one account and change roles by switching between them. From the security point of view, assigning to the same user *AUDIT* role and any other is forbidden. Next two roles, defined as:

$$B.ide\_dom\_rig(?rol, ?dom, ?rig) \text{ in } V \tag{2}$$

and

$$B.ide\_dom\_lev(?rol, ?dom, ?lev) \text{ in } V, \tag{3}$$

where:
rol – may take values from set {*ADMIN,USER,AUDIT*},
dom – may take all defined domains,
rig – may take all defined values {*R,RW*},
lev – may take all defined values {*J, Z, P-PL, P-EU,* {*P-NA, T-PL, T-EU, T-NA, S-PL, S-EU S-NA*},

are holding information about an access of user identities to functional domains. Role (2) determines the type of an access to domain resources and role (3) defines restriction on the highest sensitivity level of resources which a given identity is allowed to access. User identity can only access those resources with sensitivity level not exceeding clearance.

4.3. Virtual Machine O-Sets

The security model allows to group objects related to virtual machines in two ways. The first one (4) defines membership of a virtual machine to resources of a particular functional domain. Sensitivity level of data located on a virtual machine is an attribute of an object representing the virtual machine. The assignment is performed with a specific security level which may not differ from the sensitivity level assigned to the virtual machine itself.
The second one (5) describes a system role which a user has to be assigned to gain access to a particular virtual machine.

$$B.mv\_dom(?dom, ?lev) \text{ in } V \tag{4}$$

where:
dom – may take all defined domains,
lev – may take all defined values.

$$B.mv\_rol(?rol) \text{ in } V \tag{5}$$

where:
rol – may take values from set {*ADMIN,USER,AUDIT*}.

4.4. Shared Storage Area O-Sets

Similarly as for virtual machines, the security model allows two groupings of objects related to shared storage areas. The first one (6) defines functional domain with a particular sensitivity level that has the shared storage area in its resources. The second one (7), aside from defining a system role a particular user has to be granted with, defines the type of access. There is no possibility to assign shared storage area as a resource designed for users in *USER* system role.

$$B.ss\_dom(?dom, ?lev) \text{ in } V \tag{6}$$

where:
dom – may take all defined domains,
lev – may take all defined values.

$$B.ss\_rol(?rol) \text{ in } V \tag{7}$$

where:
rol – role – may take values from set {*ADMIN,AUDIT*},
rig – rig – may take all defined values.

4.5. Main Role

The most important role (8) of the security model can provide information about specific time when user is allowed to access a particular resource, based on user credentials and deductive system.

$$B.main(?rol, ?dom, ?rig, ?lev) \text{ in } V \tag{8}$$

where:
rol – may take values from set {*ADMIN,USER,AUDIT*},
dom – may take all defined domains,
right – may take all defined values,
lev – may take all defined values.

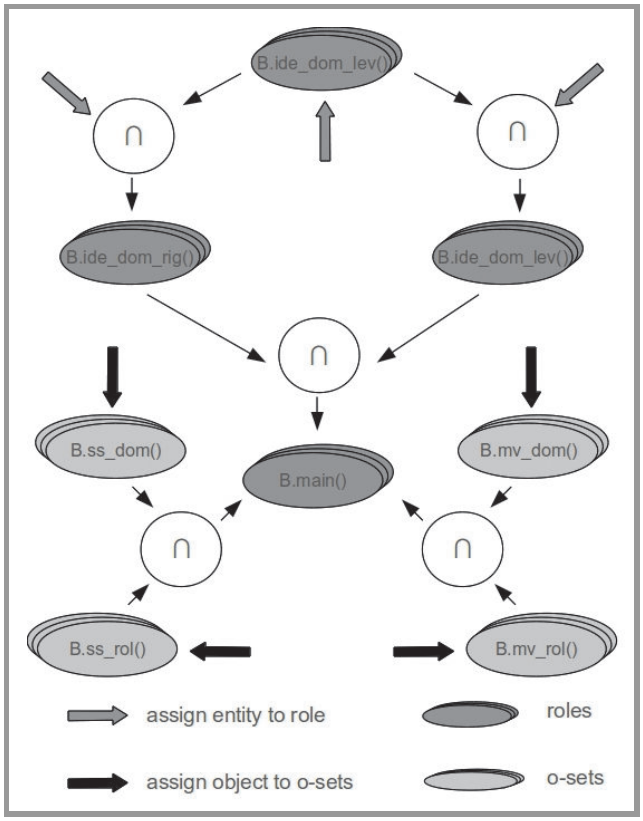


Fig. 2. Visualization of user access control model.

5. Security Model – Credentials

A security model based on the RT_{2+} language featuring time validity of credentials is presented in Fig. 2. It is composed of roles and o-sets, as defined in the previous section. Furthermore, it presents all possible credentials the system is able to process. Two types of credentials were distinguished: credentials created by an Information Security Officer, and credentials obtained from the deductive system process.
The first type of credentials includes:

- a user role attribution, creating a user identity (9);
- an identity access type to resources of a particular functional domain (10);
- an identity access to resources of a functional domain with particular sensitivity level (11);
- a virtual machine inclusion as a resource of a particular domain with specific sensitivity level (13);
- a virtual machine attribution to users holding specific system role (14);
- a shared storage area inclusion as a resource of a particular domain with specific sensitivity level (16);
- a shared storage area attribution to users holding specific system role and type of access (17).

The second type of credentials include:

- a type of access allowed to resources of a domain with particular sensitivity level for a user holding specific role (12);
- a virtual machine inclusion as a resource of a domain with a particular sensitivity level, with access type available for a user holding a certain role (15);
- a shared storage area inclusion as a resource of a domain with a particular sensitivity level, with specific access type available for a user holding a certain role (18).

5.1. User Credentials

$$\begin{gathered} B.ide(rol = X) \text{ in } V \\ \longleftarrow \\ B.USER(u\_id = I) \text{ in } v \end{gathered} \tag{9}$$

where:
X – role value which is assigned,
I – value of user unique identifier,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.ide\_dom\_rig(rol = X, dom = Y, rig = Z) \text{ in } V \\ \longleftarrow \\ B.ide(rol = X) \\ \cap \\ B.USER(u\_id = I) \text{ in } v_{ide} \cap v \end{gathered} \tag{10}$$

where:
X – role value which is assigned,
Y – domain value which is assigned,
Z – access type value which is assigned,
I – value of user unique identifier,
v_{ide} – active time validity of component credential (9),

v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.ide\_dom\_lev(rol = X, dom = Y, lev = W) \ \text{ in } V \\ \longleftarrow \\ B.ide(rol = X) \\ \cap \\ B.USER(u\_id = I, u\_lev = L : W \in L) \\ \text{in } v_{ide} \cap v_{WL} \cap v \end{gathered} \tag{11}$$

where:
X – role value which is assigned,
Y – domain value which is assigned,
W – sensitivity level which is assigned,
L – set of all security levels accessible to a particular user,
v_{ide} – active time validity of component credential (9),
v_{WL} – time validity of user sensitivity level,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.main(role = X, dom = ?Y, rig = ?Z, lev = W) \\ \text{in } V \\ \longleftarrow \\ B.ide\_dom\_rig(role = X, dom = ?Y, rig = ?Z) \\ \cap \\ B.ide\_dom\_lev(role = X, dom = ?Y, lev = W) \\ \text{in } v_{rig} \cap v_{lev} \end{gathered} \tag{12}$$

where:
X – role value which is assigned,
Y – set of domain values which are assigned,
Z – set of access type values which are assigned,
W – security level value which is assigned,
v_{rig} – active time validity of component credential (10),
v_{lev} – active time validity of component credential (11),
V – active time validity of credential,

5.2. Virtual Machine Credentials

$$\begin{gathered} B.mv\_dom(dom = Y, lev = W) \text{in } V \\ \longleftarrow \\ B.MV(mv\_id = M, mv\_lev = W) \ \text{ in } v_{mv} \cap v \end{gathered} \tag{13}$$

where:
Y – domain value which is assigned,
W – security level value which is assigned,
M – value of virtual machine unique identifier,
v_{mv}– time validity of virtual machine,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.mv\_rol(role = X) \\ \text{in } V \\ \longleftarrow \\ B.MV(mv\_id = M) \ \text{ in } v_{mv} \cap v \end{gathered} \tag{14}$$

where:
X – role value which is assigned,
M – value of virtual machine unique identifier,
v_{mv}– time validity of virtual machine,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.main(role = ?X, dom = ?Y, rig = ' RW', lev = W) \\ \text{in } V \\ \longleftarrow \\ B.mv\_dom(dom = ?Y, lev = W) \\ \cap \\ B.mv\_rol(role = ?X) \\ \text{in } v_{dom} \cap v_{rol} \end{gathered} \tag{15}$$

where:
X – set of role values which are assigned,
Y – set of domain values which are assigned,
W – security level value which is assigned,
v_{dom} – active time validity of component credential (13),
v_{rol} – active time validity of component credential (14),
V – active time validity of credential.

5.3. Shared Storage Area Credentials

$$\begin{gathered} B.ss\_dom(dom = Y, lev = W) \text{ in } V \\ \longleftarrow \\ B.SS(ss\_id = S, ss\_lev = W) \ \text{ in } v_{ss} \cap v \end{gathered} \tag{16}$$

where:
Y – domain value which is assigned,
W – security level value which is assigned,
S – value of shared storage area unique identifier,
v_{ss} – time validity of shared storage area,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.ss\_rol(role = X, rig = Z) \\ \text{in } V \\ \longleftarrow \\ B.SS(ss\_id = S) \ \text{ in } v_{ss} \cap v \end{gathered} \tag{17}$$

where:
X – role value which is assigned,
Z – access type value which is assigned,
S – value of shared storage area unique identifier,
v_{ss} – time validity of shared storage area,
v – base time validity of credential,
V – active time validity of credential.

$$\begin{gathered} B.main(role = ?X, dom = ?Y, rig = ?Z, lev = W) \\ \text{in } V \\ \longleftarrow \\ B.ss\_dom(dom = ?Y, lev = W) \\ \cap \\ B.ss\_rol(role = ?X, rig = ?Z) \\ \text{in } v_{dom} \cap v_{rol} \end{gathered} \tag{18}$$

where:

X – set of role values which are assigned,
Y – set of domain values which are assigned,
Z – set of access type values which are assigned,
W – security level value which is assigned,
v_{dom} – active time validity of component credential (16),
v_{rol} – active time validity of component credential (17),
V – active time validity of credential.

6. Prototype

The software implementing the adopted user access control model can be divided in two parts: server and client side. The architecture of the software divided into functional modules is presented in Fig. 3. The server part of

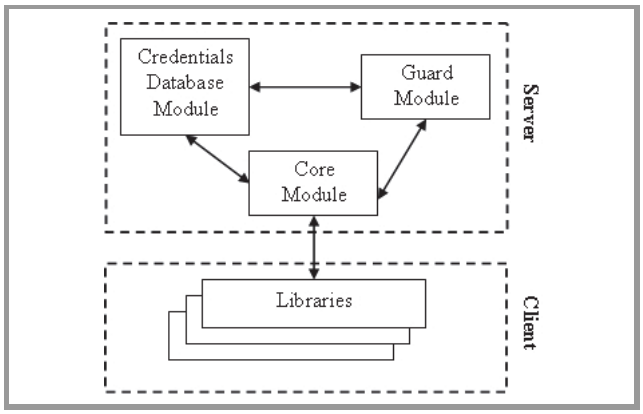


Fig. 3. Architecture of prototype solution.

the software is responsible for handling client requests according to adopted security model, and it is divided into three modules:

- *The Core* – responsible for handling communication with clients and other server modules.
- *The Credential Database* – provides functionality of knowledge base about possible actions of the SSP system users.
- *The Guard* – responsible for translating existing credentials located in the knowledge base into SELinux rules, effectively granting actual access to resources for running applications to users of the secure workstation.

The client part is implemented as a set of dedicated libraries used by all applications, requiring communication with the security model provider.

6.1. Server – Core Module

The Core module is responsible for communication with client applications and validation of incoming requests for access and modification of user access rights to resources. Its purpose is also to confirm a connecting users right to access, and to modify data located in the Credential Database. It coordinates cooperation between the rest of the server side modules implementing the access control model: the Guard and the Credential Database. It was implemented in C++ language using the functionality of an XML-RPC server library.

6.2. Server – Guard Module

The Guard Module is responsible for implementing rules of granting access to resources for applications and users of the secure workstation. The rules are derived from credentials located in the credential database and implemented with functionality of SELinux. The temporal validity of credentials allows the Guard module to use Linux cron daemon for issuing and revoking access rights at specific points of time. The SELinux policy used by SSP is based on a modified and extended Multi Level Security (MLS) policy.

6.3. Server – Credential Database Module

The Credential Database module represents the adopted security model. It provides the functionality of a knowledge base of allowed actions in the SSP system performed by the users. It consists of two elements: API and a database. Communication is implemented via an API implemented in C++. Database used in the system is PostgreSQL. Access to information in the database is restricted by the control mechanism available in PostgreSQL, based on the model Role Base Access Control (RBAC). Requests coming from different versions of the library, prepared for specific applications are handled by the Core Module on a separate restricted connection.

6.4. Client – Libraries

The client part of the software is composed from several versions of libraries designed for applications requiring communication with the Core Module of the server. The libraries provide interface for reading and modifying credentials in the Credential Database. The functionality shared by the libraries corresponds to the API of the Credential Database module. The libraries were implemented in C++ with the same XML-RPC library as the Core Module. Keys required for digital signing of messages are obtained from the key database protected by the Trusted Platform Module (TPM). The set of shared functions of a particular library corresponds to the actual purpose of the library. There are five versions of the library:

- *Basic version* – intended for the application used to login, allows validating the logged-in user role.
- *User version* – intended for applications with active USER role. It only provides ability to check which resources are accessible by the logged-in user.

- *Admin version* – intended for applications dedicated for administrators of virtual machines. It allows checking which resources are accessible by the logged-in user and obtaining information required for correct configuration of the system, e.g., which users have access to the virtual machine.
- *Spec version* – intended for Information Security Officers. It allows for unrestricted reading and modification of data in the Credential Database.
- *Audit version* – intended for Chief Audit Executives, it allows for unrestricted read-only access to data in the Credential Database and has access to history of changes.

7. Conclusion

The article presents a security model of user access control to virtual machines, which complies with functional requirements of the "Special workstation for special applications" project. Additionally, it includes new functionality, like introduction of a new SSP user role – Administrator of virtual machines or adding shared storage area as a separate resource with controlled access. A presentation of a fully operational prototype software, implementing presented security model underlines the model's usability in real applications. The paper presents the access control module's architecture and a short description of all defined modules.

The Guard module is of particular interest, as it shows how complex access models can be mapped in a practical way into simple rules for the well known SELinux solution. *RT*-family languages can be even more powerful and complex (see, e.g., RT_+^T), and they are very well suited for large, distributed systems with many independent centers of authority providing users with credentials, with complex trust relations between them. The presented architecture is adaptable to this setting, showing that local security mechanisms, such as SELinux, are still applicable in such distributed systems. The necessary approach is to infer simple access rights from the *RT* credentials. These rights can be applied in an automatic way, even taking into account limited periods of validity of credentials.

Acknowledgements

This work is part of the project called "Secure workstation for special applications" and is funded by a grant number OR00014011 from the National Center for Research and Development – science funding for years 2010-2012.

References

[1] A. Kozakiewicz, A. Felkner, J. Furtak, Z. Zieliński, M. Brudka, and M. Małowidzki, "Secure workstation for special applications", in *Secure and Trust Computing, Data Management, and Applications*, C. Lee, J.-M. Seigneur, J. J. Park, R. R. Wagner, Eds., Communications in Computer and Information Science, vol. 187. Berlin: Springer, 2011, pp. 174–181.

[2] N. Li, J. Mitchell, and W. Winsborough, "Design of a role-based trust-management framework", in *Proc. IEEE Symp. Secur. Priv.*, Oakland, CA, USA, 2002, pp. 114–130.

[3] A. Felkner, "Modeling trust management in computer systems", in *Proc. IX Int PhD Worksh OWD 2007, Conf Archives PTETiS*, Wisła, Poland, 2007, vol. 23, pp. 65–70.

[4] D. Ferraiolo and D. Kuhn, "Role-based access control", in *Proc. 15th Nat. Comp. Secur. Conf.*, Barltimore, USA, 1992, pp. 554–563.

[5] N. Li and J. Mitchell, "RT: A role-based trust-management framework", in *Proc. 3rd DARPA Inform. Survivability Conf. Exp.*, Washington, DC, USA, 2003, pp. 201–212.

[6] N. Li, W. Winsborough, and J. Mitchell, "Distributed credential chain discovery in trust management", *J. Comput. Secur*, vol. 1, pp. 35–86, 2003.

[7] A. Felkner and K. Sacha, "Deriving RT^T credentials for role based trust management", *e-Informatica Softw. Engin. J. (ISEJ)*, vol. 4, pp. 9–19, 2010.

[8] A. Felkner and A. Kozakiewicz, "Time validity in role-based trust management inference system", in *Secure and Trust Computing, Data Management, and Applications*, C. Lee, J.-M. Seigneur, J. J. Park, and R. R. Wagner, Eds., Communications in Computer and Information Science, vol. 187. Berlin: Springer, 2011, pp. 7–15.

[9] A. Felkner and A. Kozakiewicz, "Czasowa ważność poświadczeń języka RT_+^T", *Studia Informatica*, vol. 32, pp. 145–154, 2011 (in Polish).

[10] D. D. Bell and L. J. La Padula, "Secure Computer System: Unified Exposition and Multics Interpretation", ESDTR-75-306, Bedford, MA: ESD/AFSC, Hanscom AFB, 1974 [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf

[11] D. E. Bell, "Looking back at the Bell-La Padula model", in *Proc. 21st Ann. Comp. Secur. Appl. Conf. ACSAC 2005*, Tucson, AZ, USA, 2005, pp. 337–351.

Krzysztof Lasota works as Research Associate at Network and Information Security Methods Team in the Research Division of NASK. He received his B.Sc. (2010) and M.Sc. (2012) degrees in Telecommunications from Warsaw University of Technology, Faculty of Electronics and Information Technology. Currently he is a Ph.D. student there. He is co-author of several publications. He participated in security-related projects at NASK, including FISHA and HoneySpider Network. Currently he participates in the project "Secure workstation for special applications", aiming to create a workstation using virtualization to separate sensitive information from different security domains. The project scope also includes access control issues (access control models, biometric and non-biometric identification) and audit support. Furthermore, his research aimes at developing new heuristic methods for threat detection. The study focuses on the possibilities of using lexical properties of domain names for detection of malicious WWW sites.

E-mail: krzysztof.lasota@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warszawa, Poland

Adam Kozakiewicz got his M.Sc. in Information Technology and Ph.D. in Telecommunications at the Faculty of Electronics and Information Technology of Warsaw University of Technology, Poland. Currently he works at NASK as Assistant Professor and Manager of the Network and Information Security Methods Team, also as part-time Assistant Professor at the Institute of Control and Computation Engineering at the Warsaw University of Technology. His main scientific interests include security of information systems, parallel computation, optimization methods and network traffic modeling and control.
E-mail: adam.kozakiewicz@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Paper

Cryptographic Protection of Removable Media with a USB Interface for Secure Workstation for Special Applications

Jan Chudzikiewicz and Janusz Furtak

Military University of Technology, Warsaw, Poland

Abstract—This paper describes one of the essential elements of Secure Workstation for Special Applications (SWSA) to cryptographic protection of removable storage devices with USB interface. SWSA is a system designed to process data classified to different security domains in which the multilevel security is used. The described method for protecting data on removable Flash RAM protects data against unauthorized access in systems processing the data, belonging to different security domains (with different classification levels) in which channel the flow of data must be strictly controlled. Only user authenticated by the SWSA can use the removable medium in the system, and the data stored on such media can be read only by an authorized user by the SWSA. This solution uses both symmetric and asymmetric encryption algorithms. The following procedures are presented: creating protected a file (encryption), generating signatures for the file and reading (decryption) the file. Selected elements of the protection systems implementation of removable Flash RAM and the mechanisms used in implementation the Windows have been described.

Keywords—filter driver, removable media protection, symmetric and asymmetric encryption.

1. Introduction

Nowadays, the most comfortable Removable large-capacity data devices are connected to the system via the bus Universal Serial Bus (USB). Such devices include flash memory RAM with a capacity of several tens of GB and hard disk drives with a capacity of several TB. The popularity of these devices forces the need for mechanisms to ensure an adequate level of protection of data stored on them. This is important in the case of sensitive data which have a significant impact on the safety of the institution. This fact is particularly important in the systems where confidential data are processed. It is hard to imagine a contemporary computer system in which the data storage devices cooperating with the system through the USB bus are not available. This observation also applies to Secure Workstation for Special Applications[1] (SWSA) [1], [2].

[1]SWSA is a computer system in which multi-level protection mechanisms have been implemented. In this system, the stored and processed data (objects) are classified due to the required level of security. Users of the system (subjects) have specific authorization to work with classified data. In order to ensure confidentiality and integrity of data for subjects are used mechanisms of mandatory access control to objects.

In ordinary systems to protect data on the media Flash RAM, the most commonly used software (e.g., USB Flash Security, Secure Traveler, Rohos Mini Drive, etc.) must be installed on the media prior to its use. During the installation of such software, in Flash RAM is created an encrypted volume which is accessed by using then defined password. The power of safeguard of the medium using this type of software depends on the used symmetric encryption algorithm and key length. This type of security is sufficient in the case of a loss or theft of the media. The use of such solutions in systems with multilevel security (MLS) which include the SWSA is insufficient for the following reasons:

– system does not provide the possibility of the control of an access to data copied to such a secured removable media, in particular, an entity with the lower clause cannot be blocked by system, while trying to read to the data with a higher classification contained on the medium;

– transfer of data between different entities that use such media possesses problems arising mainly from the need to provide the transfer of the medium and an encryption key with help which the medium has been encrypted;

– entity that creates a copy of the data on the media does not assure that the data are only available for the appropriate recipient and the recipient does not have an assurance that data is received from the expected sender.

The article presents a solution enabling to such a preparation of data stored in Flash RAM, so that the recording medium can be used to secure the transfer of data files, during which the sender of data (i.e., the creator of the protected media content) is assured that data will be available only for designated recipient, and the recipient is assured that the received data comes from the expected sender. The described mechanism uses both symmetric and asymmetric encryption algorithms and asymmetric. The presented solution uses a filter driver [3]–[5].

In this solution, it is assumed that in terms of operating system data can be processed in two directions: from plain

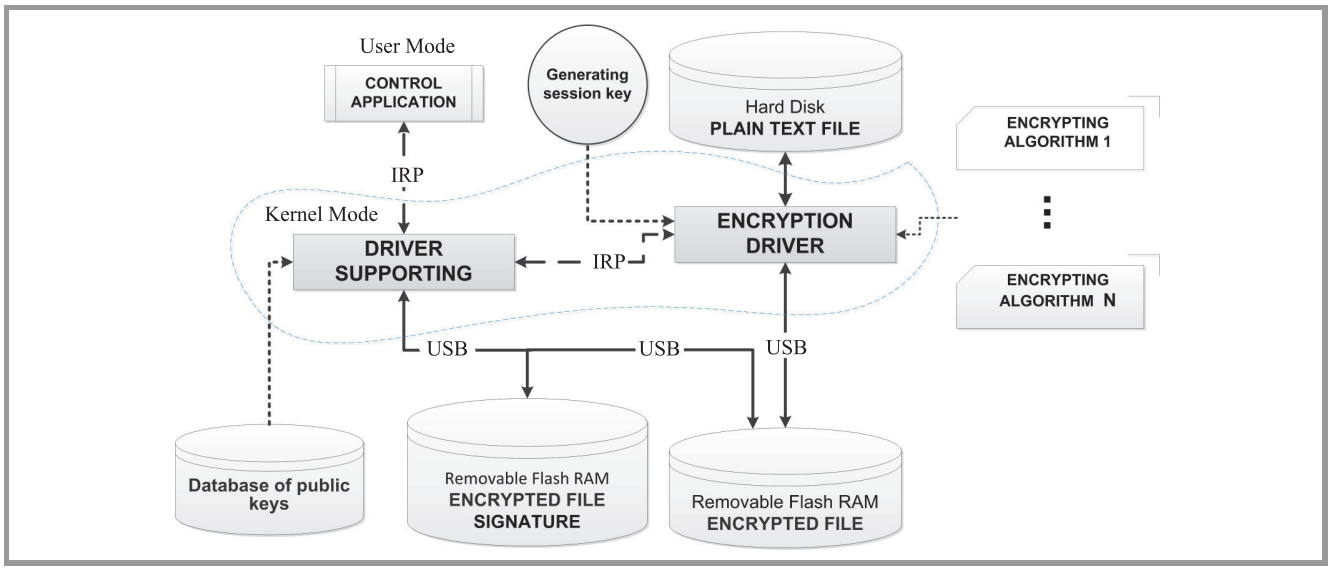


Fig. 1. Schematic diagram of securing data stored on removable media.

text form stored on your hard disk to secure form on removable media (e.g., Flash RAM, hard drive) connected to the system via the USB bus and, conversely, from secure form on removable media to plain text form on the hard drive. Do not allow the possibility of using the software for the direct exchange of data files between removable media (e.g., flash memory) connected to the system via the USB bus.

The process of securing the data exchange using removable media should satisfy the following functional requirements:

- it should be implemented in a manner transparent to the user,
- it should not cause any noticeable loads of the operating system to the user,
- it should not have a significant impact on the speed of read and write data onto data media,
- it should allow the use of various encryption algorithms to ensure the required level of confidentiality,
- it should be built on removable media, protected file and the signature for this file (signature should contain securely stored data necessary to encrypt/decrypt the protected file, and to ensure the integrity of the file,
- it should allow to perform any operation allowed for data storage media, such as volume, surface checking for errors, and defragment the disk-based data.

These requirements force the use of the process of securing the data separate modules (drivers), operating at the kernel-level of operating system [4]–[7]. Schematic representation of a solution being developed in the environment of Windows systems family is shown in Fig. 1.

Described solution is available for a user of the secure workstation through the control application (CApp). The main elements of the built system are interacting drivers: encryption driver [3] and driver supporting, which are compatible to the Windows Driver Model [6]. Both elements work in kernel mode, operating system and communicate with each other using the internal mechanisms of the operating system (in the figure they are labeled as IRP) [6], [8]. These mechanisms are described in details in Section 2.

The purpose of the encryption driver (EnD) is the realization of the process of encryption/decryption of data and determination of its hash value for these data. The driver supporting (DSu) sets the signature for protected data (according to the algorithm presented in Section 3) and mediates the transfer of messages/commands between EnD and CApp. The other components of the system are: the .DLL library that provides the functionality of the implemented encryption algorithms, module of generation of session key, and the database of public keys of users.

The product of the process securing the original file consists of two files: a file with encrypted data and the file containing the signature for the encrypted file. Both files can be stored on one medium or each file on a separate medium. Choosing a storage location of the signature file is defined by the user through CApp. It should be noted that saving the encrypted file and the signature file on separate media increases the security of stored data, but it is inconvenient to use.

2. Filter Drivers in Windows

Construction of Class Windows operating system assumes the use of two modes: user mode and kernel mode [6], [8]. Architecture of such a system is shown in Fig. 1. Modular design allows for an easy expansion of system functions (which is clearly visible in Fig. 2, showing the components operating in kernel mode), while the use of hardware ab-

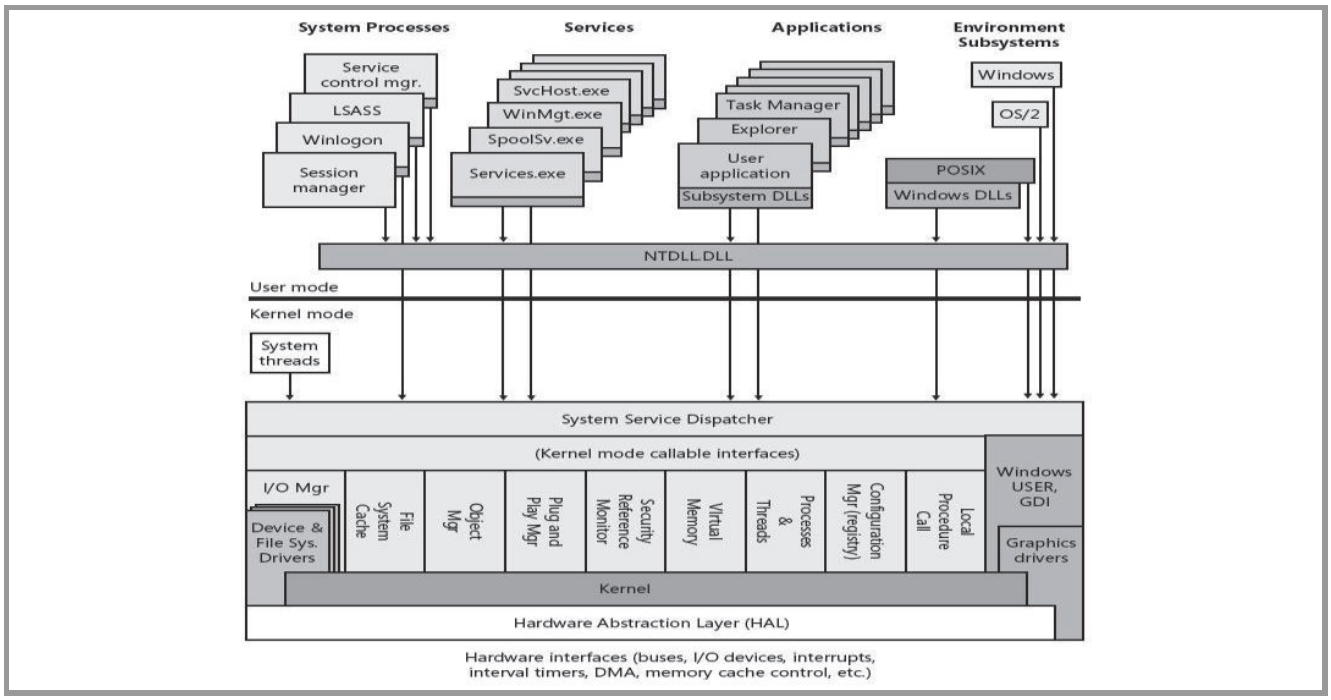


Fig. 2. Architecture of operating systems Windows [6].

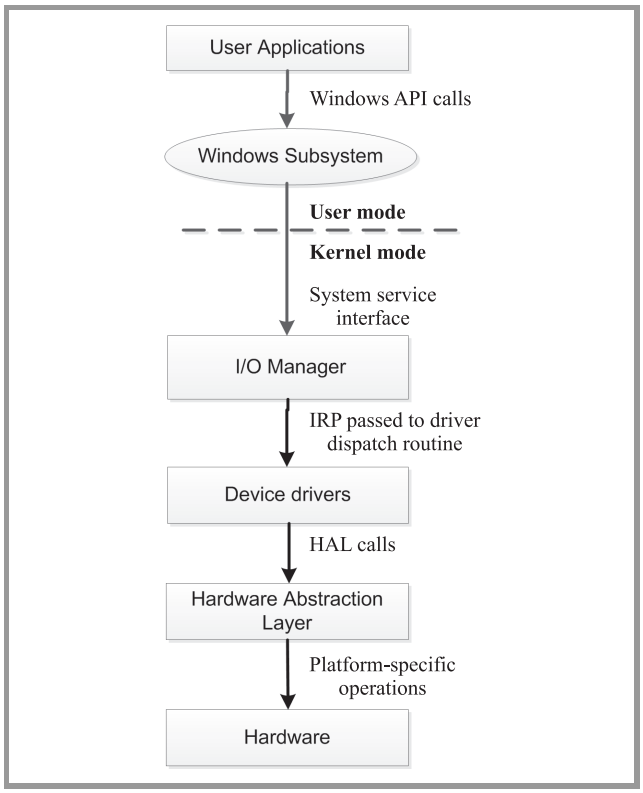


Fig. 3. Process of communication between user application and the driver [6].

straction layer (HAL) provides portability between different hardware architectures.

From the point of view the task of securing the content of removable storage media in the SWSA and the drivers of these devices (in the part relating to user mode) are relevant only the following components:

– user application (the CApp in the solution which is presented in this paper);

– Environment Subsystem (Windows);

– ntdll.dll system library that allows a communication with the elements working in kernel mode.

Environment subsystems form a working environment for applications running on them. It translates the application call to the system and its resources to the primary functions of Windows. The process of communication between user application and driver uses packets IRP that are created

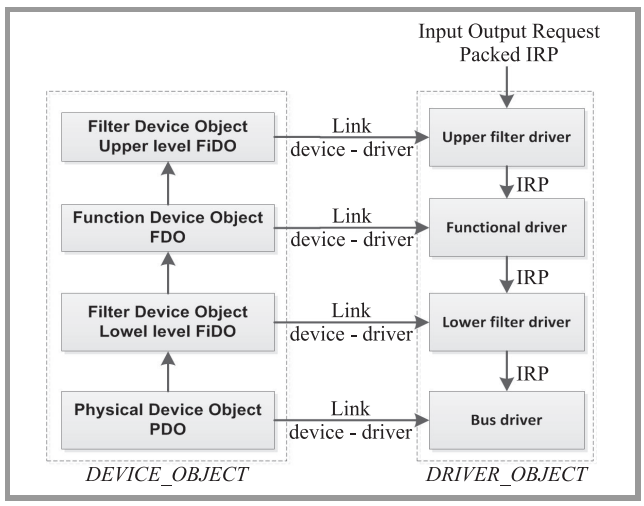


Fig. 4. Windows Driver Model [6].

by the I/O Manager on the basis of the generated request type. The process of handling requests from the user's application in which the IRP packet is generated is shown in Fig. 3.

The Windows Driver Model assumes that the devices are controlled by a stack of drivers working together, each of which is responsible for the implementation of other tasks of the device. The driver stack model is shown in Fig. 4. In this model, there are always two drivers: a bus driver at the bottom of the stack and the functional driver which defines the utility functions of the device. The model allows for the possibility of using additional filter drivers that are placed in the stack and allow you to monitor and modify the contents of packages I/O requests directed to device. DEVICE_OBJECT is a representative of a particular device, such as flash RAM, and is associated with a driver (DRIVER_OBJECT) that supports it.

Driver objects are created by the I/O Manager when the driver is loaded. Drivers are responsible for creation of

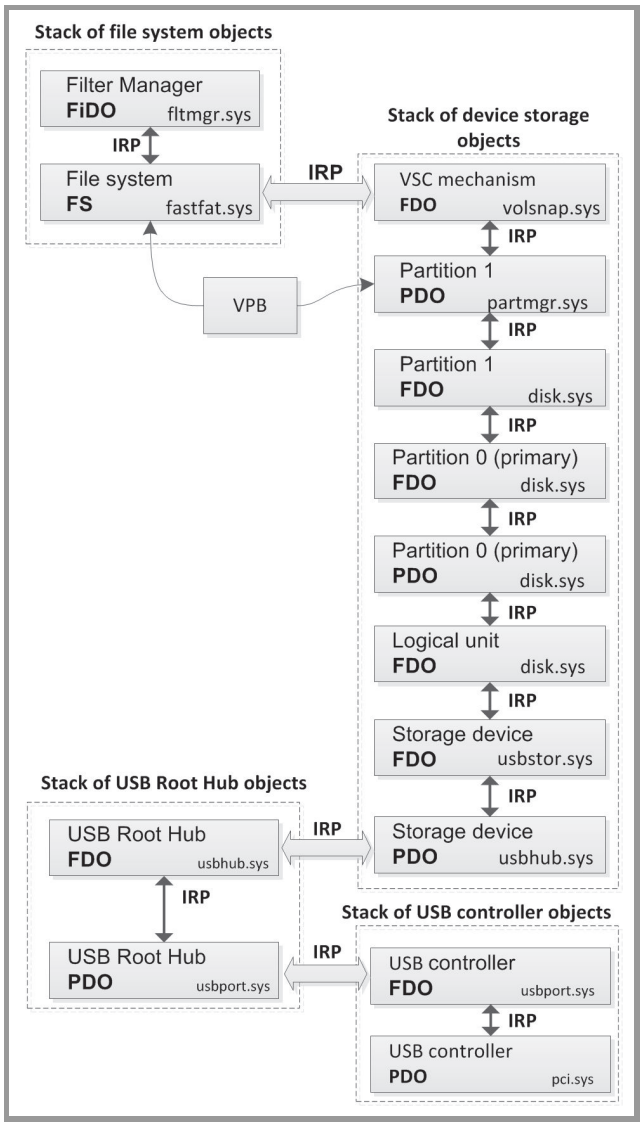


Fig. 5. Objects stack for the storage device connected to the system via the USB [3].

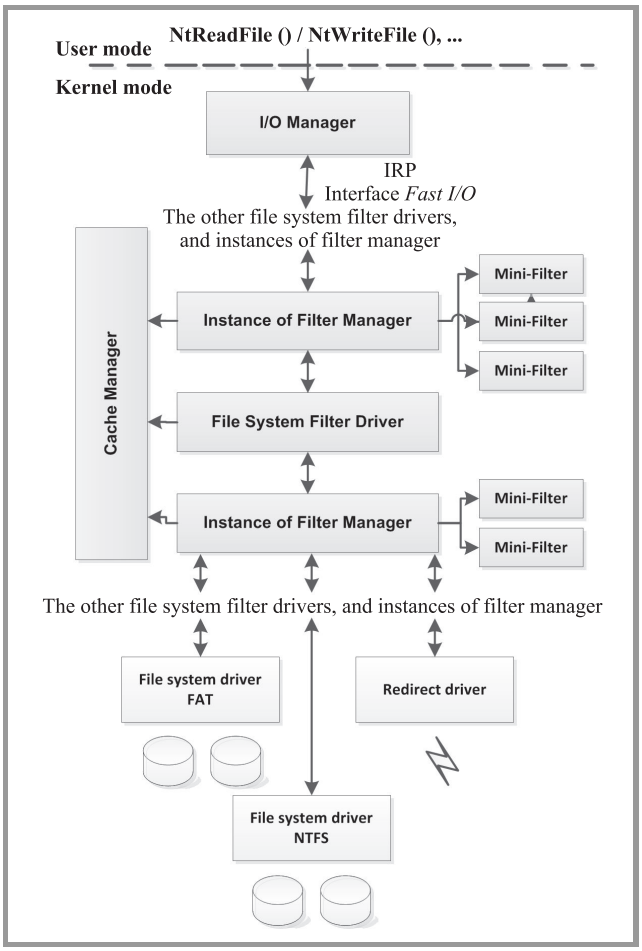


Fig. 6. The drivers stack for data storage devices using mechanisms of Filter Manager [4].

the DEVICE_OBJECT representing the system devices. Creating an object of this type occurs when the AddDevice procedure is called by the I/O Manager. Depending on the role of the driver, created object can represent:

- Physical Device Object (PDO) – representing the connection between the device and the bus;
- Functional Device Object (FDO) – functional driver uses it to determine functions of the device;
- Filter Device Object (FiDO) – filter driver uses it to process data from the packet I/O requests, and in particular their encryption, which is used in the described solution.

The I/O Manager can start sending packets I/O requests to the supported device after all DEVICE_OBJECT objects will be created.

Implementation of the described drivers model for the storage device connected to the system via the USB bus is more complex. Figure 5 shows objects stack associated with drivers of such a device. The presented diagram shows that on the top of the objects stack there is an object of filters manager. Below are the objects corresponding to

the FAT32 file system (marked on the diagram as an object of type FS). An attention should be paid to the file system drivers that are not explicitly included in the driver stack. The association between driver of file system FS and object representing a partition in the storage device is executed by a data structure Volume Parameters Block (VPB).

The presented solution uses a kernel component called the Filter Manager. Filter Manager performs a significant part of the tasks. Otherwise it would have to be performed by the filter drivers. As a result, using the filter manager (called mini-filters) is simpler and easier to implement. Filter Manager is available in all systems ranging from Windows Server 2003 to Windows XP with Service Pack 2. Figure 6 shows the drivers stack for data storage devices using mechanisms of Filter Manager.

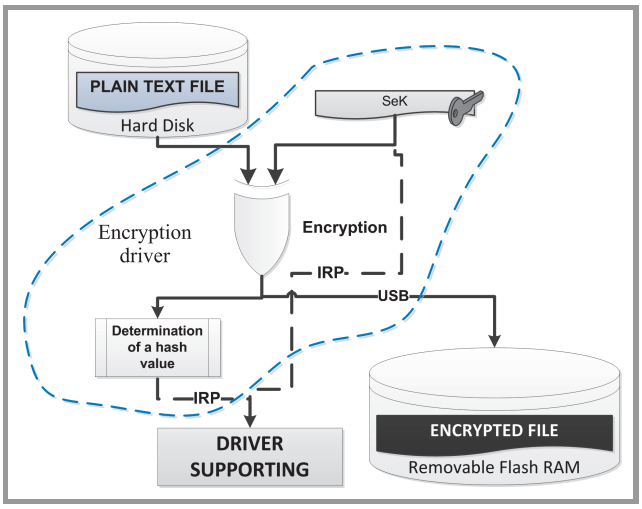


Fig. 7. The process of writing data to removable flash memory.

3. The Process of Creating and Reading a Protected File

In the process of creating a protected file on removable flash memory (that is a creating an encrypted file and the signature of this file), and reading (decrypting) the file from the removable flash memory are the necessary attributes of the user who created the protected file (this user will be called the sender), and user for whom the protected file was created (this user will be called the recipient). When creating a protected file, sender is role plays a user logged into the system and he specifies a file recipient using CApp (you can select one from among the users who meet the requirements of SWSA closely related to the multilevel security of system). When reading a protected file with the use of CApp, the logged user plays the recipient role, and the sender attributes are read from signature file after the successful decryption of this file, using the private key of the logged on user. Permissible is a situation in which the logged user is simultaneously the sender and the recipient of data.

The process of creating a protected file includes the step of encryption, and then creating a signature for that file. However, during the process of reading a protected file in a first step, the attributes needed for decrypting this file are obtained from the signature. In the second step, the file is decrypted.

3.1. Creating a Protected File

The process of writing the file, including file encryption and hash generation is performed by the EnD. Operation of EnD has been presented in [3]. The diagram describing the process of writing the file is shown in Fig. 7. Dashed line in that figure indicates operations implemented by the EnD. During the process of file encryption, the value of the hash function is determined to ensure the integrity of the file.

The determined value of the hash function and the generated session key after completion of record are transferred to the DSu in order to generate a signature for the stored data. The process transferring of the hash function value and the session key transferring is implemented using the system mechanisms marked in Fig. 7, as the IRP.

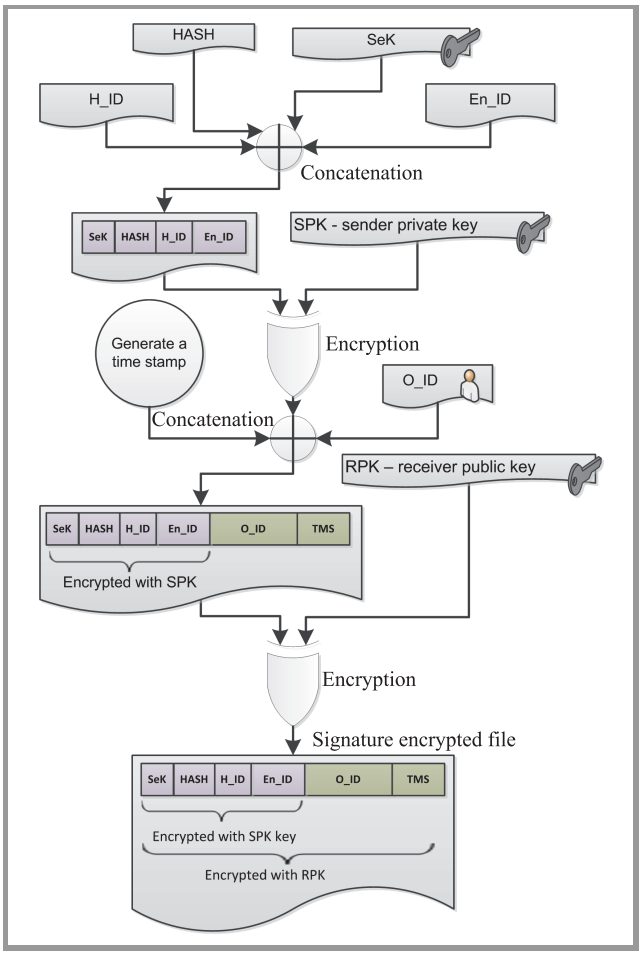


Fig. 8. The algorithm of signature generation for protected file.

3.2. Determining the Signature

For each of the protected file the signature is generated which contains the information needed to read this file. Signature of the file contains the following fields:

- SeK – random key to encrypt/decrypt the secure file,
- HASH – value of hash function which is determined on the basis on the content of protected file after encrypting this file,
- H_ID – identifier of the algorithm used to generate the hash,
- En_ID – identifier of the algorithm used to encrypting,
- O_ID – identifier of the logged user (the sender) who initiated the operation of data write – this identifier is required to determine the public key of sender when the file is read,
- TMS – time stamp of file creation – this value corresponds to the date of file creation.

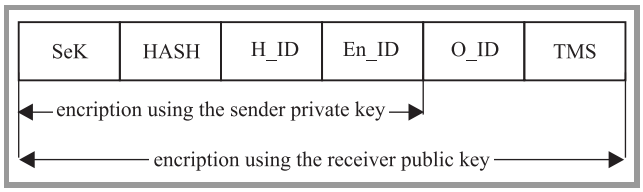


Fig. 9. The structure of signature secure file.

The process of signature creation proceeds according to the diagram shown in Fig. 8, and the structure of signature is shown in Fig. 9.

3.3. Reading a Protected File

The process of reading of the file requires that the signature to be read before and then decrypted. These activities are performed by the logged user (recipient of file) using CApp. The process starts with decrypting the signature file using the private key of the logged user, then reading time stamp and user identifier (O_ID) which assumed the role the sender creating a protected file. The time stamp protects the encrypted file before moving it to another medium that it was originally written on. Incompatibility of date and time stored in the time stamp and date and time, when the file was created, causes displaying the message and terminating the procedure of file reading. Along compatibility of the parameters, the next part of the signature is decrypting using the user public key of which identifier (O_ID) has been read. The next steps of file decoding are schematically shown in Fig. 10.

In Fig. 10, the operations performed by the EnD are marked using thick dashed line, and the operations performed by the DSu are marked using the thinner line (two dots dash).

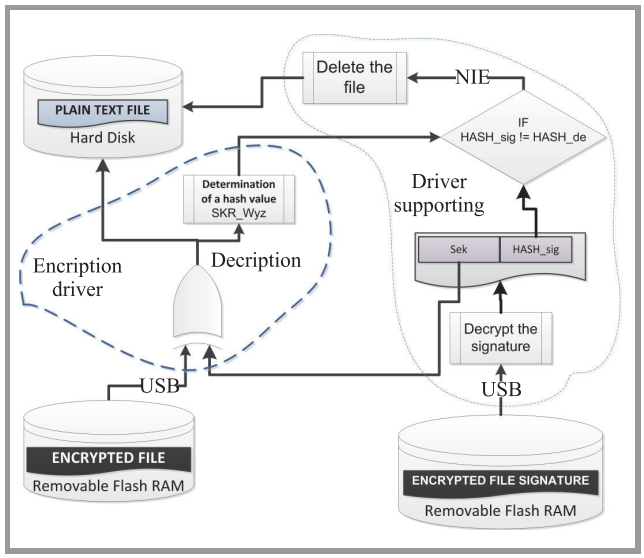


Fig. 10. The process of data reading from an external file.

During the data reading, the value of hash function (HASH_de) is determined. If the value HASH_de is different from the values obtained from the signature (HASH_sig), a message is displayed and the decrypted file, which was saved on hard disk, is being deleted.

4. Implementation Mechanisms of Encryption/Decryption

This section presents the implementation of selected elements of drivers EnD and DSu. In particular, it describes the pieces of code relating to the registration of the EnD driver in the driver stack of the file system, and initiate a logical device by the DSu driver and its associated data structure that stores the signature components of the protected file.

4.1. Implementation the Selected Elements of Encryption driver (EnD)

The driver registration process in the stack of file system drivers is realized when loading it into the system by the I/O Manager. The I/O Manager calls the *DriverEntry* driver function, which will be the entry point to the driver [6]. In this function, as the first step shown in Fig. 11, *Lookaside* list of type is initialized, in which there are held objects representing the context of substitution data buffers.

In the next step, the functions made available by the filter manager are called [4], [6]. First, the driver is registered in the system by these functions and then filtering process is launched (Fig. 12). After registering a mini-filter using *FltRegisterFilter* function, the filter manager takes over the management of the filter.

```
. . .
   ExInitializeNPagedLookasideList(&BufferSwapCo
      ntextList, NULL, NULL, 0,
      sizeof(BUFFER SWAP CONTEXT),
   BUFFER_SWAP_CONTEXT_TAG, 0);
. . .
```

Fig. 11. Initializing a *Lookaside* list.

```
. . .
   Status = FltRegisterFilter(IN DriverObject, IN
      &FilterRegistration,
      OUT &KMSECData.FilterHandle);
   Status = FltStartFiltering(IN
      KMSECData.FilterHandle);
. . .
```

Fig. 12. The filter registering and activating filtering process.

From this moment, a mini-filter captures all messages sent to the functional driver of file system. One of these messages may be a message about trying to mount a new volume. Then *InstanceSetup* function is called. This function checks whether there is a mounted volume on a storage device connected via USB and includes the correct file system type. If these conditions are met, the volume is connected and its context[2] is initialized. Otherwise, to the filter manager is returned the STATUS_FLT_DO_NOT_ATTACH status code, which block the connection of the mini-filter to the volume. A piece of code referring to the previous steps is shown in Fig. 13.

```
. . .
   if ((Flags !=
      FLTFL_INSTANCE_SETUP_MANUAL_ATTACHMENT) &&
          ((VolumeDeviceType ==
      FILE_DEVICE_DISK_FILE_SYSTEM) &&
        (VolumeFilesystemType ==
FLT_FSTYPE_FAT)))
   {
        if (KMSECIsUSBDevice(FltObjects-
        >Volume))
        {
            . . .
        }
   }
   else
   {
        Status = STATUS_FLT_DO_NOT_ATTACH;
   }
. . .
```

Fig. 13. Verification of volume data.

Initiating the volume context requires allocating the necessary quantity of non-paged memory, which is done by calling the *FltAllocateContext* function. After the correct memory allocation, this function returns STATUS_SUCCESS. One should pay attention to the need to verify this value.

[2]The volume context is defined by the volume status, which is described in a data structure containing the properties of a volume. In this structure the basic data about a mounted mass storage device are described.

A piece of code executing this operation is shown in Fig. 14. If one omits this check, in case of failure of memory allocation can cause an unstable system.

```
. . .
   Status = FltAllocateContext(FltObjects->
      Filter, FLT_VOLUME_CONTEXT,
      sizeof(VOLUME CONTEXT), NonPagedPool,
      &pVolumeContext);
   if (!NT SUCCESS(Status))
   {
      leave;
   }
. . .
```

Fig. 14. Appointment of the volume context.

```
. . .
   Status = FltGetVolumeProperties(FltObjects->
      Volume, &VolumeProperties,
      sizeof(VolumeProperties),
      &LengthReturned);
   . . .
   pVolumeContext->SectorSize =
      max(MIN_SECTOR_SIZE,
      VolumeProperties.SectorSize);
. . .
```

Fig. 15. Setting the volume sector size.

For correct implementation of the operation encryption/decryption the data included in the volume context is necessary. These data can be downloaded each time during encryption/decryption from the volume context, but the storage of this data in the driver structure accelerates the process of encryption/decryption. Therefore, the recommended solution is a single download of this data by using function *FltGetVolumeProperties*, as it is shown in Fig. 15.

4.2. Implementation the Selected Elements of Supporting Driver (DSu)

In the structure of DSu driver are used drivers marked as LEGACY DRIVER that runs in kernel mode. This solution has been accepted because the driver doesn't need to create the device objects dynamically. This driver, due to the need for communication with other components of the system, creates one logical device object which is also used to store base elements of signature. At the moment of loading, the driver in the system is being called by I/O Manager the *DriverEntry* that has a driver function which in the first step initiates an *MajorFunction* array. These functions will be handled by I/O's request addressed to the driver. A piece of code which implements an operation of MajorFunction array initialization is shown in Fig. 16.
Then in the *DriverEntry* function is being called *IoCreateDevice* function which creates a device object in the non-paged memory area. Device object is not related to any physical device occurring in system, but is used only

```
. . .
for (i = 0; i <= IRP_MJ_MAXIMUM_FUNCTION; i++)
   DriverObject->MajorFunction[i] =
        USB_WSP_LEGDefaultHandler;
   DriverObject->MajorFunction[IRP_MJ_CREATE] =
        USB WSP LEGCreateClose;
   DriverObject->MajorFunction[IRP_MJ_CLOSE] =
        USB WSP LEGCreateClose;
   DriverObject->DriverUnload =
        USB_WSP_LEGUnload;
   DriverObject->MajorFunction[IRP_MJ_WRITE] =
        USB_WSP_LEGWrite;
   DriverObject->MajorFunction[IRP_MJ_READ] =
        USB_WSP_LEGRead;
   DriverObject->
        MajorFunction[IRP MJ DEVICE CONTROL] =
        USB_WSP_LEGIoControl;
   DriverObject->MajorFunction[IRP_MJ_SHUTDOWN] =
        USB_WSP_LEGShutdown;
. . .
```

Fig. 16. Initializing the *MajorFunction* array.

```
. . .
   RtlInitUnicodeString(&DeviceName,L"\\Device\\U
   SB_WSP_LEG");
   RtlInitUnicodeString(&Win32Device,L"\\DosDevic
   es\\USB_WSP_LEG0");
   status = IoCreateDevice( pDriverObject,
               sizeof(DEVICE_EXTENSION),
               &DeviceName, FILE_DEVICE_UNKNOWN,
               0, TRUE, &pDevObj );
   if (!NT_SUCCESS(status))
          return status;
   if (!pDevObj)
          return STATUS_UNEXPECTED_IO_ERROR;

. . .
```

Fig. 17. Initializing the device object.

as an element that stores necessary data to generate the signature. Figure 17 shows a piece of code responsible for creating the device object.

```
. . .
typedef struct _DEVICE_EXTENSION
{
 PDEVICE_OBJECT pDevice;// pointer to the device object linked
                        // with the that structure

 UNICODE_STRING ustrSymLinkName;// the internal name of
                                //the device
 PVOID Hash; // pointer to the hash value for the encrypted file
 PVIOD SessionKey; // pointer to the session key
 PVIOD IDCryptAlg; // pointer to the identifier of the encryption
                   //algorithm
 PVIOD IDHashAlg; // pointer to the ID of hash generation
                  //algorithm
 PVOID IDOperator; //pointer to the current user/ID
 PVIOD IDReciver; // pointer to the recipient ID
 PVIOD TimeStamp; // pointer to a time stamp
 PVOID deviceBuffer; // pointer to the final signature
} DEVICE_EXTENSION, *PDEVICE_EXTENSION;
. . .
```

Fig. 18. Definition of the extension structure.

IoCreateDevice function call is preceded by a definition of the internal name (which is functioning on the kernel mode of system) and the symbolic name used as the references to the driver from user mode. Verification of correctness of the device object creation is realized by checking whether the value of status returned by the *IoCreateDevice* function is equal to the value of pointer to the device object – *pDevObj*. Validation of creating a device object is achieved by checking whether the status value returned by the function *IoCreateDevice* is consistent with the value of a pointer to the device object – *pDevObj*. If the returned value of status will be different from STATUS_SUCCESS the function terminate working and returns the error code of input/output to the parent function. After successful creation of the device object, the function returns the handle to it, which will be then used to realize the later references to this device. Then the initialization process of the extension structure is realized which definition is shown in Fig. 18.

A communication between the encryption driver (EnD) and the driver supporting (DSu) is always initiated by the EnD driver. The type of request directed to the driver supporting depends on the direction of copying data, which is initiated by the currently logged user on the system.

4.3. Handling for Creating and Reading a Secure File

Logged user (file sender) configures the parameters of the process of creating and reading a protected file using CApp which window is shown in Figs. 19 and 20, respectively.

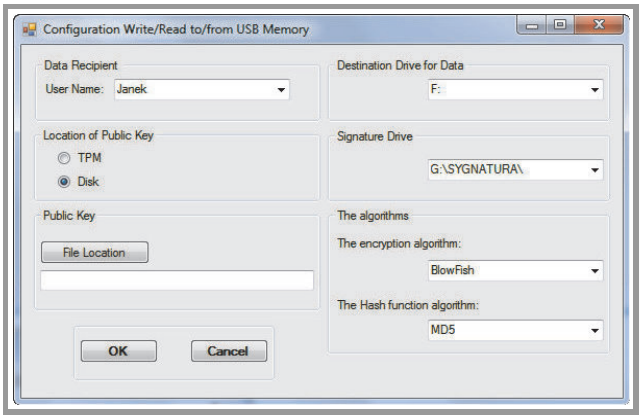


Fig. 19. The window of control application CApp for write data.

First of all, the process of creating protected file requires connection one or two (depending on where the file with the signature will be stored) removable Flash RAM memories to a computer through USB interface. The devices are automatically detected by EnD which transmits information about them via the DSu to CApp.

The logged user should determine the parameters required for encrypting the file and generating a signature. He does this by selecting (see the Fig. 19):

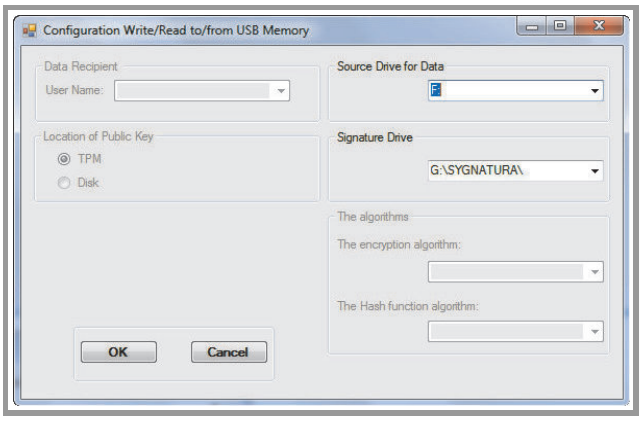


Fig. 20. The window of control application CApp for read data.

– drive in which the protected file will be stored ("Data Drive" field),

– drive and path to the directory in which the file with the signature will be stored ("Signature Drive" field),

– identifier for the algorithm used to encrypt ("The encryption algorithm" field),

– identifier for the algorithm used to generate hash value for the protected file ("The Hash function algorithm" field),

– identifier for user (receiver) encrypted data ("Data Recipient" field),

– location of public key data recipient file ("Location of Public Key" field).

Identifier (O_ID) and the private key of the sender (the elements required to generate the signature) are automatically retrieved from the system. After determining the data configuration, logged user can begin the process of copying the file using, e.g., Windows Explorer. The name of a file which stores the signature will be concatenation of the name of protected file and string "SIG". The process of creating a file with the signature is started after the encryption process is finished and, just as the encryption process, it is invisible to the user. When the next file for the same recipient is being encrypted it does not need to change the configuration data unless the other parameters (that is the identifier of encryption algorithm or identifier of algorithm generating of hash value) will be changed. Always for the next file, a new session key will be automatically generated.

The process of reading protected file requires a connection to a computer through USB interface one or two (depending on where is stored the file with the signature) removable Flash RAM memories. The devices are automatically detected by EnD which transmit information about them via the DSu to CApp. The logged user (recipient of the data) has to specify the drive using CApp on which encrypted file is stored and indicate the file with the signature corresponding to the encrypted file. He accomplishes this by selecting (see the Fig. 20):

– drive on which is stored the protected file ("Data Drive" field),

– drive and path to the directory on which is stored the file with the signature ("Signature Drive" field).

Other parameters required to decrypt the file are determined based on the signature. After initializing by the logged user, the process of copying a file EnD sends to the DSu the name of the copied file and pauses the copy process to the moment when are receives data required to decrypt the file (that is the identifier of encryption algorithm, session key and identifier of algorithm generating of hash value). Based on submitted by the EnD the name of encrypted file, DSu identifies a file containing the signature and performs the process of signature decryption and reading the configuration data. Then performs the verification process read out TMS with the date and time of the creation of an encrypted file. In the case of inequality of these values, message is displayed and the file reading process is interrupted. In the case of equality of those values, other configuration data read from the signature are passed to EnD, which resumes the process of decryption. During the process of decrypting the file, the EnD determines the value of a hash function for that file. After completion of the copying process, EnD transmit to DSu determined value of the hash function for verification. If the designated hash value is not equal to the value read from the signature, a message is displayed and the DSu deletes the file.

5. Conclusion

The SWSA uses proprietary solution for securing data on removable Flash RAM. There have not been applied widely available tools to secure the contents of this type of removable storage media due to the fact that these solutions typically uses only symmetric encryption when writing files. In these solutions, the key needed for encryption/decryption is specified by the user who creates a protected file and it is assumed that this key is known to the user when a protected file is read. The problems associated with the transmission of the key between the users are not taken into account. Such a solution in the SWSA was not useful.

The solution presented in this paper is an unique and more complicated one. The problem with the transmission of the key does not apply users of SWSA, because they are using the advantages of asymmetric encryption which gives assurance secured transfer of encryption key between parties, involved in the exchange of data. In addition, the SWSA uses Trusted Platform Module which supports the creation and management of cryptographic keys.

The developed system requires the user who creates a protected file, to specify only the recipient's file and the file encryption parameters. The recipient of the file can only

be the user authorized by the SWSA. The process of protecting file is closely linked with the mechanisms of systemic support for removable media Flash RAM, and is transparent to the user. When the protected file is read, user is not burdened with any additional activities. In addition, protections are constructed in such a way that the reading of a file is possible only by the authorized user by the SWSA, and only with the medium on which the file was originally saved. An attempt to copy the protected file to a different medium locks the ability to read the file.

The described method for protecting data on removable Flash RAM protects data against unauthorized access in systems processing the data, belonging to different security domains (with different classification levels) in which channel the flow of data must be strictly controlled. The described solution protects data stored on removable Flash RAM in case of loss or theft of the medium, but also makes it possible to secure transfer of that data through an unsecured transmission channel, for example using a courier.

Acknowledgements

This work was supported by The National Center for Research and Development, Project OR00014011.

References

[1] A. Kozakiewicz, A. Felkner, J. Furtak, Z. Zieliński, M. Brudka, and M. Małowidzki, "Secure workstation for special applications", in *Secure and Trust Computing, Data Management, and Applications*, C. Lee, J.-M. Seigneur, J. J. Park, and R. R. Wagner, Eds., Communications in Computer and Information Science, vol. 187. Berlin: Springer, 2011, pp. 174–181.

[2] Z. Zieliński *et al.*, "Secured workstation to process the data of different classification levels", *J. Telecom. Inform. Technol.*, no. 3, pp. 5–12, 2012.

[3] J. Chudzikiewicz, "Zabezpieczenie danych przechowywanych na dyskach zewnętrznych", in *Metody wytwarzania i zastosowania systemów czasu rzeczywistego*, L. Trybus and S. Samolej, Eds. Warszawa: Wydawnictwo Komunikacji i Łączności, 2010, pp. 211–221 (in Polish).

[4] R. Nagar, "Filter Manager", Microsoft Corporation, Redmond, 2003.

[5] R. Nagar, *OSR's Classic Reprints: Windows NT File System Internals*. Redmond: OSR Press, 2006.

[6] Technical Documentation "Microsoft Windows Driver Kit (WDK)", Microsoft Corporation, Redmond, 2009.

[7] M. E. Russinovich and D. A. Solomon, *Microsoft® Windows® Internals, Fourth Edition: Microsoft Windows Server™ 2003, Windows XP, and Windows 2000*. Redmond: Microsoft Press, 2005.

[8] W. Oney, *Programming the Microsoft® Windows® Driver Model*. Redmond: Microsoft Press, 2003.

Jan Chudzikiewicz – for biography, see this issue, p. 11.

Janusz Furtak – for biography, see this issue, p. 12.

Paper

A Hybrid CPU/GPU Cluster for Encryption and Decryption of Large Amounts of Data

Ewa Niewiadomska-Szynkiewicz[a,b], Michał Marks[a,b], Jarosław Jantura[b], and Mikołaj Podbielski[b]

[a] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[b] *Research and Academic Computer Network (NASK), Warsaw, Poland*

Abstract—The main advantage of a distributed computing system over standalone computer is an ability to share the workload between cores, processors and computers. In our paper we present a hybrid cluster system – a novel computing architecture with multi-core CPUs working together with many-core GPUs. It integrates two types of CPU, i.e., Intel and AMD processor with advanced graphics processing units, adequately, Nvidia Tesla and AMD FirePro (formerly ATI). Our CPU/GPU cluster is dedicated to perform massive parallel computations which is a common approach in cryptanalysis and cryptography. The efficiency of parallel implementations of selected data encryption and decryption algorithms are presented to illustrate the performance of our system.

Keywords—AES, computer clusters, cryptography, DES, GPU computing, parallel calculation, software systems.

1. Introduction

Data encryption and decryption are generally complex problems and involve cumbersome calculations, especially when consider processing of large amounts of data. The restrictions are caused by demands on computer resources, i.e., processor and memory. However, in many cases the calculations performed by cryptography algorithms can be easily partitioned into large number of independent parts and carried out on different cores, processors or computers. It was observed that parallel implementation based on MapReduce programming model improves the efficiency of the algorithm and speeds up a calculation process.

CPU and GPU clusters are one of the most progressive branches in a field of parallel computing and data processing nowadays, [1], [2]. A cluster is a set of computers working together so that in many aspects they can be viewed as a single system. Typical cluster consists of homogenous Central Processing Units (CPUs). A new model for parallel computing based on using CPUs and GPUs (Graphics Processing Units) together to perform a general purpose scientific and engineering computing was developed in the last years, and used to solve complex scientific and engineering problems. Using CUDA or OpenCL programming toolkits many real-world applications can be easily implemented and run significantly faster than on multi-processor or multi-core systems [3].

In this paper we describe and evaluate a hybrid cluster system HGCC that integrates two types of multi-core CPUs, i.e., Intel and AMD processors equipped, adequately, with NVIDIA and AMD graphical units. We have designed and developed a dedicated software framework for parallel execution of computing tasks which aim is to hide a heterogeneity of the cluster – from the user's perspective, the cluster system serves as one server. The objective of this software is to divide the data into separate domains, allocate the calculation processes to cluster nodes, manage calculations and communication.

The remainder of this paper is organized as follows. We present a brief survey of modern GPU clusters in Section 2. The architecture of our cluster and software framework that manages calculations are described in Section 3. Finally, in Section 4 we briefly summarize results of tests for selected types of data encryption and decryption algorithms. The paper concludes in Section 5.

2. Survey of CPU and GPU Clusters

Every year in June and November the TOP500 list is published. The announcement of the list is not only the chance to find out what are the most powerful supercomputers but also a great opportunity to observe new trends in the HPC technologies. In June 2012, as compared to November 2011 list, when there was no turnover in the Top10, this time around, there are six brand new machines, plus one, Jaguar, that has benefitted from an upgrade to faster processors. The majority of these new machines is built using latest IBM solution called Blue Gene Q. Only one of new supercomputers is equipped with GPU. However it doesn't mean that the interest in applying GPU technology in supercomputers is falling down. Many of the most powerful supercomputer centers are waiting for new accelerators from NVIDIA, AMD and Intel. For example a new supercomputer Titan, which will be a successor of Jaguar and is currently being built in Oak Ridge National Laboratory, will be equipped with almost 15 000 of NVIDIA cards from "Keppler" family.

When we look at the whole Top500 list we can observe a rising significance of GPU accelerators. In June 2012 ranking, there are 58 machines that are equipped with GPU

accelerators, up from 17 only one year ago – see Fig. 1. It is worth noting that 53 of them use NVIDIA Tesla GPU co-processors while only two of them are equipped with IBM's Cell coprocessors and other two with AMD's Radeon cards. Moreover, a new product of Intel utilizing Intel MIC accelerator had its debut on TOP500 in an experimental cluster with pre-production Knights Corner chips. MIC chip will be available in the end of 2012, as Intel Xeon Phi coprocessor.

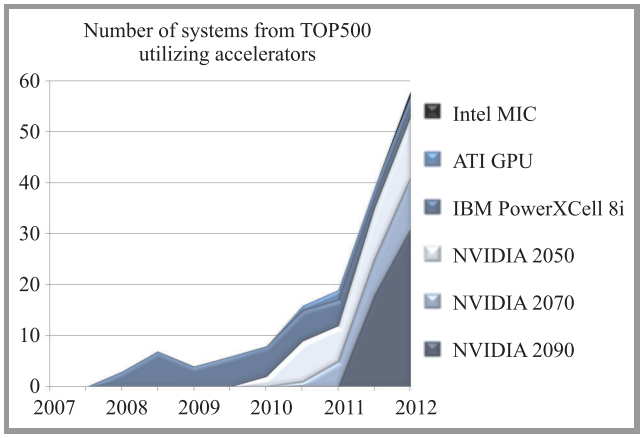


Fig. 1. Number of systems from TOP500 utilizing accelerators.

The most common operating systems used for building clusters are UNIX and Linux. Clusters should provide following features: scalability, transparency, reconfigurability, availability, reliability and high performance. There are many software tools for supporting cluster computing. In the beginning of XXI century, the common idea was to provide a view of one supercomputer for a cluster built from a group of independent workstations. The SSI (Single System Image) clusters were designed and developed. In this approach all servers' resources such as disks, memory, processors are seen by a user as one unique machine. The whole cluster is identified from outside by one IP address. The popular systems that implement the idea of SSI are Mosix (http://www.mosix.org) that does not cover all SSI features, and two comprehensive clustering solutions offering full SSI environments: OpenSSI (http://openssi.org) and Kerrighed (http://www.kerrighed.org). A brief overview and comparative study of stability, performance and efficiency of Mosix, OpenSSI and Kerrighed systems is presented in literature [4].

Other commonly used systems that can be applied to high performance data processing and calculations in cluster systems are software frameworks that perform job scheduling. Commonly used Portable Batch System PBS (www.pbsworks.com) provides mechanisms for allocating computational tasks to available computing resources. Various versions of the system are available, open source and commercial: OpenPBS, MOAB with Torque, PBS Professional.

Most of the presented cluster systems are mature solutions. However, they have some limitations. Mosix, OpenSSI and Kerrighed systems focuss on load balancing. The idea is to implement an efficient load balancing algorithm which is triggered when loads of nodes are not balanced or local resources are limited. In general, processes are moved from higher to less loaded nodes. Unfortunately, migration of processes involves extra time for load calculation and overhead in communication. Moreover, Mosix, OpenSSI, Kerrighed systems were designed for CPU clusters.

Currently, users are provided with software environments that allow to perform calculations on a single GPU device. There are only a few software tools for running applications on GPU clusters. Virtual OpenCL VCL (www.mosix.org/txt_vcl.html) is a software platform for GPU clusters. It can run unmodified OpenCL applications on Linux clusters with or without the Mosix system. VCL provides a view of one superserver for cluster built from a group of GPU units. The components of VCL, its performance and applications are presented in [5].

Our goal was to develop a software framework that allows unmodified OpenCL applications to transparently and concurrently run on multiple CPU and GPU devices in a cluster. In case of our application we need a simple functionality, i.e., a calculation speed up, resistance and ease of use. We perform static decomposition of the problem in calculation startup, hence the dynamic load balancing is superfluous. Our software framework is quite similar to VCL platform [5], however, in our solution it is possible to utilize both CPUs and GPUs on computational nodes.

3. HGCC System Overview

3.1. Hardware Components of HGCC

The aim of our work on utilizing a cluster composed of CPUs and GPUs in cryptography and complex data analysis is providing the system which functionality allow us to perform: effective computing of applications implementing MapReduce programming model, comparison of performance of CPUs and GPUs from many vendors along with comparison of different interconnects performance.

We have built a heterogenous cluster system with multi-core CPUs working together with many-core GPUs. The system consists of 24 nodes and integrates two types of CPUs: 12 servers with two Intel Xeon processors each and 12 servers with two AMD Opteron processors each. All servers are equipped with advanced GPUs, adequately, NVIDIA Tesla and AMD FirePro units. It is worth to note that FirePro V7800 has a peak performance in the single precision almost two times better than NVIDIA Tesla M2050, and a peak performance in the double precision equal 0.8 of Tesla's performance. Moreover, AMD GPU is approximately four times cheaper than NVIDIA GPU. However, effective programming of GPU based on Very Long Instruction Word (VLIW5) architecture – AMD FirePro V7800 – is not a simple task and not every application is capable to achieve performance close to the peak one.

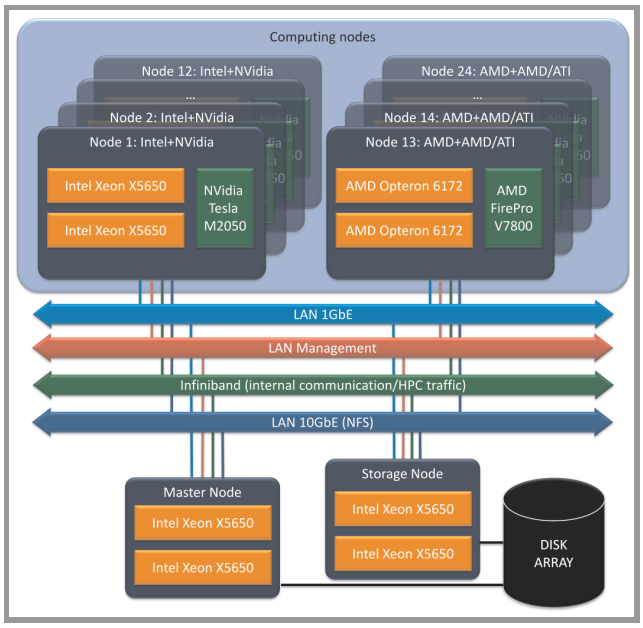


Fig. 2. Hybrid system architecture with Intel+NVIDIA and AMD+ATI/AMD nodes.

The system architecture is depicted in Fig. 2. The specification of components that form the HGCC cluster is as follows:

CPU Intel : Intel Xeon X5650, 2.66 GHz/3.06 GHz turbo, 6 cores / 12 threads, 6x256 L2, 12 MB L3 cache.

CPU AMD : AMD Opteron 6172, 2.1 GHz, 12 cores / 12 threads, 12x512 KB L2, 12MB L3 cache.

GPU NVIDIA : NVIDIA Tesla M2050, 448 CUDA cores, 384-bit memory bus.

GPU AMD : AMD FirePro V7800, 1440 stream processors (equivalent of 288 CUDA cores), 256-bit memory bus.

The computing nodes are supported by a dedicated master and storage nodes providing access to disk arrays and management capabilities. Communication between nodes is organized using different interconnects: InfiniBand 4x QDR, 10GbE and 1GbE. Such redundant network configuration allows us to verify the impact of selected interconnects on computation efficiency. Moreover, it is possible to separate communication connected with IO operations from computational traffic. The current configuration assumes utilizing InfiniBand network for providing access to data storage. 10GbE Ethernet and the 1GbE Ethernet are used for computational purposes.

3.2. HGCC Software Framework

The HGCC software framework provides an environment for parallel calculations that are performed on a cluster formed by heterogenous CPU and GPU devices. The goal was to hide a heterogeneity of the cluster and minimize the user's effort during the design, implementation and execution of the application. From a user's perspective, the cluster system should serve as one server. So, it allows a user to focus only on the numerical part of his application. The concept was to allow applications developed by users to transparently utilize many CPU and GPU devices, as if all the devices were on the local computer. A single system image model is implemented – all servers' resources such as CPU, GPU or memory are seen by the user as one unique machine. Therefore, applications written for HGCC benefit from the reduced programming complexity of a single computer, the availability of shared memory and multi threads, as in OpenMP (http://openmp.org/wp), and a concurrent access to cluster nodes and their devices, as in MPI (http://mpi-forum.org).

In order to take advantage of GPU accelerators from different vendors, we decided to use OpenCL, which is a low level GPU programming toolkit, where developers write GPU kernels entirely by themselves with no automatic code generation [6]. OpenCL is an industry standard computing library developed in 2009 that targets not only GPUs, but also CPUs and potentially other types of accelerator hardware. In OpenCL, an efficient implementation requires preparation slightly different codes for different devices, however, it is much less complicated than writing code in many native toolkits for NVIDIA and AMD devices.

The facilities of the HGCC system are provided in the form of four groups of services. These are: user interface, calculation management, communication services and data repository services. *User interface services* provide a consistent user interface supporting the process of defining an application, processing of the calculation results and providing communication with the user. *Calculation management services* allocate the calculation processes into cluster nodes and manage execution of the user's application. *Communication services* manage communications between running processes and system kernel, and finally *data repository services* provide a store for all data objects.

HGCC Architecture. The cluster framework consists of several components presented in Fig. 3. The most important are: *MasterApp* – master node application, the main component that is responsible for the user-system communication and calculation management and *SlaveApp* – the computational node application, the component that is responsible for calculations that are performed by the assigned server. Each computational node contains some number of resources. In our framework we distinguish and collect information about two types of such resources: CPUs – central processing units and GPUs – graphics processing units. The computational resource can be in one of the following states: *waiting* – ready for loading a new task to execution, *working* – occupied, calculations are executed and *lost* – lost because of the node failure.

Inter-process Communication. The system implements the master-slave communication scheme. An XML-based communication protocol based on the TCP/IP protocol and BSD sockets is used to perform communication between

master and slave nodes. Our goal was to develop a simple, flexible and failure resistance mechanism.

HGCC System Operation. A user implements the computational task in an object oriented way and defines his problem in the *task descriptor*. The XML Schema specification for building XML files with task description is provided in HGCC. The task descriptor contains: a type of the task, an algorithm, a destination platform and device. All these parameters are mandatory. The rest of this file is filled by parameters specific to a given task. The cluster framework can handle any computational task which was implemented by the user. A committed task is sent to the *MasterApp* component. All parameters defined in the *task descriptor* are parsed inside *MasterApp*. Next, the task is divided into smaller subtasks. *MasterApp* creates the list of such subtasks. They are allocated to the slave nodes, which contain any free resources. Two operations are performed after *SlaveApp* initialization: a plugin list is loaded from a plugin descriptor file, and a socket is opened and wait for *MasteApp*'s connection. The plugin descriptor file contains information about all plugins currently available in the system. Whenever a slave node gets a new set of subtasks to execute it looks for available valid plugin, and loads it to the memory. Next, the control flow inside *SlaveApp* splits, and the newly spawned thread launches calculations stored in the loaded plugin.

4. Case Study Results: Parallel Cryptography

Cryptanalysis and cryptography techniques are natural candidates for massively parallel computations. The algorithms for encryption and decryption of large amounts of data can be easily decomposed and executed in parallel. The popular schemes using symmetric ciphers were found to give a significant speed up when ported to GPU, especially such schemes like Data Encryption Standard (DES), Advanced Encryption Standard (AES) [7]–[10] or BlowFish [11]–[13]. GPU-based implementations of algorithms using asymmetric ciphers (RSA, ECC, NTRU and GGH) are described in the following papers [14]–[18]. In this paper we present the evaluation of the performance of our HGCC-based implementations of symmetric DES and AES cryptography algorithms.

4.1. DES and AES Implementations in HGCC

The HGCC cluster is the general purpose hardware and software system that can be used to solve any complex computing problems that require a processing of large amounts of data (see [19], [20]). In our research, which results are presented in this paper we used HGCC to efficient cryptanalysis and cryptography. The evaluation of selected techniques of cryptanalysis, i.e., the password recovery from hashes are described in [21]. In this paper we focus on effective cryptography working on CPU and GPU units. The numerical results of extensive tests of our implementations of DES and AES algorithms are presented and discussed.

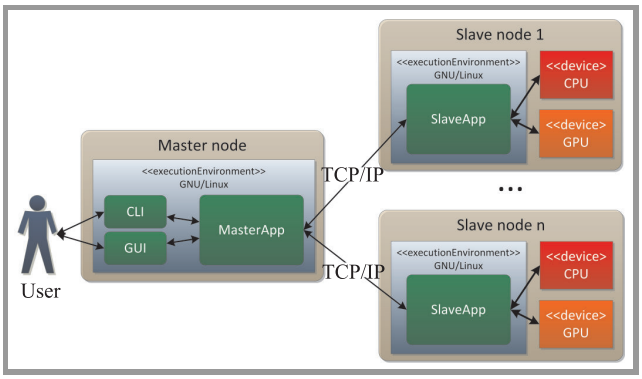


Fig. 3. Core components of the cluster framework.

We performed four series of experiments. The aim of the first series was to test the efficiency of parallel implementations of the DES and AES algorithms on the cluster formed only by the CPU units. Various modes of the algorithms operation were compared. The current modes of operation listed in Table 1 are specified in http://csrc.nist.gov/index.html and in [22]. The simplest of the encryption modes is the electronic codebook (ECB) mode. The message is divided into blocks and each block is encrypted separately. In CBC (cipher-block chaining) mode, each block of plain text is XORed with the previous cipher text block before being encrypted. This way, each cipher text block depends on all plain text blocks processed up to that point. The PCBC (propagating cipher-block chaining) mode was developed to cause small changes in the ciphertext to propagate indefinitely both when decrypting and encrypting. The CFB (cipher feedback) mode that is relative to CBC makes a block cipher into a self-synchronizing stream cipher. The OFB (output feedback) mode makes a block cipher into a synchronous stream cipher. The CTR (counter) mode has similar characteristics to OFB, but also allows a random access property during decryption. It should be pointed that several of listed modes are suited to parallel implementation (see Table 1).

Table 1
Modes of operations of symmetric ciphers

Mode	Parallel encryption	Parallel decryption
ECB	Yes	Yes
CBC	No	Yes
PCBC	No	No
CFB	No	Yes
OFB	No	No
CTR	Yes	Yes

Table 2 and Fig. 4 demonstrate the performance of the single thread CPU-based implementations of the block ciphers: DES, 3DES and AES. The table collects the

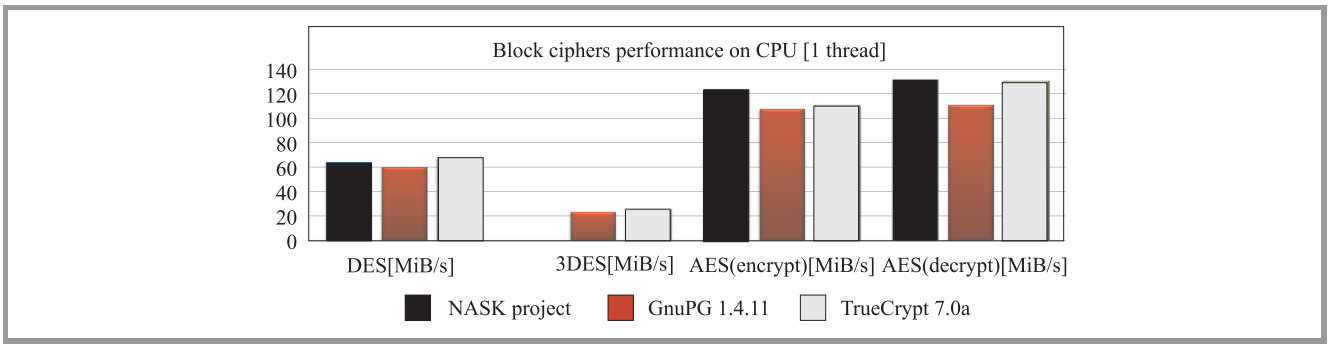


Fig. 4. Block ciphers performance on CPU (1 thread).

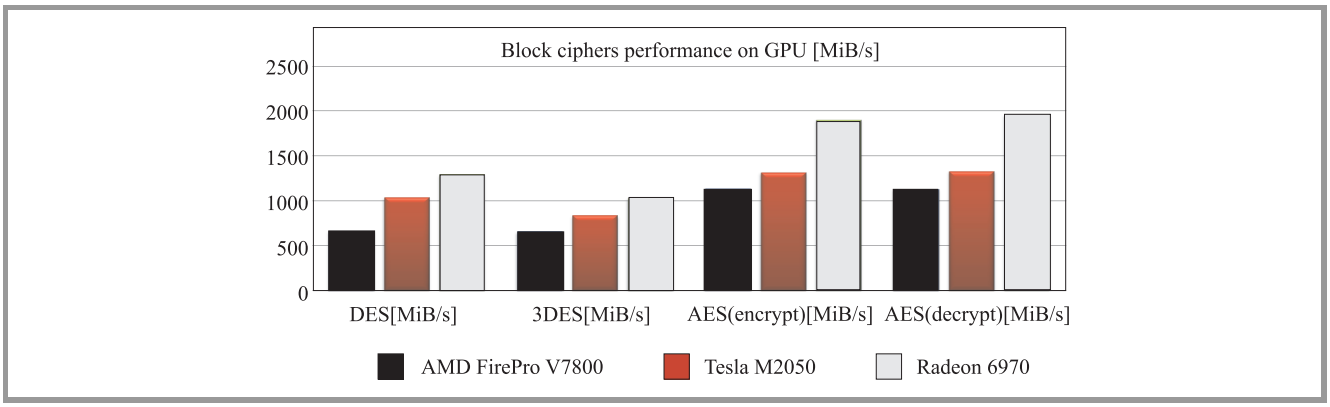


Fig. 5. Block ciphers performance on GPU.

amounts of data in MiB/s (MiB/s = 1,048,576 bytes/s) that were encrypted/decrypted per second in case of all tested algorithms. The efficiency of our implementations of DES and AES (NASK) were compared with the results presented in the Internet: GNU Privacy Guard GnuPG 1.4.11 (http://www.gnupg.org/download/) and free open-source disk encryption software TrueCrypt 7.0a (http://www.truecrypt.org/).

Table 2
Block ciphers performance on CPU (1 thread) in MiB/s

Algorithm	Implementation		
	NASK	GnuPG 1.4.11	TrueCrypt 7.0a
DES	63.82	60.01	67.95
3DES		23.10	25.48
AES(encrypt)	123.70	107.50	110.60
AES(decrypt)	131.80	110.80	130.70

The presented results show that the efficiency of our implementations of DES and AES on CPU is similar to the results provided by other projects. It is worth to mention that both GnuPG and TrueCrypt are widely used products of teams that have extensive experience in cryptography.

The aim of the second series of experiments was to compare the efficiency of the DES and AES implementations on different GPUs provided by various vendors. The calculations were carried out on three types of GPU units: AMD FirePro V7800, NVIDIA Tesla M2050 and AMD Radeon 6970. Table 3 and Fig. 5 present the amounts of data in MiB/s that were encrypted/decrypted per second in case of all tested algorithms and decryption and encryption operations.

Table 3
Block ciphers performance on GPU in MiB/s

Algorithm	Graphics Processing Unit		
	AMD FirePro V7800	NVIDIA Tesla M2050	AMD Radeon 6970
DES	660.73	1038.02	1295.60
3DES	658.11	837.67	1031.73
AES(encrypt)	1135.96	1316.82	1901.25
AES(decrypt)	1129.54	1329.62	1963.91

In general, the GPU-based implementations of all algorithms were much more efficient than implementations working only on the CPU unit. The best results were obtained for the Radeon 6970 graphics processing unit.

The aim of the next series of experiments was to compare the performance of two implementations of the AES algorithm. The original AES implementation was compared with implementation utilizing an Advanced Encryption Standard – New Instruction Set (AES-NI) extension. New Instruction Set is an extension to the x86 instruc-

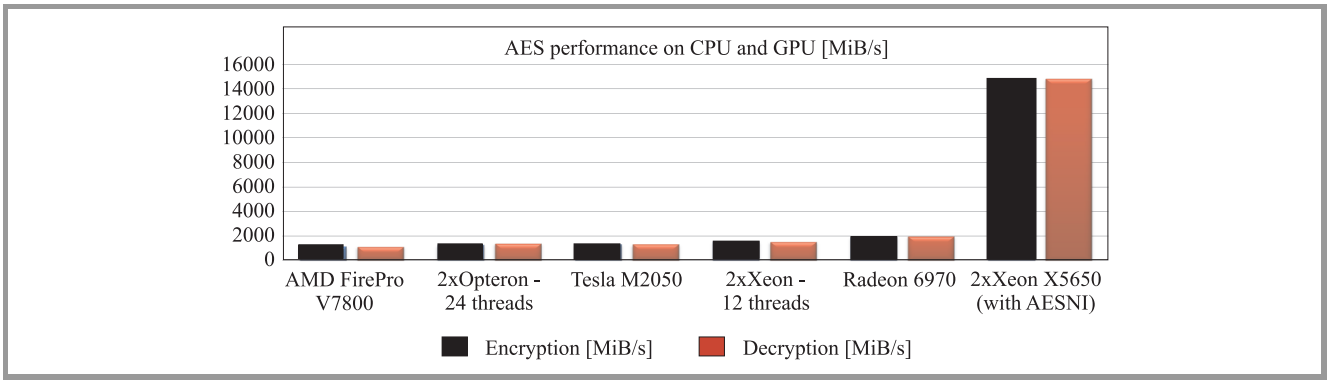


Fig. 6. Performance of the AES algorithm implementations with/without AES-NI extension; CPU and GPU.

tion set architecture for microprocessors from Intel and AMD (http://ark.intel.com/). The purpose of this instruction set is to improve the speed of applications performing encryption and decryption using AES, which is an industrial standard nowadays. In our cluster only Intel Xeon X5650 Westmere processors provides AES-NI extension. Unfortunately none of our processors provides support for Advanced Vector Extensions (AVX), so we were not able to assess impact of AVX instruction set on AES performance. Table 4 and Fig. 6 demonstrate the performance of AES executed on three types of GPU and two types of CPU units. We can see that the application of the new instruction set causes massive acceleration of the AES algorithm.

Table 4
Performance of the AES algorithm implementations with/without AES-NI extension; CPU and GPU

Processing Unit	AES-NI extension	Encryption [MiB/s]	Decryption [MiB/s]
AMD FirePro V7800	no	1135.96	1129.54
NVIDIA Tesla M2050	no	1316.82	1329.62
AMD Radeon 6970	no	1901.25	1963.91
2xOpteron – 24 threads	no	1272.00	1394.40
2xXeon – 12 threads	no	1546.80	1533.60
2xXeon – 12 threads	yes	14848.80	14841.60

Table 5
Scalability of the AES algorithm

Processor	Speedup		
	1 node	2 nodes	4 nodes
Xeon X5650	1	1.95	3.71
Opteron 6172	1	1.93	3.73

The aim of the last series of experiments was to present the efficiency of our cluster system. In this paper we present the evaluation of the AES implementation in two subclusters: the first formed by four Opteron processing units and the second one formed by four Xeon processing units. As it can be seen in Table 5 the AES algorithm scales up very well – the speed up value for four nodes is between 3.71 and 3.73.

5. Summary and Conclusion

The paper provides a short overview of the components of our heterogenous cluster system integrating CPU and GPU devices from various vendors. We described the hardware architecture and the software framework that form our cluster. The cluster system was designed to be powerful, effective, scalable, flexible, and easy to use. It is especially useful in complex calculations and parallel processing of large volumes of data in which a speed of calculation and data decomposition are of essence. Cryptography algorithms are natural candidates for massively parallel computations in GPU/CPU clusters. Our experimental results presented in this paper demonstrate the effectiveness and scalability of the HGCC cluster system, and confirm that the direction to speed up cryptography techniques is to port them to GPU units. As a final observation we can say that heterogeneous computing systems offer a new opportunity to increase the performance of parallel HPC applications on clusters, by combining traditional CPU and general purpose GPU devices.

Acknowledgment

The work was supported by the National Centre for Research and Development (NCBiR) under grant number O R00 0091 11.

References

[1] F. Berman, G. Fox, and A. J. G. Hey, *Grid Computing: Making the Global Infrastructure a Reality*. New York: Wiley, 2003.

[2] A. Karbowski and E. Niewiadomska-Szynkiewicz, *Parallel and distributed computing (in Polish)*. WUT Publishing House, 2009.

[3] Wen-Mei W. Hwu (Ed.), *GPU Computing Gems Emerald Edition*. Morgan Kaufman, 2011.

[4] R. Lottiaux, B. Boissinot, P. Gallard, G. Vallee, and C. Morin, "Openmosix, openssi and kerrighed: a comparative study", in *Proc. IEEE Int. Symp. Cluster Comput. and the Grid CCGrid'05*, Cardiff, UK, 2005, vol. 2, pp. 1016–1023.

[5] A. Barak and A. Shiloh, "The mosix virtual opencl (VCL) cluster platform", in *Proc. Intel Eur. Res. Innov. Conf.*, Leixlip, Ireland, 2011, p. 196.

[6] V. Kindratenko, J. Enos, G. Shi, M. Showerman, G. Arnold, J. Stone, J. Phillips, and W. Hwu, "GPU clusters for high-performance computing", in *Proc. Worksh. Paral. Programm. Accelerator Clust. PPAC'09*, New Orleans, LA, USA, 2009.

[7] A. Di Biagio, A. Barenghi, G. Agosta, and G. Pelosi, "Design of a parallel AES for graphics hardware using the CUDA framework", in *Proc. 23rd IEEE Int. Parallel Distrib. Proces. Symp. IPDPS 2009*, Rome, Italy, 2009, pp. 1–8.

[8] J. W. Bos, D. A. Osvik, and D. Stefan, "Fast implementations of AES on various platforms", Tech. rep., Cryptology ePrint Archive, Report 2009/501, 2009 [Online]. Available: http://eprint.iacr.org

[9] D. Le, J. Chang, X. Gou, A. Zhang, and C. Lu, "Parallel aes algorithm for fast data encryption on GPU", in *Proc. 2nd Int. Conf. Comp. Engin. Technol. ICCET 2010*, Chengdu, China, 2010, vol. 6, pp. V6–1.

[10] C. Mei, H. Jiang, and J. Jenness, "CUDA-based aes parallelization with fine-tuned GPU memory utilization", in *Proc. IEEE Int. Symp. Parallel & Distrib. Proces., Worksh. and Phd Forum IPDPSW 2010*, Atlanta, Georgia, USA, 2010, pp. 1–7.

[11] C. Li, H. Wu, S. Chen, X. Li, and D. Guo, "Efficient implementation for MD5-RC4 encryption using GPU with CUDA", in *Proc. 3rd Int. Conf. Anti-Counterfeiting, Secur., Identif. Commun. ASID 2009*, Hong Kong, China, 2009, pp. 167–170.

[12] Z. Wang, J. Graham, N. Ajam, and H. Jiang, "Design and optimization of hybrid MD5-blowfish encryption on GPUs", in *Proc. Int. Conf. Paral. Distrib. Proces. Techn. Appl. PDPTA'11*, Las Vegas, Nevada, USA, 2011, pp. 18–21.

[13] G. Liu, H. An, W. Han, G. Xu, P. Yao, M. Xu, X. Hao, and Y. Wang, "A program behavior study of block cryptography algorithms on GPGPU", in *Proc. 4th Int. Conf. Frontier Comp. Sci. Technol. FCST'09*, Shanghai, China, 2009, pp. 33–39.

[14] S. Fleissner, "GPU-accelerated montgomery exponentiation", in *Proc. Int. Conf. Computat. Sci. ICCS 2007*, Beijing, China, 2007, pp. 213–220, 2007.

[15] O. Harrison and J. Waldron, "Efficient acceleration of asymmetric cryptography on graphics hardware", in *Proc. 2nd Int. Conf. Cryptol. in Africa AFRICACRYPT 2009*, Gammarth, Tunisia, 2009, pp. 350–367.

[16] A. Moss, D. Page, and N. P. Smart, "Toward acceleration of RSA using 3D graphics hardware", in *Proc. 11th IMA Int. Conf. Cryptography and Coding*, Springer, 2007, pp. 364–383.

[17] J. Hermans, F. Vercauteren, and B. Preneel, "Speed records for NTRU", in *Proc. Topics in Cryptol. CT-RSA 2010*, San Francisco, CA, USA, 2010, pp. 73–88.

[18] J. Hermans, F. Vercauteren, and B. Preneel, "Implementing NTRU on a GPU", Darmstadt University of Technology, Darmstadt, Germany, 2009.

[19] M. Marks, "Enhancing wsn localization robustness utilizing HPC environment", in *Proc. Eur. Conf. Model. Simul. ECMS 2012*, Koblenz, Germany, 2012, pp. 167–170.

[20] W. Szynkiewicz and J. Błaszczyk, "Optimization-based approach to path planning for closed chain robot systems", *Int. J. Appl. Mathema. Comp. Sci. ACMS*, vol. 21, no. 4, pp. 659–670, 2011.

[21] M. Marks, J. Jantura, E. Niewiadomska-Szynkiewicz, P. Strzelczyk, and K. Gózdz, "Heterogenous GPU/GPU cluster for high performance computing in cryptography", *Comp. Sci.*, vol. 14, no. 2, pp. 63–79, 2012.

[22] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. New York: CRC Press, 1996.

Ewa Niewiadomska-Szynkiewicz D.Sc. (2005), Ph.D. (1995), Professor of control and information engineering at the Warsaw University of Technology, head of the Complex Systems Group. She is also the Director for Research of Research and Academic Computer Network (NASK). The author and co-author of three books and over 120 journal and conference papers. Her research interests focus on complex systems modeling and control, computer simulation, global optimization, parallel computation, computer networks and ad hoc networks. She was involved in a number of research projects including EU projects, coordinated the Groups activities, managed organization of a number of national-level and international conferences.
E-mail: ens@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

E-mail: ewan@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Michał Marks received his M.Sc. in Computer Science from the Warsaw University of Technology, Poland, in 2007. Currently he is a Ph.D. student in the Institute of Control and Computation Engineering at the Warsaw University of Technology. Since 2007 with Research and Academic Computer Network (NASK). His research area focuses on wireless sensor networks, global optimization, distributed computation in CPU and GPU clusters, decision support and machine learning.
E-mail: M.Marks@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19
00-665 Warsaw, Poland

E-mail: M.Marks@nask.pl
Research and Academic Computer Network (NASK)
Wawozowa st 18
02-796 Warsaw, Poland

Jarosław Jantura received his M.Sc. in Electrical Engineering from the Kielce University of Technology, Poland, in 2004. Currently he works as Technical Project Leader at NASK. His interests focus on software engineering and computer graphics.

E-mail: jaroslawj@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Mikołaj Podbielski received his M.Sc. in Telecommunications from the Warsaw University of Technology, Poland, in 2010. Currently he works as software engineer at NASK. His interests focus on software engineering and GPGPU.

E-mail: mikolajp@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Paper

Secure Biometric Verification Station Based on Iris Recognition

Adam Czajka[a,b] and Krzysztof Piech[a]

[a] Research and Academic Computer Network (NASK), Warsaw, Poland

[b] Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland

Abstract—This paper describes an application of the Zak-Gabor-based iris coding to build a secure biometric verification station (SBS), consisting of a professional iris capture camera, a processing unit with specially designed iris recognition and communication software, as well as a display (LCD). Specially designed protocol controls the access to the station and secures the communication between the station and the external world. Reliability of the Zak-Gabor-based coding, similarly to other wavelet-based methods, strongly depends on appropriate choice of the wavelets employed in image coding. This choice cannot be arbitrary and should be adequate to the employed iris image quality. Thus in this paper we propose an automatic iris feature selection mechanism employing, among others, the minimum redundancy, maximum relevance (mRMR) methodology as one, yet important, step to assess the optimal set of wavelets used in this iris recognition application. System reliability is assessed with approximately 1000 iris images collected by the station for 50 different eyes.

Keywords—application of biometrics, feature selection, iris recognition, Zak-Gabor-based iris coding.

1. Introduction

Iris recognition has recently emerged as one of the top biometric authentication methods due to its accuracy and outstanding identification efficacy. It is also commonly believed that the pattern of iris tissue is highly stable throughout the human live, although recent scientific notifications start to surprisingly suggest the opposite hypothesis. However, without a doubt, iris recognition became a mature technology supported by numerous implementations in places requiring reliable identity verification, and the most common applications concern border and physical control. This paper is conformable with this trend, as we apply the iris recognition as a key element of a secure verification station, being a server of biometric-based verification. The station is autonomous, i.e., it consists of image capture hardware, the processing unit with the operating system and the display for communication with verified subjects. The electronic communication with the station is secured by a protocol specially designed to the purpose of this application.

The iris recognition used in this work is based on original methodology employing Zak-Gabor transformation [1]. The optimal features selection procedure consists of two stages: selecting the best iris features – in terms of an iris recognition efficiency – in the first stage, and selecting the optimal Zak-Gabor-based coding parameters based on the results of the first stage [2]. For reading fluency, the applied Zak-Gabor-based coding is briefly explained in Section 2, some remarks related to the iris template creation and matching are provided in Subsection 2.4, and an in-depth explanation of the feature selection mechanism is presented in Section 3.

Selection of coding parameters, i.e., the optimal wavelet families emphasizing the relevant individual iris features, is strictly dependent on the database, in particular on the quality of iris images used. The resulting parameters estimate an optimal, yet unknown, configuration of wavelet transformation adequate in the iris template creation. However, when applied to other image sets (not used at the estimation stage), these transformation parameters typically result in a lower accuracy than expected (in particular the genuine comparison results deteriorate, e.g., see p. 289 in [3]). It is thus reasonable to adapt – if possible – the coding parameters to the expected iris image quality by estimating the optimal features using the database collected in the assumed scenario, i.e., employing target equipment operated in a target (or precisely modeled) environment, and applying all possible procedures that are expected at the operational stage.

However, feature selection is a very demanding process, especially when rich datasets are to be applied and the feature space is significantly large. Therefore, achieving a global minimum of the recognition error cannot be often guaranteed within acceptable time (or cannot be achieved at all due to a huge number of calculations that make the search process infeasible), and we have to be satisfied by quasi-optimal solutions. Repeating this process prior to any application of our recognition methodology may be thus annoying. To make the iris feature selection an easy process, and further to select satisfying parameters of the Zak-Gabor-based coding, we have built a tool that performs this task automatically for provided dataset of iris images, Section 4.

To build a secure verification station we decided to use one of the existing iris capture cameras and our choice was mainly motivated by speed of capture and ease of use, due to high priority of practical aspects of the station. As we expected differences in quality of images captured by the

station and images used to build the original method, we finally applied the developed feature selection tool to adapt the coding and improve the accuracy. The achieved three-fold reduction of the EER justifies the need of adaptation for a new image quality. The station development is described in Section 5.

2. Zak-Gabor-Based Iris Recognition

This section briefly presents the Zak-Gabor-based iris coding [1], [2], developed earlier by the first author, and used in this work to build a tool for automatic iris features selection, which finally is applied in the developed secure station.

2.1. Iris Images and their Segmentation

Iris recognition starts from the acquisition of an iris image of sufficient quality. The raw image contains the iris but also its surroundings, and the iris is often disturbed by occlusions, thus it has to be processed prior to feature extraction, Fig. 1. Building a map classifying the image points into those representing the iris and lying outside the iris is called the segmentation. Although the feature extraction routines are directly responsible for delivering iris features, the segmentation process mostly influences a reliability of iris recognition, and most of current endeavors go towards development of robust iris segmentation methods.

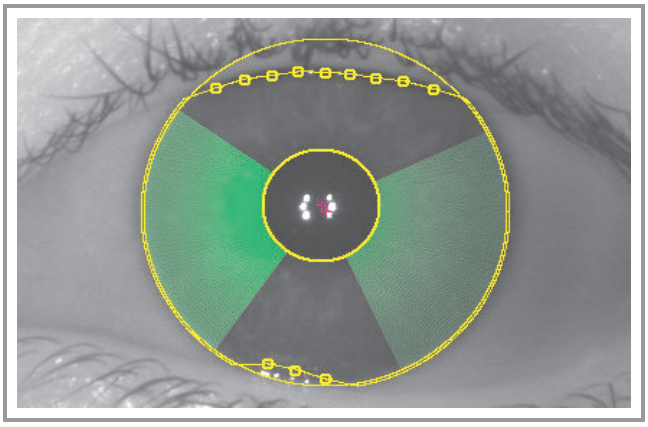


Fig. 1. Iris image captured by the station and the segmentation result. The two circles approximate inner and outer boundaries of the iris. Small empty circles point to the detected occlusions, and the broken line approximates them between detection points. Two sectors, automatically selected by the segmentation procedure, cover the iris portion used in feature extraction.

The segmentation method used in this work comprises of two main stages: localization of the inner and outer boundaries of the iris, and localization of any disruptions occluding the iris tissue. Neither the iris nor the pupil are circles, nevertheless, due to the simplicity and speed of algorithm implementations, it is a common practice to model the iris boundaries by circular, non-concentric shapes. The method applied in this paper employs a modified Hough transform to detect circular shapes, and the coarse-to-fine analysis is applied to speed up the calculations, i.e., rough positions of the pupil and the iris are first determined in a low resolution, and higher resolutions are used to precise the final result.

Two determined circles representing inner and outer boundaries only roughly approximate the actual iris region, due to occlusions that typically exist in iris images. To detect local non-uniformity within the iris body, a set of angular directions is constructed, originating from the iris center in which the iris texture is analyzed. Among all the angular directions, a set of *reference directions* are singled out, which cover those sections that are known to be always free of occlusions, and they are used for calculation of the maximum allowed non-uniformity. Once the maximum allowed non-uniformity is set, the iris region is analyzed for each analysis direction and an occlusion is detected when the irregularity in a given direction exceeds the reference. This procedure constructs an occlusion map as a series of radii, each representing the distance between the iris center and the boundary of the occlusion (cf. small empty circles in Fig. 1).

Based on localized occlusions, the method sets two independent iris sectors, each of 90° in angular width, to be further used for features extraction. The choice of angular width was made at the stage of development of the Zak-Gabor-based method, and it was based on experiments with the iris enrollment images. Since users of the secure station are not forced to open their eyes in a particular way during capture, it is concluded that once the eyelid coverage is too high, the system may ask for the eye to be opened more widely to finally extract two such sectors with minimum effort by the user. As ISO recommends [4] to have at least 70% of iris body not obscured, the assumption that two iris sectors – constituting only 50% of the iris body – free from occlusions can be found, seems not to be very demanding.

Iris texture analysis may be qualified as a 2D pattern analysis task, yet it is often simplified to a set of 1D problems. The Zak-Gabor-based method maps two iris sectors into R one-dimensional P point functions. We further call these functions *stripes*, as they represent image intensity distributed along angular direction, and for each angle the resulting value is calculated within small radial neighborhood, resembling the division of each iris sector into $R/2$ stripes. These one-dimensional functions are further used in the Zak-Gabor-based method to extract iris features.

In order to ascertain whether the acquired image is of the required quality (in terms of focus and iris body availability), the Zak-Gabor-based method investigates two additional image attributes: focus factor and the iris body coverage by eyelids. These attributes are basis for separate 'experts' judging on the image quality, and giving a binary decision whether the image passed the test. It is assumed

that image quality (from the iris recognition perspective) is sufficient if both experts' answer is affirmative.

2.2. Zak-Gabor-Based Iris Features

The Zak-Gabor-based method characterizes a discrete-time signal in the joint time-frequency domain, describing its stationary energy distribution locally, thus finding the distribution of signal energy in local (possibly overlapping) time segments. As we deal with images, not time series, the 'time' variable is replaced by the 'position' variable in this work. The Zak-Gabor-based method performs this local analysis by a family of wavelets, each characterized by a quadruple: scale, frequency and radial/angular positions. Since it is difficult and not recommended to set an arbitrary scales, the natural extension of such position-frequency analysis is to allow the scale and to be adapted independently of the frequency coordinate. Thus directing these calculations toward the so called *wavelet packet* analysis. Following [2], we explain briefly the calculation principles of the Zak-Gabor coefficients, being a base for iris features used in this work.

Let g_s be a one-dimensional Gaussian function characterized by scale index s, sampled at points $0 \ldots P-1$, namely

$$g_s(p) = e^{-\pi\left((p+\frac{1}{2})/2^s\right)^2} \tag{1}$$

where $s = 2, \ldots, S$ and $S = 8$, and the factor of $\frac{1}{2}$ lets g_s to be symmetric over the sampling grid when P is even. The *Gabor elementary function* (GEF) [5] is defined as shifted and modulated version of g_s, namely

$$g_{mk;s}(p) = g_s(p - mK)e^{ikp2\pi/K}, \quad p = 0 \ldots P-1. \tag{2}$$

Let M be the number of translations of g_s, and K be the number of frequency shifts, where m and k denote position and frequency shifts, respectively ($m = 0, \ldots, M-1$, $k = 0, \ldots, K-1$), and g_s is wrapped around the P-point domain. Zak-Gabor-based iris coding applies critical sampling and always takes $M = P/K$. Let f_ℓ be the intensity function defined on a stripe ℓ. The finite discrete Gabor transform of f_ℓ is defined as a set of complex-valued coefficients $a_{mk;s\ell}$ that satisfy the Gabor signal expansion relationship, namely

$$f_\ell(p) = \sum_{m=0}^{M-1}\sum_{k=0}^{K-1} a_{mk;s\ell} g_{mk;s}(p), \quad p = 0 \ldots P-1 \tag{3}$$

and $K = 2^s$ is set in further analysis. Note that the number S of scales together with the stripe size P determine both M and K.

The set of $a_{mk;s\ell}$ coefficients is a base of iris features. Gaussian-shaped functions, used in the Zak-Gabor transform, are however not orthogonal (the inner product of any two of all functions is nonzero), therefore its coefficients cannot be determined in a simple way, and Zak's transform is applied for this purpose. The discrete finite Zak transform $\mathscr{Z} f_\ell(\rho, \phi; K, M)$ of a function f_ℓ sampled equidistantly at P points is defined as the one-dimensional discrete Fourier transform of the sequence $f_\ell(\rho + jK)$, $j = 0, \ldots, M-1$, namely [5]

$$\mathscr{Z} f_\ell(\rho, \phi; K, M) = \sum_{j=0}^{M-1} f_\ell(\rho + jK)e^{-ij\phi 2\pi/M} \tag{4}$$

where $\phi = 0, 1, , M-1$, $\rho = 0, 1, \ldots, K-1$ and $M = P/K$. Application of the discrete Zak transform (4) to both sides of (3) yields

$$\mathscr{Z} f_\ell(\rho, \phi; K, M) = \mathscr{F} a_{s\ell}(\rho, \phi; K, M)\, \mathscr{Z} g_s(\rho, \phi; K, M) \tag{5}$$

where $\mathscr{F} a_{s\ell}[\rho, \phi; K, M]$ denotes the discrete 2D Fourier transform of an array of $a_{s\ell}$, representing the coefficients determined for the iris stripe ℓ and scale s, and $\mathscr{Z} g_s[\rho, \phi; K, M]$ is the discrete Zak's transform of the Gaussian window g_s. The expansion coefficients $a_{mk;s\ell}$ can be thus recovered from the product form (5) and choosing K and M to be a power of 2 yields possibility of FFT application, thus making computation times proportional to those in the FFT. As this way of calculating the Gabor expansion coefficients is often called the Zak-Gabor transform [5] (instead of simply Gabor transform), we further call $a_{mk;s\ell}$ coefficients as the *Zak-Gabor coefficients*.

Zak-Gabor-based iris coding defines the signs of the real and imaginary parts of Zak-Gabor coefficients $a_{mk;s\ell}$ as iris feature set $\mathbb{B}$, namely

$$\mathbb{B} = \left\{\mathrm{sgn}(\Re(a_{mk;s\ell}), \mathrm{sgn}(\Im(a_{mk;s\ell}))\right\} \tag{6}$$

where $m = 0, \ldots, M-1$, $k = 0, \ldots, K-1$, $\ell = 0, \ldots, R-1$ and $s = 2, \ldots, S$. Since the Fourier transform of real signals (e.g., iris stripes f_ℓ are real) consists of two parts being complex conjugates of each other, for each position m the coefficients with the frequency index $k > K/2$ are ignored. Since $M = P/K$, for each s there are $(N-1)P/2$ coefficients determined. Taking into account that this analysis is carried out for all iris stripes, and remembering that $R = 32, S = 8$ and $P = 512$, the total number of coefficients calculated for the iris image is $R(S-1)P/2 = 57,344$. Both real and imaginary parts are coded separately by one bit, hence $N = \#(\mathbb{B}) = 114,688$ features (bits) are achieved.

2.3. Iris Features Matching

The order of features (bits) is kept identical for all images and thus the matching requires only a XOR operation between two feature sets. The Hamming distance is applied (as it is typically done for binary feature vectors) to calculate the score ξ, namely

$$\xi = \frac{1}{N}\sum_{n=0}^{N-1}(b_n^{(1)} \text{ XOR } b_n^{(2)}) \tag{7}$$

where $b_n^{(i)}$ is the n-th bit of i-th sample. Factor $\frac{1}{N}$ makes $\xi \in \langle 0, 1\rangle$.

2.4. Iris Template Creation and Verification

Iris templates used by the Zak-Gabor-based coding consist of iris code bits and positions of the individual sectors (calculated within the segmentation procedure) for which the iris features are determined. The sector positions are necessary to allow calculating the code at the same angular positions each time the images have to be matched. The implementation of the Zak-Gabor-based which is used in this work [6] may use any number of iris images (starting from just one) to create the iris template, however, capturing more than one image at the enrollment is recommended. In the latter case we may take advantage of the Zak-Gabor-based implementation that selects the best iris code among a number of those calculated for the enrollment images. Namely, if multiple images are available at the enrollment time, the template creation procedure first rotates all the images to the one representative which is the least rotated to all the remaining images. Next, for all images the iris sectors are determined and their average positions are taken. The consistency of the codes is checked by calculating the comparison scores between the analyzed code and the codes related to the remaining enrollment images, obtained for these new (averaged) sector positions. To finalize the template creation, all the comparison scores must fall below the acceptance threshold, which denotes that the enrollment images were of sufficient quality to deliver information about the iris texture. If any of the matching results exceeds the acceptance threshold, the procedure allows for a replacement of the defective image and calculations are partially repeated.

In contrary to the enrollment procedure, the verification should proceed quickly, in particular only single image is acquired. However, the absolute eyeball slope cannot be assessed accurately in one-eye capture system, as it was applied in the secure station, and no such information is linked to the template. Once an eye is rotated during capture relatively to the images employed in the enrollment process, corresponding features apply to different parts of the iris, thus making the features inadequate. Using a raw iris image at the verification stage, without correcting eye rotation may lead to false rejections. The implementation of the Zak-Gabor-based coding solves this problem by generating a series of iris templates at the enrollment stage and each of the generated template corresponds to one micro-rotation of the original iris (in both directions). The angles of these micro-rotations are not equidistantly placed in the angular axis, and were selected according to the sample distribution of observed rotations in this application. It produced a map of rotations, with greater number of elements near zero (small rotations are more probable) and less elements near the maximum rotations observed (as they are still probable to occur, yet less than the smallest ones). It slightly extended the enrollment procedure (as the Zak-Gabor-based features has to be calculated several times), yet the eyeball rotation compensation at the verification stage is done in the blink of the eye, as only a few (instead of one) XOR operations are needed to conclude the match. Introducing this eyeball rotation correction was compulsory, and neglecting this compensation would lead to significant and unacceptable false rejections.

3. Iris Features Selection

Not all elements in $\mathbb{B}$ are useful for iris recognition, and only a subset of the features in $\mathbb{B}$ constitutes an optimal feature set $\mathbb{B}^0$. However, simple selection of optimal features yields to a subset of coefficients indexed by selected scales and frequencies, but also by selected positions (angular and radial). There is however no rationale behind selecting only subsets of positions, as entire areas of both iris sectors are considered useful in the recognition.

Therefore, a two-stage procedure of Zak-Gabor-based iris features selection proposed in [2] is used in this work. In the first stage, the optimal parameter quadruples are selected that yield features maximizing the classification margin between the same and different irises. This is an exhaustive computation problem, yet many feature selection routines may be applied here (e.g., we may use Fisher's information related to each feature to sort them out and iteratively check the features' usefulness). In the second stage, it is checked how the scale-frequency pairs are populated by the optimal features (the more a scale-frequency pair is populated, the more it is significant in iris coding). The latter stage allows for selecting scales and frequencies optimal for a given database of iris images, being a good prediction of optimal scale-frequency pairs in iris recognition.

Peng *et al.* [7] proposed to use a *mutual information* (i.e., a Kullback-Leibler divergence of a product $P(X)P(Y)$ of two marginal probability distributions $P(X)$ and $P(Y)$ from the joint probability distribution $P(X,Y)$) to select best features employed in classification problems. Therefore, their selection criterion is based on the maximum statistical dependency between the variables X and Y. Due to difficulties in direct implementation of the maximum dependency condition, Peng *et al.* developed an equivalent form of this criterion, called the minimal-redundancy maximum-relevance (mRMR). In this paper we are applying the mRMR search methodology in the first stage to select the optimal feature set (instead of applying Fisher's information, as it was used in [2]). To make this process an automatic one, and to allow an easy adaptation of Zak-Gabor-based coding to various iris datasets, the methodology was integrated in the form of a convenient tool.

3.1. Estimation Database

Estimating optimal parameters for the Zak-Gabor-based coding requires suitable data. A database of 1000 iris images was collected by the station operated in similar environmental conditions, as the expected in final application, guarantying the appropriate quality of images. Twenty five volunteers, males and females, aged between 20 and

40 years, participated in the iris images acquisition process. The acquisition station was organized in a way that helped to take advantage of natural lighting to narrow the pupil. The quality of collected images was manually analyzed, and images of poor quality, or those generating segmentation failures were removed. Censoring, not carried out in system evaluation, is necessary when the optimal coding parameters have to be established. Any failures in delivering good quality data to the feature selection mechanism may have fatal consequences of selecting features not relevant to given iris tissue. Finally, in this work we use 946 samples representing 50 different eyes.

3.2. Stage I: Selection of Optimal Features $\mathbb{B}^0$

Zak-Gabor-based coding, being a member of wavelet packets family methods, analyses both the scale and frequency interdependently, so they should be considered simultaneously in feature selection. The optimal selection of features is complicated, since the most common frequencies that characterize all iris images cannot be guessed a priori due to significant and undetermined iris texture variability.

In general, when introducing new elements to the feature set, one expects the system to behave better in the sense of unambiguous biometric recognition. To assess the usefulness of a given feature (or feature set) we have to calculate, e.g., equal error rate (or a similar error measure, which is usually an increasing function of the feature set size), and compare the result with those obtained for alternative configuration of features. Single equal error rate may be obtained for a particular dataset of iris images, given the actual variant of the coding. This means that every selection of Zak-Gabor-based features requires recalculating all genuine and impostor scores. As the last step is extremely exhaustive, one may find alternative methods of feature usefulness assessment.

The mRMR method used in this paper, similarly to the routine employing Fisher's information, does not require calculation of the genuine and impostor scores each time we want to judge about the usefulness of a particular feature. A reference implementation of mRMR method has been made public [8], thus its incorporation into our feature selection tool was straightforward and it saved development time.

All features belonging to the set $\mathbb{B}$ were sorted in descending order of their usefulness, expressed as their mutual information, ending up with a set $\mathbb{B}^{\text{sorted}}$. Then each first N' features constitute the intermediate iris binary template that is used in EER calculation. Note that during the search procedure required to sort elements of $\mathbb{B}$ into $\mathbb{B}^{\text{sorted}}$ we need the genuine and impostor scores only when probing subsets of the sorted set of features, and not for all combinations of features.

The EER is one of many possible measures of a biometric system reliability. EER may be however useless when it falls to 0, thus we need other measure assessing the usefulness of a given set of features. For this purpose we cumulate both sample mean and sample variances of comparison scores ξ in the form of the so-called *decidability* factor (*detectability* or *d-prime*), namely

$$d' = \frac{|\overline{\xi}_g - \overline{\xi}_i|}{\sqrt{\frac{1}{2}\left(\overline{\overline{\xi}}_g + \overline{\overline{\xi}}_i\right)}} \tag{8}$$

where $\overline{\xi}$ and $\overline{\overline{\xi}}$ denote sample mean and sample variance of ξ, respectively. The further the mean values are located for same variances, the better is the separation of distributions. Similarly, keeping the same $\overline{\xi}_g$ and $\overline{\xi}_i$ and simultaneously narrowing $\overline{\overline{\xi}}_g$ and $\overline{\overline{\xi}}_i$ one may get higher level of distinction between genuine and impostor patterns. Consequently, the value of d' estimates the degree by which the distributions of ξ_g and ξ_i overlap (the higher d' is, the lower is the overlap).

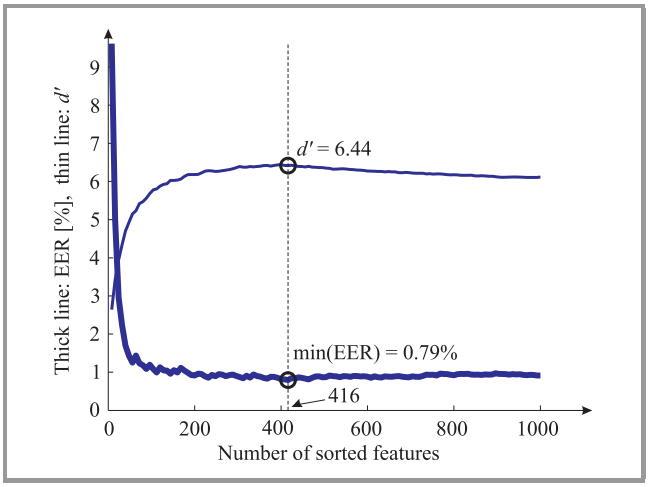


Fig. 2. EER (thick line) and decidability d' (thin line) vs. the number of sorted iris features. Best solution (minimizing the EER), achieved for first 416 features, and the corresponding d' are marked by circles.

Starting from a small number of best features from $\mathbb{B}^{\text{sorted}}$, we iteratively add new features (yet keeping the $\mathbb{B}^{\text{sorted}}$ order), each time building a new iris recognition system. According to observations (Fig. 2), the system reliability (in terms of the EER and d') first increases then deteriorates once the feature set enlarges, and the minimum EER = 0.79% for $N^0 = 416$ can be found. Figure 3 depicts the distribution of genuine and impostor scores achieved on the estimation database of iris images. The set $\mathbb{B}^0$ of optimal features is consequently used in the second stage, namely in the estimation of feature families (and the final iris coding configuration).

As each coefficient $a_{mk;s\ell}$ (and thus each feature in $\mathbb{B}$) is positioned within the iris sector, we may analyze a population of the iris sectors by selected features. Experiments show that the population of the iris sectors by their features is uneven, Fig. 4. Areas located in the middle of iris sectors are apparently more attractive, while the boundary parts are almost neglected by the feature selection methodology. This behavior related to the angular positions is easy

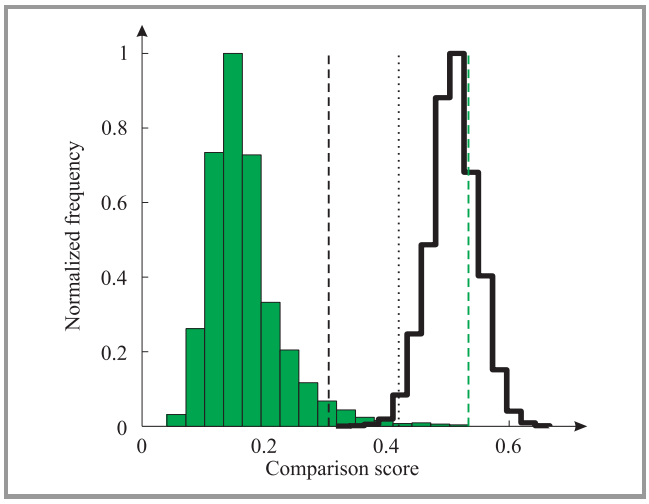


Fig. 3. Genuine (solid bars) and impostor (thick line) scores distribution achieved for a system variant employing 416 optimal features, i.e. the set $\mathbb{B}^0$ (cf. Fig 2). Dashed lines show the worst scores (minimum impostor and maximum genuine) and the dotted line points the acceptance threshold value set to calculate the EER.

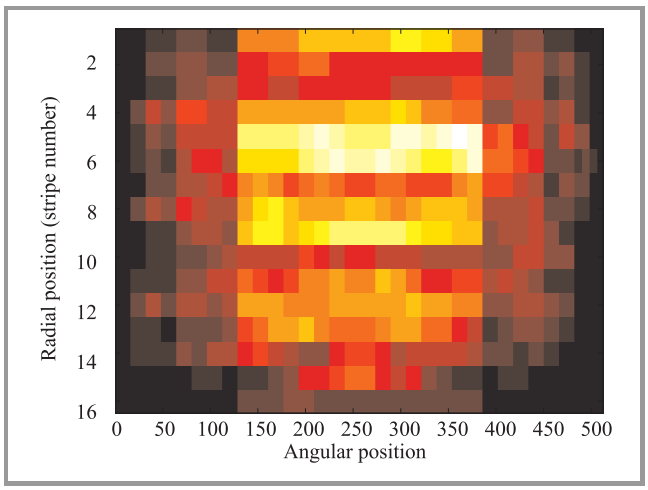


Fig. 4. The population of the iris sectors by their features. Values are averaged for two iris sectors, and lighter color denotes higher usage of the sector area in the optimal feature set $\mathbb{B}^0$.

to be explained, as Zak-Gabor-based coding treats each iris stripe as a periodic function (this assumption comes from the principles of Zak's transform application). The same assumption, however, results in incorrect iris features calculated for non-continuity points at zero positions, i.e., features with weak abilities in distinguishing between same and different irises. Hence, the feature selection methodology correctly neglects those element of $\mathbb{B}$. However, the interpretation of the selection of middle elements in radial direction should refer to the anatomy of the iris, and the result may prove that the population of individual areas within the iris is uneven, with more discriminating parts located closer to the pupil than to the sclera. This interesting observation is worth of making further investigation, yet larger iris image datasets should be used.

3.3. Stage II: Selection of Optimal Feature Families $\mathbb{B}^*$

The second stage of feature selection, proposed for the Zak-Gabor-based coding, considers partitions of the set of all bits $\mathbb{B}$ onto *feature families* $\mathbb{B}_{k,s}$, namely

$$\mathbb{B}_{k,s} = \{\mathrm{sgn}(\Re(a_{mk;s\ell})), \mathrm{sgn}(\Im(a_{mk;s\ell})) : m = 0,\ldots,M-1, \ell = 0,\ldots,R-1\}. \quad (9)$$

A single family collects all bits corresponding to the given frequency and scale indices, k and s, respectively. This second stage aims at searching the optimal frequencies and scales, and the previously determined set $\mathbb{B}^0$ is used to prioritize pairs (k,s) by their population of features from $\mathbb{B}^0$.

Following [2], we plot the number of elements in the set $\mathbb{B}_{k,s} \cap \mathbb{B}^0$ separately for real and imaginary parts of the Zak-Gabor coefficients, Figs. 5 and 6.

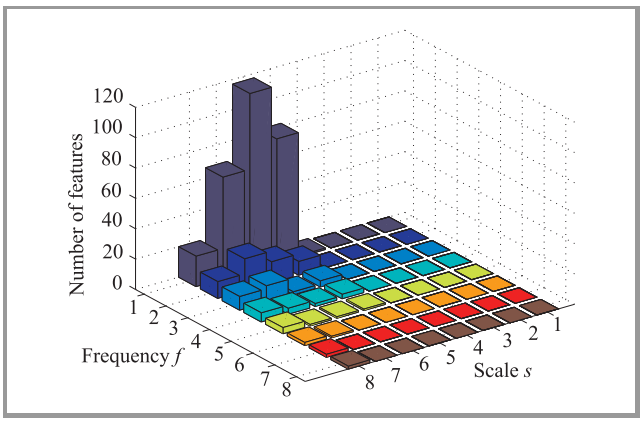


Fig. 5. A 2D histogram showing how families $\mathbb{B}_{k,s}$ are 'populated' by optimal features $\mathbb{B}^0$ determined for the imaginary part of Zak-Gabor coefficients.

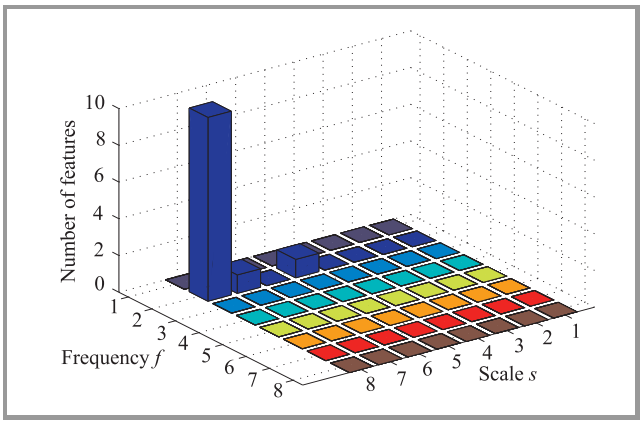


Fig. 6. Same as in Fig. 5 except that the real part of Zak-Gabor coefficients is used.

Note that the number of winning features is not identical for all families, which means that families differently contribute to the final discrimination capability of the resulting iris feature set. Thus the last step selecting the final set of families is required. Selecting the families of features may be done in various ways, and we use two variants of this selection. In the first variant we sort the families $\mathbb{B}_{k,s}$ by

the decreasing number of winning features $\mathbb{B}^0$ included in a given family, separately for real and imaginary parts of coefficients. This procedure prioritizes families that are most 'populated' by optimal features, and estimates each family's usefulness. Including a new feature family into the final code results in a new iris recognition system that may be, as in the first stage, assessed by calculating EER (or additionally d'), Fig. 7. Families characterized by the minimum EER may be thus chosen and deliver the final configuration of the Zak-Gabor-based coding in this application.

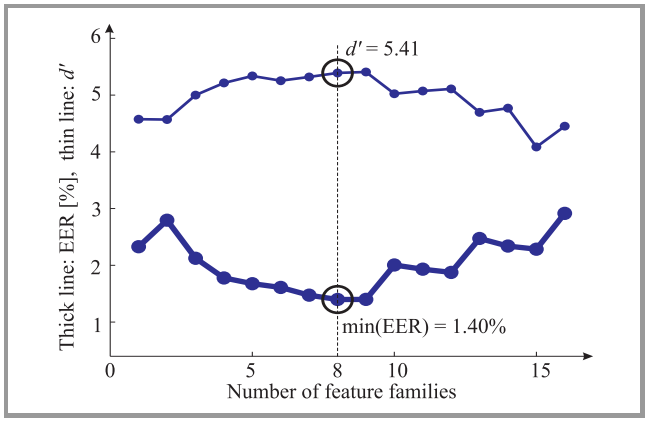


Fig. 7. EER (thick line) and decidability d' (thin line) versus the number of sorted iris feature families $\mathbb{B}_{k,s}$. Best solution minimizing the EER (achieved for the first 8 families) and the corresponding d' are marked by circles.

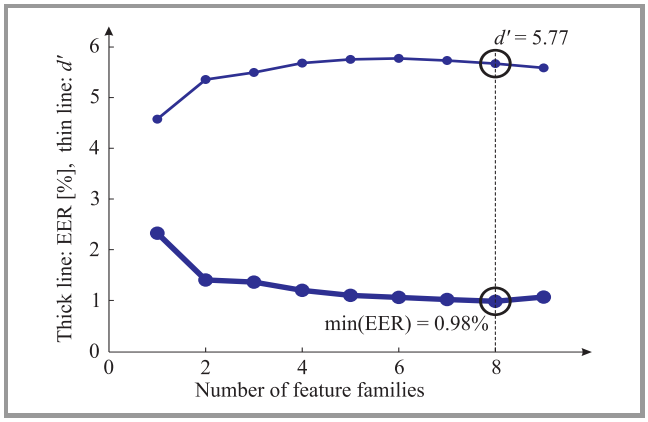


Fig. 8. Same as in Fig. 7 except that only families increasing the system's accuracy are iteratively added to the final set.

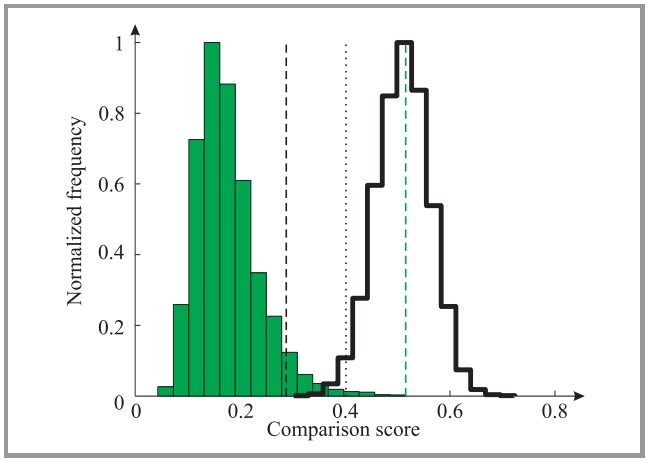


Fig. 9. Same as in Fig. 3, except that the optimal feature families $\mathbb{B}^*$ are employed.

However, one may observe that a few families increase the EER once they are added to the feature set. Thus the second variant of feature families selection adds only those $\mathbb{B}_{k,s}$ which increase system accuracy (i.e., decrease EER), Fig. 8. This variant results in best EER = 0.98% achieved for 8 families, constituting a set $\mathbb{B}^*$ of 576 elements (i.e., bits of the code). Figure 9 presents the distribution of genuine and impostor scores obtained for the winning variant.

4. Implementation of the Automatic Iris Feature Selection Tool

For the convenience and repeatability of calculations in the first stage of the iris feature selection, an automatic iris feature selection tool is introduced. It is a hybrid environment where Matlab scripts manage executable programs written in C/C++. This conjunction gives flexibility of script implementation and easiness in replacing any element of the tool, if needed. All the calculations related to iris image processing (segmentation, Zak-Gabor-based feature extraction, iris template creation and matching) is realized by ACIrisSDK libraries [6], integrated into the tool.

The tool contains two main non-volatile file sets. The first one is the Iris Data Base (IDB) for which optimal encoding parameters are calculated. The second one is called Support Repository (SR) where processed intermediate objects are stored. IDB has a simple internal structure where images are located in folders named by user identifier and an indicator of the left/right iris. Because of the huge amount of data loaded to the tool, not all operations and its variables might be stored in the RAM simultaneously. Each phase of automatic feature selection collects needed data from the SR, processes it and puts back the results there.

In the first phase, the proposed tool performs iris segmentation of images stored in the IDB. For each representation, the implementation of methods described in Subsection 2.1 provides the set of segmentation coordinates: centers of circles approximating inner and outer boundaries of the iris, detected occlusions and sectors chosen for features calculation. The results in a serialized form are placed as files in the SR.

Images stored within the IDB includes samples that contain distortions typical for iris image acquisition (e.g., blurred images, half-closed eye or incorrect cropping). To ensure the highest quality of the used data, a simple and heuristic algorithm of best samples selection is also build-in in the proposed tool. Namely, for each iris image two kinds of biometric templates are calculated: the enrollment template and the verification template. These templates are matched within the iris class. The results are collected

and these irises which are not matched to some (experimentally determined) number of other are rejected from further processing. This simple mechanism automatically finds outliers within a class, i.e., images of significantly worse quality than the remaining in the class. If necessary, other poor quality images, not rejected automatically, may be removed manually.

Third phase concerns calculation of a full Zak-Gabor-based feature set for each iris that was not rejected during previous stage. Before that, a very last distortion must be compensated. When the image of the eye is taken, there is a possibility of unintentional tilt of the camera or the object's head. The best way of fixing this problem is to shift the polar iris images in angular direction with a simultaneous checking of their correlation. The highest obtained correlation determines the angle for tilt correction of one iris in relation to the others. Then the configuration that minimizes the average tilt correction angle is selected, and all polar images are modified this way. During the next step, based on shifted polar images, the full Zak-Gabor-based feature set $\mathbb{B}$ is calculated for each iris and stored in the SR separately.

The final phase begins with aggregation of calculated feature sets into one matrix. First row of this matrix contains feature indices, and the remaining rows are composed with the iris class index, and values of its Zak-Gabor-based features. This matrix is an input for chosen mRMR method used for feature selection. The output is a set $\mathbb{B}^{\text{sorted}}$ of feature indices sorted by decreasing usefulness.

The proposed feature selection tool was previously used to estimate Zak-Gabor-based features (and coding parameters) for a few databases, e.g., publicly available Bath Iris Image Database, and our proprietary databases: BioBase collected by an IrisCUBE camera [2], and DatastripBase collected by a Datastrip DSV2+TURBO-SC mobile camera. The estimated time needed for feature selection for 1000 images is approximately 8 hours (assessed when running on the machine with Intel®Core i5 processor, equipped with 4 GB of RAM).

5. Implementation of the Secure Biometric Verification Station

Secure biometric verification station (SBS) is designed to be a part of a larger system [9], consisting of one or more external PC units, which send requests of user's identity biometric verification. In the following subsections, we present the hardware and software components, respectively.

5.1. Hardware Specification

For the purpose of the station development we used a conjugation of two ready-to-use devices. The first one, *Kontron Micro Client IIA 70* is a fanless microcomputer equipped with the Intel®Atom™N270 1.6 GHz CPU, 2 GB of RAM and 8 GB of a Compact Flash memory, a 7.0" TFT LCD touch screen, USB and LAN 10/100/1000 interfaces and Kontron customized Windows XP Embedded. This configuration provides a compact and fully functional environment for Windows based x86 applications.

Fig. 10. Components of the developed station: a processing unit with a display for communication with a subject and an iris capture camera.

Second device, *Corvus Vista FA2*, is a face and iris capture camera. Connected to the Kontron microcomputer with USB 2.0 interface guarantees fully automated capturing of iris images compliant with ISO/IEC requirements [4] at the resolution of 640×480 pixels. The iris camera is equipped with multi-wavelength IR illuminants, a distance-sensed auto focusing system and the LED-based feedback for captured subjects convenience. Provided SDK for C/C++ enables also gaining RGB face images (up to 2048×1536 pixels) and setting basic illumination adjustments for both images acquisition modes. Figure 10 shows hardware components of the station (intentionally presented without casing).

5.2. Software Functional Requirements and Implementation

Designing the secure biometric verification station induces a few security, comfort and simplification aspects to be considered. In particular they concern:

- secure protocol of communication with an external unit,
- clear and understandable Graphical User Interface,
- proper iris image acquisition,
- internal data organization system,
- handling of biometric processes.

The implementation of the proposed station functionality is divided into individual components, Fig. 11. They are designed as follows:

- **SBS** – exports three main biometric functions (`SBS_Enroll`, `SBS_Verify`, `SBS_Delete`) in a form of Dynamic-Link Library (DLL); SBS integrates also the remaining components;

– **GUI** – handles the Graphical User Interface;

– **FILE SYSTEM** – contains necessary files (GUI items, users biometric templates and log files) in a specially designed folder structure;

– **BIOMETRICS** – provides biometric operations with the use of ACIrisSDK libraries, including creating and matching of user templates stored in the OS file system;

– **CORVUS VISTA FA**2 – provides communication and control over the iris camera used.

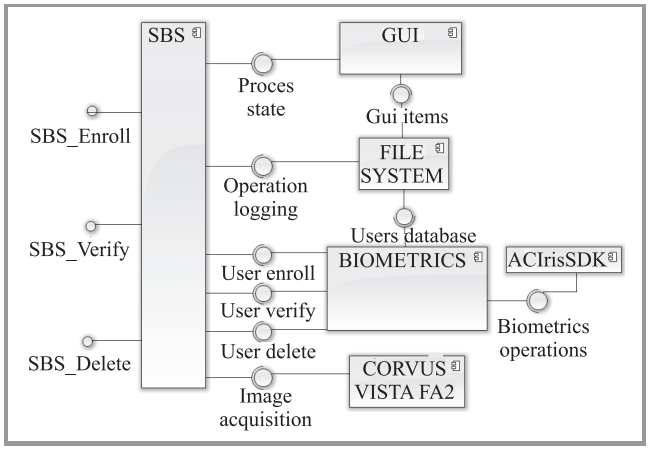


Fig. 11. Component diagram of the SBS software modules.

Communication protocol. Cooperation with any other PC unit (even in the immediate vicinity) requires establishing a secure communication between these two machines. If a communication channel is not appropriately protected, any biometric data transmitted may be stolen or modified. To prevent the system from abuses, special communication protocol is developed.
For flexibility in developing of the proposed solution, a communication layer is separated from the verification procedure, in particular from handling biometric devices, controlling acquisition processes and data processing (templates creation and matching), as well as database and errors logging. To achieve this goal, all non-communication functionality is supplied in C/C++ Dynamic-Link Library (DLL) which provides three general functionalities: new user enrollment, user verification, and deleting users data from the system.

Graphical User Interface with Qt. Neither enrolling to the system nor verifying the user is an atomic process. Both processes consist of several stages where each of them may take some time and result in an error or warning. Therefore, for making a proposed solution more user friendly, we also introduce Graphical User Interface (GUI) implemented in Qt 4.7 technology, and fully compatible with the provided 7" LCD touch screen of the microcomputer. The Information about the type of current process, its result and commands may be displayed in a clear form to the user. In the presented solution, there is no need for a user to input any information using the screen, mouse or keyboard, besides presenting the iris after the message. This is due to controlling current biometric processes by the external unit.

Iris image acquisition. Prepared solution contains necessary functionality (based on the Corvus Vista FA2 vendor's SDK) for ensuring proper device initialization, handling of image capture timeouts, illumination adjustments and terminating of the connection with the camera, after receiving captured iris images. The procedure of iris capture is fully automatic. Provided SDK uses build-in distance sensor for estimating face position in front of the lens, it controls the NIR illuminants for appropriate scene illumination and corrects sharpness of the image. The quality of the iris image is assessed on-line, and when it is acceptable the capture process terminates, sensor and illuminants are set off, and the image is available in the indicated memory buffer.

Users database and log files. In order to be simple, biometric templates, logging results and selected GUI elements are stored in a local nonvolatile memory. There is no need for installing detached Database Management System (DBMS) (like *mySQL* or *Microsoft SQL Server*) as the user database structure is defined by biometric templates, serialized to binary files placed in the `USERS` local folder, and the configuration data is kept within the `CONF` local folder. Users can be added and removed only upon the external unit secured request. The station implements a simple logging mechanism. The results of component functions along with threat level (one of six possible), time stamp (set with thousandth of a second precision), associated user ID information and a message with a description are appended to a text file (named by daily date) in the `LOG` folder.

Biometric operations. The main software component of the station is responsible for biometric operations, i.e., enrolling new users and verifying them upon the request of the external unit. The enrollment process starts with checking of the possibility of adding a new user (specified by UID). It may not be possible when the maximum number of registered users has been reached or indicated that the UID already exists. In the latter case, the request is revoked and no further action is taken. Otherwise, three enrollment iris images are captured (each after a specified time interval), the biometric template is created, and it is placed under provided UID in the user database. If timeout occurs during the acquisition of images, but at least one image was captured, template can still be created with the obtained samples. The enrollment process is also revoked when no image is captured.

At the verification stage, only one image on the eye is captured, and temporarily created verification template is matched with the enrollment template stored in the user database (if UID provided is registered within the database).

Verification result is displayed on the LCD ('MATCH / NO MATCH') and it is sent to the external unit requesting the verification.
The enrollment process takes typically one minute (including capturing of three iris images, template creation and its storage) and the verification process does not exceed a second, which very favorably compares to the most of commercial iris recognition systems.

6. Summary

This paper describes an application of the well-established Zak-Gabor-based iris coding to build a secure verification station. To adapt the coding parameters (i.e., iris feature families, corresponding to frequencies and scales of wavelets emphasizing individual iris features) we used the mRMR method using the mutual information as an indicator of the iris feature usefulness. To make the selection process an automatic one, the iris feature selection tool was designed and built. A database of iris images collected by the developed station was used to automatically adapt the iris coding to the quality of iris images employed. The feature selection tool allowed for convenient adaptation of the Zak-Gabor-based method parameters (in a reasonable time of several hours) and a promising EER = 0.98% was achieved for iris images collected by the designed verification station.

Acknowledgements

This work was partially funded by a grant number OR00014011 (project entitled "Secure workstation for special applications") from the National Center for Research and Development, within science funding program for years 2010–2012.

References

[1] A. Czajka and A. Pacut, "Zak's transform for automatic identity verification", in *Proc. 4th Int. Conf. Rec. Adv. Soft Comput. RASC 2002*, Nottingham, United Kingdom, 2002, pp. 374–379.

[2] A. Czajka and A. Pacut, "Iris recognition system based on Zak-Gabor wavelet packets", *J. Telecom. Inform. Technol.*, no. 4, pp. 10–18, 2010.

[3] J. Daugman, "The importance of being random: statistical principles of iris recognition", *Pattern Recognition*, vol. 36, pp. 279–291, 2003.

[4] "Information Technology – Biometric data interchange formats – Iris image data", ISO/IEC FDIS 19794-6:2011, 2011.

[5] M. J. Bastiaans, "Gaborñs expansion and the Zak transform for continuous-time and discrete-time signals", in *Signal and Image Representation in Combined Spaces*, J. Zeevi and R. Coifman, Eds. Academic Press, 1995, pp. 1–43.

[6] A. Czajka and A. Pacut, "SDK for iris recognition", *NASK Review*, pp. 34–39, 2009.

[7] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 27, no. 8, 2005.

[8] H. Peng, "Mutual Information computation", MATLAB implementation, August 23, 2007 [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/14888

[9] A. Kozakiewicz, A. Felkner, J. Furtak, Z. Zieliński, M. Brudka, and M. Małowidzki, "Secure workstation for special applications", in *Secure and Trust Computing, Data Management, and Applications*, C. Lee, J.-M. Seigneur, J. J. Park, R. R. Wagner, Eds., Communications in Computer and Information Science, vol. 187. Berlin: Springer, 2011, pp. 174-181.

Adam Czajka Adam Czajka received his M.Sc. in Computer Control Systems in 2000 and Ph.D. in Control and Robotics in 2005, both from Warsaw University of Technology, Poland, with honors. He is an Assistant Professor at the Warsaw University of Technology (2003–) and at the Research and Academic Computer Network NASK (2002–). V-ce Chair of the NASK Biometric Laboratories, member of the NASK Research Council (2006–). Expert of ISO/IEC JTC1 SC37 on Biometrics (2011–), NASK representative in TC on Biometrics (2009–) and on Information Security in IT Systems (2007–) of Polish Normalization Committee (PKN). Head of postgraduate studies on 'Security of IT Systems and Biometrics' at the Warsaw University of Technology (2011–). Member of the IEEE (Institute of Electrical and Electronics Engineers, Inc., 2002–) and the EAB (European Association for Biometrics, 2012–).
E-mail: Adam.Czajka@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Krzysztof Piech received his M.Sc. in 2012 from the Faculty of Electronics and Information Technology of the Warsaw University of Technology, Poland. Since 2010 he works at Biometric Laboratories of Research and Academic Computer Network NASK. Currently he is applying for Ph.D. studies. He is interested in biometrics, image processing, security systems and related areas.
E-mail: Krzysztof.Piech@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Paper

BSBI – a Simple Protocol for Remote Verification of Identity

Adam Kozakiewicz and Tomasz Pałka

Research and Academic Computer Network (NASK), Warsaw, Poland

Abstract—The paper presents the design and the rationale behind a simple verification protocol for autonomous verification modules, and the architecture enabling use of such modules. The architecture assumes strict separation of all personal metadata and the actual verification data. The paper also describes a prototype implementation of the protocol and its extension enabling the state of the module to be monitored from the main system. The proposed design solves the problem of using advanced verification methods, especially biometric ones, in systems where direct implementation is not possible due to hardware incompatibilities, insufficient resources or other limitations.

Keywords—access control, authentication, biometric verification, network protocols.

1. Introduction

The security of any computer system depends to a large extent on the proper verification of the identity of its user. The access control policy used in the system is meaningless if the user is misidentified and allowed to work as someone else, especially if the user is not authorized to use the system at all. In most systems, this verification of identity is very simple, usually requiring the knowledge of a password. Naturally, systems with higher security requirements should employ more secure methods of verification. The password protection is actually quite good, there is no reason to eliminate it, but it should be accompanied by other methods, preferably employing hardware identification devices (tokens, cards, etc.) and/or biometric measurements.

In this paper we are dealing with the problem of applying advanced biometric verification methods in a system, which – for reasons stated in the next section – cannot implement them directly. We introduce a separate module, called BSB, responsible for performing the verification. We present the architecture of such a solution and the protocol used to connect the host system with the verification station.

Section 2 shows the background of this research, explaining the reasons why a separate verification module is needed in our solution. The section also presents a simple architecture which satisfies our requirements. As the architecture requires a special protocol for communication between the BSB and the host, we review the existing solutions in Section 3 and – having explained why none of them fit our needs – describe the protocol we designed, called BSBI, in Section 4. Section 5 describes our prototype implementation and Section 6 explains how it could be perfected in the future. We conclude with a short summary in Section 7.

2. Background, Assumptions and Design

The secure workstation for special applications [1] is a Linux-based system using visualization to process data from different security domains in separate environments. The guest systems can be either Linux- or Windows-based, but the main point of access control is the host system, based on Linux (specifically the project uses Red Hat Enterprise Linux). The task of verifying the identity of the user is performed at the host level. The built-in capabilities, including password-based verification are available. However, the system's high security level requires more advanced access control, preferably using several different methods in parallel.

Among the project goals is the demonstration of different authentication mechanisms [2], including hardware-based verification (e.g., token or card [3], [4]) and biometric verification [5], [6]. Specifically, we chose to implement iris recognition [7], [8]. This biometric modality is relatively reliable from the technical point of view, guarantees low false negatives and – more importantly – false positives.

The combination of iris recognition and Linux host operating system is unfortunately a significant compatibility problem. The commercially available specialized cameras for iris recognition rarely have drivers for Linux systems. Even the few that do, do not offer full functionality with those drivers. Furthermore, iris recognition requires quite advanced numerical analysis, available in the ACIrisSDK library developed at NASK [7], [8]. However, due to the driver issues described above, that library was written for the Windows system and porting it is not an easy task. Implementing iris recognition on the host system level would therefore be prohibitively difficult.

The difficulty is not the only reason against a host-level implementation. For security reasons, the host operating system should be minimal and based on well-tested software. The biometric processing is quite complex and works with externally provided data (photos of the iris). This makes it a potential weak spot in the system, especially since – as stated before – it would have to be new, implemented from scratch or ported in a non-trivial way. With no prior production use, even after rigorous tests this implementation would be likely to contain numerous bugs, some of which may be exploitable security vulnerabilities.

This situation left the designers with only two possible choices. The iris recognition would have to be performed on a Windows-based system, separate from the host. That system could be a virtualized guest or an external module. Since the workstation is basically designed as a minimal, secure virtualization environment, a hypervisor is already present on the host and the virtualization approach may seem sufficient. In fact, this is illusory. Implementation as a virtual machine would have several important consequences for the whole workstation, quite contradictory to the projects goals. The physical separation of virtual machines, postulated in the project, would require a whole physical CPU to be reserved for the biometric virtual machine[1]. A computer with two physical processors would then not be enough to run two protected, user-accessible virtual machines in parallel, as required. The alternative, deactivating the biometric VM when not necessary, would solve the issue, but greatly increase the verification delay.

The requirements of the biometric machine are also problematic. It would require full access to the camera, most likely connected via USB, and two-way communication with the host. The other virtual machines are not allowed to have any two-way communication with the host and their access to USB ports is heavily controlled (only special, cryptographically protected USB drives are allowed). Adding exceptions increases the probability of introducing vulnerabilities.

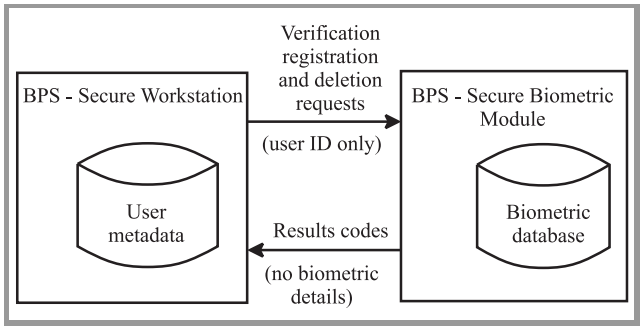


Fig. 1. The general architecture of the system. Note that the two types of data are kept separate and linked only with a numeric identifier.

For the above reasons, the final decision seems clear – the iris recognition will be performed on a separate physical machine. The system will consist of two separate modules, as shown in Fig. 1: the secure workstation and the verification station (called BSB). The authentication tasks are clearly divided between the two modules – every action is only performed on one of the stations, e.g., the password-based authentication is performed by the host station without any interaction with the BSB, while biometric authentication is performed entirely by the BSB. The BSB is autonomous in its operations. The biometric database and the biometric software are placed only on the BSB, no such details are passed to the host station. Any necessary hardware must therefore be also connected to the BSB. The only information about the user passed between the two modules is a numeric identifier. The BSB is equipped with its own screen and does not need the host system as a proxy for user communication.

The architecture described in the previous paragraph is inspired by the way in which credit card terminals are often integrated with ticket vending machines, etc. The terminal is only activated on request and is autonomous. The machine instructs the user to follow the instructions displayed by the card terminal and waits for a confirmation from the terminal. This makes the integration very easy for the designers of the vending machine.

The construction of the BSB itself is a separate issue, outside the scope of this paper. If the autonomous nature of the BSB is preserved, then the issue is not at all important for the host system. The integration would be done exactly the same way if the verification used a completely different biometric modality or hardware ID. However, this elasticity depends on the flexibility of the communication protocol connecting both modules.

3. Related Work

There is a multitude of existing authentication protocols. In theory, using them would be the simplest and best way to meet the requirements of the project. However, the existing protocols are in fact designed for a completely different task. They are usually used as scalability enhancers for distributed systems, delegating the task of user authentication, authorization and accounting to a central server. This approach is used in protocols like RADIUS [9] or Diameter [10]. Also Kerberos [11], [12] uses a central server approach, focusing on a cryptographic ticket mechanism to provide single logon. All of these protocols are powerful tools enabling effective management of large networks. However, our needs are much simpler. The authentication, assignment of rights, etc., are all performed on the host system. However, the verification tasks delegated to the BSB are handled there completely.

The TACACS+ [13] protocol and its predecessors (TACACS, XTACACS) take a different approach. They do provide the means to perform remote interactive authentication, separated logically from authorization. The protocol handles password-based authentication very well and can be adapted to any other similar authentication method, e.g., hardware tokens. Adapting it to use biometric verification is a lot more difficult, but probably possible. However, the assumptions of the TACACS family of protocols are a reversed version of our design. The authentication server performs the actual authentication, but the user interaction is done on the terminal side. In our setting the entire verification process is performed on the BSB side. The host system simply requests the authorization to be performed and awaits results. This makes TACACS+ a suboptimal

[1]This is a simplification, multiple VMs may in fact use the same CPU if and *only* if they belong to the same security domain.

choice for our system. Taking into account the minimal amount of necessary communication, the decision was to develop a specialized protocol.
Since biometric verification was our main goal in this project, it was also possible to use an existing biometric protocol. Such a protocol exists – the standard biometric API called BioAPI [14] can be mapped into network messages using ASN.1, resulting in BIP – BioAPI Internetworking Protocol [15]. However, from our point of view this protocol is very low-level, giving access to many details of the biometric processing. It would be useful, if the user database was placed on the host machine, but since all biometric activity is limited to the BSB, introducing such low level information in the protocol only serves to make it less universal. A simpler protocol could easily be used with different configurations of the BSB, using different biometric modalities, or even non-biometric methods.

4. Communication Protocol

The design of a communication protocol for identity verification is relatively simple in this setting – the required set of operations is very small. In fact, the minimal set would include only two operations: adding a user by registering the necessary data given the user's identifier or verification of the identity of the user given the identifier of the user he claims to be. For practical reasons, a third operation is quite useful – deleting the user with the given identifier, removing all data collected for him.
The basic design requirements, apart from providing the specified minimal set of operations, were as follows:

1. The design must be based on well known, standard solutions.
2. The autonomous nature of BSB, as specified in the proposed architecture, must be preserved.
3. The messages should be limited to ASCII characters and not excessively long, so that the protocol can be used unmodified on any link.
4. The messages should be human readable when unencrypted (useful in implementation and testing phase).
5. The protocol should be easy to implement.
6. The protocol should be easily extendable.

The requirements and the natural request-response nature of communication with BSB resulted in a solution based on Web Services. The choice between SOAP and XML-RPC was also clear, as SOAP is unnecessarily complex – we have no use for its advanced capabilities and the large envelope directly contradicts the requirements. Since ASCII is a subset of UTF-8, it is easy to satisfy requirement 3 using XML entities and BASE64 encoding where necessary. Also, the clear hierarchical structure of XML documents satisfies requirement 6.

4.1. Extended Functionality for More Advanced BSBs

While the three basic operations are sufficient for the verification module, the actual protocol provides two additional, optional features – capabilities and log collection.
Capabilities, although simple, are a powerful extension, allowing more advanced verification modules with multiple verification methods to be designed and used without further modifications to the protocol. A capability is simply a name of a verification method. The protocol allows the host to obtain a list of capabilities supported by the verification module and to specify the desired set of methods during verification or registration of a user's identity. The verification module is also required to define internally a default set of verification mechanisms, allowing it to be used with a host which does not support capabilities.
Capabilities are described in the format `<capabilityname>_<mode>`. The `<capabilityname>` part identifies the general method of verification (e.g., biometric modality) and should be unified. The BSBI protocol provides names for most popular verification methods:

- **CHIP** Chip ID card or other intelligent hardware ID inserted into a reader,
- **FACEGM** Face geometry,
- **FPRINT** Fingerprint,
- **FVEIN** Vein pattern in a finger,
- **HANDGM** Hand geometry,
- **IRIS** Iris,
- **PASS** Password or PIN code,
- **RCHIP** Wireless chip card or other intelligent wireless hardware ID,
- **RETINA** Retina,
- **RFID** RFID-based ID card or other simple wireless hardware ID,
- **SIGN** Written signature,
- **SPASS** One-time-password,
- **SWIPE** Magnetic ID card or other simple hardware ID inserted into a reader,
- **TOKEN** Hardware token,
- **VOICE** Voice.

The above list is by no means complete. Additions to it should be done in a consistent manner.
The `<mode>` part is not strictly defined – it may be any string (without whitespace) specifying a variant of the method, preferably in a clear and descriptive way. Short mode names are preferred. If only one variant is imple-

mented for a given capability, it is allowed to use only the capability name as its identifier, but the preferred name uses mode `default`. This mode is not reserved for that use, so it is, for example, perfectly acceptable to offer capabilities `SIGN_default`, `SIGN_press` and `SIGN_nopress`, where the latter two modes define whether pressure measurements should be taken into account and the `default` mode is synonymous to one of the other variants. Synonyms are not treated in any special way, from the point of view of the protocol all modes are different.
The log collection is simply a way to access more detailed information about the operations of the verification module. The host system may use it to obtain a copy of some of the module's logs. The proposed specification assumes that the logs are treated as a single stream of entries by the host, delegating the selection of entries to the verification module. Better granularity may be provided by extending the protocol, if necessary.

4.2. Specification of Operations

The protocol consists of five methods – one for each of the elementary operations of the BSB, one listing the capabilities of the BSB and one used to collect the logs. The methods generally report their execution status using numeric error codes, which are defined as follows:

- 0 `BSBI_SUCCESS` – successful completion;
- 99 `BSBI_UNKNOWN` – unknown error;
- 100 `BSBI_USER_NEXIST` – the given user ID is not registered in BSB database;
- 101 `BSBI_USER_EXIST` – the given user ID is already registered in BSB database;
- 110 `BSBI_USER_LIMIT` – number of registered users exceeds a preset limit;
- 200 `BSBI_PROC_TIMEOUT` – processing timeout in verification module;
- 210 `BSBI_PROC_INTERNAL` – internal error in verification module;
- 299 `BSBI_PROC_UNKNOWN` – unknown error in verification module;
- 300 `BSBI_CAPA_NSUPPORT` – requested capability not available in the BSB;
- 301 `BSBI_CAPA_NCOLLECT` – data required by the requested capability not collected;
- 310 `BSBI_CAPA_NODATA` – no user template for the requested capability for the specified user.

Additionally the following library-level error codes (900–999) are defined. These are never sent as part of the protocol and are reserved for use by libraries implementing BSBI:

- 900 `BSBI_CONN_FAIL` – cannot establish connection to BSB (connection refused or other problem);
- 901 `BSBI_CONN_TIMEOUT` – timeout while waiting for BSB response;
- 910 `BSBI_CONN_PARSE` – bogus response from BSB;
- 920 `BSBI_CONN_FATAL` – BSB or link security breach suspected (e.g. invalid BSB certificate);
- 930 `BSBI_CONN_INTERNAL` – BSBI library internal error;
- 940 `BSBI_CONN_PROTO` – protocol incompatibility – BSB does not support requested functionality;
- 999 `BSBI_CONN_UNKNOWN` – unknown connection error.

The methods of the protocol are defined as follows:

enroll.user(uid, *capa*) – the method registers a new user with the supplied ID. The BSB is expected to collect and store autonomously all necessary information, e.g., biometric data.
The mandatory parameter `uid` contains the numeric ID of the user.
The optional parameter `capa` may contain a list of verification methods the host system intends to use in the future. The BSB is required to collect and store all necessary data for these verification methods, failure to collect any of them must cause the entire operation to fail. It may also collect data for other variants, but failure to obtain those must not be considered an error. Specifying a capability not supported by the BSB is an error and user enrollment must fail in this case. If the list contains a capability specified only by capability name (omitted mode part of the name), the BSB is free to use any of the available modes of that capability.
If the parameter `capa` is not provided, the BSB is free to choose which kinds of data should be collected. As a minimum, the BSB should collect data for all capabilities used by default (that is when the `capa` parameter is omitted) by the `verify.user` method – failure to collect those may and should be considered a failure to register the user. The preferred action is to collect data for all installed capabilities, but tolerate failure to obtain data for verification methods not used by default.
The method returns a single value `code`, which is a numeric error code. The following error codes are possible in this method:

- `BSBI_SUCCESS`,
- `BSBI_UNKNOWN`,
- `BSBI_USER_EXIST`,

– BSBI_USER_LIMIT,

– BSBI_PROC_TIMEOUT,

– BSBI_PROC_INTERNAL,

– BSBI_PROC_UNKNOWN,

– BSBI_CAPA_NSUPPORT (only if the capa parameter was provided),

– BSBI_CAPA_NCOLLECT.

verify.user(uid, *capa*) – the method verifies the identity of a user, based on the supplied ID. The BSB is expected to collect autonomously the necessary data and verify its correctness for that ID using templates created by the enroll.user method.
The mandatory parameter uid contains the numeric ID of the user.
The optional parameter capa may contain a list of verification methods that should be used. The verification is successful if and only if all of the listed capabilities were successfully used and confirmed the user's identity. The BSB may not use any capabilities not in the list. Specifying a capability not supported by the BSB is an error. If the list contains a capability specified only by capability name (omitted mode part of the name), the BSB is free to choose one or more of the available modes for which data was collected during enrollment of the user being verified. If user data is not available for any mode of the capability, then the verification obviously fails.
If the parameter capa is not provided, the BSB is free to choose which capabilities to use for verification and how to proceed if one of them fails. The choice should be limited to the capabilities used during enrollment of the user being verified.
The method returns one or two values. The first value is code, a numeric error code. The following error codes are possible in this method:

– BSBI_SUCCESS,

– BSBI_UNKNOWN,

– BSBI_USER_NEXIST,

– BSBI_PROC_TIMEOUT,

– BSBI_PROC_INTERNAL,

– BSBI_PROC_UNKNOWN,

– BSBI_CAPA_NSUPPORT (only if the capa parameter was provided),

– BSBI_CAPA_NODATA.

The second value, also numeric, called verificationResult, is provided if and only if the value of code is BSBI_SUCCESS and is either 0 if the user's identity was confirmed, or 1 if the user's identity was rejected.

delete.user(uid) – the method removes from the BSB all data related to a user identified by the supplied ID.
The mandatory parameter uid contains the numeric ID of the user.
The method returns a single value code, which is a numeric error code. The following error codes are possible in this method:

– BSBI_SUCCESS,

– BSBI_UNKNOWN,

– BSBI_USER_NEXIST,

– BSBI_PROC_UNKNOWN.

list.capa() – the method retrieves a list of all capabilities supported by the BSB.
The method does not require any parameters and returns an array of character strings called capa. There is no code value in this case – any non-empty response should be regarded as confirming the result BSBI_SUCCESS, while an empty response should be interpreted as BSBI_UNKNOWN.

get.logs(*zipmode*) – the method retrieves new log entries from the BSB. Selection of the entries depends on the configuration of BSB, the protocol does not provide any method of controlling it.
The optional parameter zipmode specifies the compression method which should be applied to the logs. It is a string from a well defined set of values. Currently, the defined values are ZIP, GZ and NONE, new ones may be added in the future. The values are not case sensitive. The BSB must use the specified compression method, unless it does not support it – in that case, no compression should be used (as if the value of zipmode was NONE).
The method returns one or four values. The first value is code, which is a numeric error code. The following error codes are possible in this method:

– BSBI_SUCCESS,

– BSBI_UNKNOWN,

– BSBI_PROC_INTERNAL (reserved for errors generated when parsing BSB's logs),

– BSBI_PROC_UNKNOWN.

The other three values are returned if and only if the value of code is BSBI_SUCCESS.
The value logs contains the log entries in BASE64 encoding. If the zipmode parameter was specified and not NONE, the entries may be compressed before BASE64 encoding is applied. There is a limit set on the length of this value – it may not exceed 32 kilobytes after BASE64 encoding.
The value zipmode defines whether compression has been used and is either a copy of the input parameter with the same name, or NONE if the specified compression method is not supported.

Finally, the value `hasnext` is a boolean value, true if and only if not all available entries fit under the 32 kilobyte limit. The host system should then immediately call this method again to retrieve the missing lines.

5. Prototype

The prototype implementation (as shown in Fig. 2) was completed in the C language, using a preexisting XML-RPC library. The BSBI was implemented as a portable library, easily adaptable to new uses.

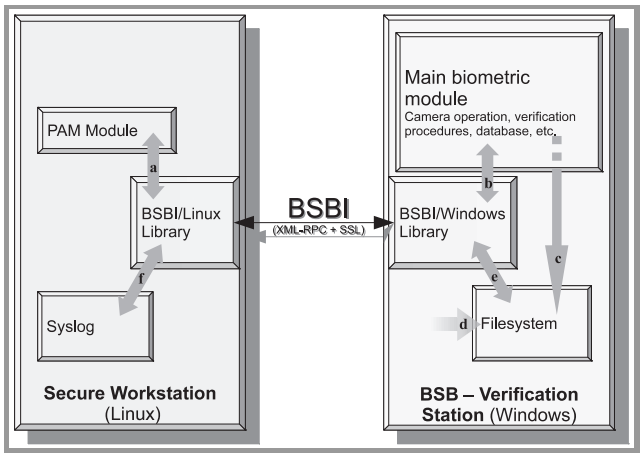


Fig. 2. The structure of the prototype implementation of the BSB: (a) verification requests are handled by a special PAM module using the BSBI library; (b) communication with the biometric module uses a simple wrapper library; (c),(d),(e),(f) events from the biometric module (c) and system-level events (d) are logged into files, which are (e) monitored by a special module of the BSBI library and (f) selected entries are reported to the host station's syslog.

The BSB has been equipped with Apache HTTP server configured to enable SSL-protected connections from the BSB. The connection requires authentication with a registered certificate by both the client and the server, ensuring a secure pairing of the BSB with the host system. The BSBI module on the server handles BSBI calls by translating them to calls to functions of a specially designed thin wrapper around the ACIrisSDK library, responsible for the actual biometric processing. The module also collects results of a separate process which parses logs generated on the BSB and extracts the entries, which will be sent to the host system on the next call to `get.logs`.

On the client side, the library is used to generate calls to BSB. It could potentially be used by regular applications requiring biometric verification of identity, but in the designed system it is reserved for use in the initial user authentication process.

Given that the host operating system in the project is Red Hat Enterprise Linux, the natural way to include the BSB in the user authentication process is to implement a module for the PAM (Pluggable Authentication Modules) system. The module is then included in PAM configuration as the second step after standard password authentication. Such a module, called `pam_biometric.so`, has been implemented. However, the most obvious authentication policy, strictly requiring successful biometric verification, is not acceptable in practice. The design of the secure workstation prohibits any external modifications (e.g., after booting from a live CD), the only way to change its configuration is to authenticate successfully as an administrator. This means that any failure of the BSB makes the host system completely unusable – even replacing the BSB is not possible, unless the original private key can be recovered.

For this reason the PAM configuration is a bit more complex. Another simple module for the PAM system was developed, called `pam_spec.so`. This module performs a simple password-based verification. However, the module only accepts users belonging to a special group `spec`, and the password is never used normally. It is also stored in a separate file, `/etc/spec_shadow`, protected using SELinux. The approach is quite secure, especially if password and BSB are not the only authentication modules used by the workstation.

The final configuration can be as follows:

```
#%PAM-1.0
auth     required     pam_unix.so nullok ↵
    try_first_pass
auth     [success=done authinfo_unavail=ignore ↵
    auth_err=die default=die]  pam_biometric.so ↵
    audit
auth     requisite    pam_succeed_if.so user ↵
    ingroup spec
auth     required     pam_spec.so audit
```

First, a successful password-based authentication is required. If it succeeds, the identity of the user is already tentatively established and a BSB-based authentication can be attempted using his identifier. As the second line specifies, successful BSB authentication is sufficient and ends the process. Rejection by BSB also ends the process – authentication fails. However, if the module reports that it is unavailable, i.e., the BSB cannot be contacted, then processing continues. The third line rejects any users not in group `spec` and the fourth line attempts special password authentication as the final, decisive step. The `audit` option used in both modules implemented in the project turns on detailed logging for audit purposes and can be omitted if not necessary.

The integration package includes both PAM modules, the BSBI library and some helper programs. The helper programs enable pairing of the host system with the BSB, testing the connection, enrollment and removal of users, setting the special passwords, etc.

6. Possible Extensions

While the BSB + BSBI verification mechanism is implemented and working well, it is still only a prototype. A fully mature commercial system would require further extensions and modifications. Some of the extensions described in this section are planned as future work, others are just proposals which may or may not be considered for implementation in the future.

Most importantly, to be used as part of a secure workstation, especially one certified for processing of classified information, the BSB would have to adhere to the same stringent security requirements as the workstation itself. Several steps in that direction were already made. The BSB, once configured, does not have any connected human interface devices and the touchscreen built into the computer used in the prototype is used only as an output device. Nevertheless, the casing is a purely temporary solution, not acceptable for production use, as it does not offer any physical security. All ports of the computer are relatively easy to access. In a final implementation the computer and the camera would be installed in a locked casing with secure, well placed ventilation holes, so that accessing any ports would require a key (it is normally only necessary during installation and initial pairing with the host). Protection of the link is also important, although as long as the BSBI communication is cryptographically protected this may not be crucial. Also note that the BSB is not really designed for use in the field – processing of classified information typically takes place in rooms providing significant physical security and well documented access control. In other applications the BSB would likely not be a separate piece of equipment. Whether wall-mounted or built into a larger device, it would probably be sufficiently protected. Even now, physical integration with the host workstation is perfectly possible if the workstation's casing is large enough. However, in the envisioned application a smaller, standalone verification device connected to the host with a cable seems much more usable as it offers a lot more flexibility in placing the camera so that using it would not require leaving the chair.

The security requirements, especially at higher levels, may require replacing SSL-based encryption and symmetric authentication with encryption hardware. This is however easy to do and does not require any changes in the BSBI protocol.

Another security-related shortcoming of the current solution is the relatively low security of the BSB itself. As long as there are no input devices connected to it (apart from the camera, of course), this is not a serious problem. However, the computer used in the BSB is small enough to carry in an average bag. The consequences of stealing the BSB or making a copy of its data are hard to define. The BSB does not have any useful metadata associated with the biometric (or other) verification data stored on it. The identifier passed to it by BSBI is not easily identifiable. However, the data may be quite useful anyway, especially if the set of registered users is sufficiently small. The connection between a person and its verification data can be recovered in several ways, e.g., by analysing the logs present on the BSB and comparing them with observations of the times when individual users accessed the workstation. The verification data, especially biometric templates, should definitely be considered sensitive.

Another interesting piece of information on the BSB is its private key, used to pair with the host. Having this certificate, it is generally possible to develop a fake BSB, which will pair correctly with the host, but will, e.g., always reply to `verify.user` calls with a positive verification. It may be possible to use the Trusted Platform Module (TPM) of the BSB to protect the key. This method, however, will not suffice to protect the verification data.

One method worth trying would be to put the biometric software along with the database on an encrypted volume. The key necessary to access that volume would be stored on external storage during the setup phase, and afterwards it could be put on the host system, preferably encrypted using the host's TPM. The BSBI protocol would then require two extensions. One new error code, called, e.g. `BSBI_NOT_READY` would be used to inform the host in reply to any call that the BSB's verification modules are not yet running. Then, a separate method of the BSBI protocol (e.g. `init`) would be used to send the necessary key to the BSB, which would then be able to start the verification services. This way the sensitive data could not be accessed without connection to the right host.

Another possible extension would be to enrich the `get.logs` method with the possibility to specify the requested type of logs, or even to implement two-way communication, where the logs are simply pushed to the host machine. The latter variant is easiest to do using syslog instead of extending BSBI.

7. Conclusions

The architecture and protocol described in this paper have been implemented and tested as part of the project "Secure workstation for special applications", but the solution is more general, not limited to this one application. We have shown that the approach borrowed from credit card terminals – making the verification station a separate, autonomous module and developing the main system as agnostic of the inner workings of that module – is indeed workable for this application. The approach may be used whenever advanced identity verification is necessary in a system in which such solutions are not readily available or may not be possible to implement due to, e.g., lack of sufficient computing power.

Acknowledgements

This work is part of the project called "Secure workstation for special applications" and is funded by a grant number OR00014011 from the National Center for Research and Development – science funding for years 2010–2012.

References

[1] A. Kozakiewicz, A. Felkner, J. Furtak, Z. Zieliński, M. Brudka, and M. Małowidzki, "Secure workstation for special applications", in *Secure and Trust Computing, Data Management, and Applications*, C. Lee, J.-M. Seigneur, J. J. Park, R. R. Wagner, Eds., Communications in Computer and Information Science, vol. 187. Berlin: Springer, 2011, pp. 174-181.

[2] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication", *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.

[3] H. K. Lu and A. Ali, "Communication Security between a Computer and a Hardware Token", in *Proc. Third Int. Conf. Sys. ICONS 2008*, Cancun, Mexico, 2008, pp. 220–225.

[4] R. Molva and G. Tsudik, "Authentication method with impersonal token cards", in *Proc. IEEE Comp. Soc. Symp. Res. Secur. Priv.*, Oakland, CA, USA, 1993, pp. 56–65.

[5] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to Biometrics*. New York: Springer, 2004.

[6] K. Ślot, *Wybrane zagadnienia Biometrii*. Warszawa: Wydawnictwo Komunikacji i Łączności, 2008 (in Polish).

[7] A. Czajka and A. Pacut, "Iris recognition system based on Zak-Gabor wavelet packets", *J. Telecom. Inform. Technol.*, no. 4, pp. 10–18, 2010.

[8] A. Czajka and A. Pacut, "Iris recognition with adaptive coding", in *Rough Sets and Knowledge Technology*, Lecture Notes in Artificial Intelligence, vol. 4481. Berlin: Springer, 2007, pp. 195–202.

[9] J. Hassell, Radius – Securing Public Access To Private Resources. O'Reilly & Associates, 2002.

[10] P. R. Calhoun, G. Zorn and P. Pan, "DIAMETER Framework Document", IETF, 2002.

[11] B. C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks", *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33–38, 1994.

[12] J. T. Kohl, B. C. Neuman, and T. Y. T'so, "The Evolution of the Kerberos Authentication System", in *Distributed Open Systems*, D. Johansen and F. M. T. Brazier, Eds. Los Alamitos, CA: IEEE Computer Society Press, 1994, pp. 78–94.

[13] D. Carrell, "The TACACS+ Protocol Version 1.78", Network Working Group INTERNET-DRAFT, Cisco Systems, 1997.

[14] "Information Technology – BioAPI – Biometric Application Programming Interface – Part 1: BioAPI Specification", ISO/IEC 19784-1.

[15] "Information Technology – BioAPI Interworking Protocol (BIP)", ISO/IEC 24708.

Tomasz Pałka graduated from the Faculty of Mechatronics of Warsaw University of Technology, Poland. Currently he works as a Specialist at Systems and Information Security Methods Team in NASK Research Division. His present areas of interest are centered around the security of information systems.

E-mail: tomasz.palka@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warszawa, Poland

Adam Kozakiewicz – for biography, see this issue, p. 21.

Paper

Drive Encryption and Secure Login to a Secure Workstation for Special Applications

Marek Małowidzki, Tomasz Dalecki, and Michał Mazur

Military Communication Institute, Zegrze, Poland

Abstract—We discuss the problem of a secure login to a virtualized workstation. For increased security, the workstation's hard drive is encrypted. During the startup, a decryption password to the drive must be entered by a user. We propose a solution that involves mutual authentication between the workstation and the user and ensures the password may be entered securely.

Keywords—drive encryption, evil maid, Linux, secure login, TPM, virtualization, Xen.

1. Introduction

Our work is a part of a project aiming at a development of a virtualization-based secure workstation, designed for running classified software or processing sensitive data. An important component of the overall security is hard drive encryption (portable media encryption is considered as well). Also, a workstation may be regarded as secure only if it runs original, unmodified software. Assuring this is easier in a physically protected environment; in case of portable computers, however, it poses a challenging problem. In the paper, we propose a solution that allows to combine hard drive encryption with a trustful boot process, preventing risk of software tampering.

The paper is organized as follows: First, we briefly describe the project. Then, we overview drive encryption software. The subsequent part of the paper defines the problem of secure logon, and discusses the boot process in details. We assess the security level offered by our solution, then propose further work and end the paper with conclusions.

2. Secure Station for Special Applications

The work described in the paper has been performed as a part of Secure Workstation for Special Applications[1] project, implemented by a consortium of four companies (the Military University of Technology, the Research and Academic Computer Network (NASK), Filbico, and the Military Communication Institute). The project's goal is to develop a virtualization-based workstation, designed for special applications. It is assumed that each virtual machine (except for the "control" domain 0) processes data from a different domain – by "data domain" we understand either a classification level (unclassified, restricted, etc.) or a purpose (e.g., a financial system, a human resources system, an IT software project), and because of that the data and related processing activities should be separated. This separation, together with increased security, provided by a hypervisor, is the most important reason for the selection of a virtualized environment.

By "special applications" of a workstation we mean its typical use in a classified system, e.g., in a chancellery for storing sensitive documents or as a machine for running classified software. While the focus of the project is put on desktop workstations, we also consider mobile computers (laptops). At this stage of the project, we do not consider networking (and related security risks). Generally, we silently assume the workstation is a standalone system.

After a review of open-source hypervisors that could be employed by the project, the consortium members selected Xen [1] (the version is 4.1). KVM [2] was another strong candidate. The final decision was made mainly on the basis of market presence and the significance of research projects related to each of the two hypervisors, and we believe in these two areas Xen evidently prevails.

3. Drive Encryption for Linux

The Linux distribution used in project is RedHat Enterprise 6.1. The default encryption system for this distribution is dm-crypt [3]. Our evaluation of dm-crypt confirmed that it fulfills all of our requirements, listed below:

- widely used and well-documented solution,
- open-source (or, with a friendly license),
- capable of full system encryption,
- easy to use in scripts,
- using standard format for encrypted volume (LUKS [4]),
- using a state-of-the-art encryption algorithm (AES).

[1]The project "Secure Workstation for Special Applications" has been founded by the Polish Ministry of Science and Higher Education under grant no. 0140/R/T00/2010/11.

Other solutions, namely, TrueCrypt [5] or eCryptfs [6], were also an option, although we rejected them as an unnecessary complication.
The dm-crypt (device mapper crypto target) package provides transparent encryption and decryption of block devices. It creates a logical device (e.g., `/dev/mapper/sda2crypt`) for each file or partition (e.g., `/dev/sda2`) we would like to encrypt. Reading from such a logical device involves a decryption operation performed in a background, while writing data through the device results in data encryption.

4. The Actual Problem: Secure Logon

In order to decrypt the drive, we need to pass a password (or, a key) to drive encryption software. Of course, as the password cannot be stored within the workstation (at least, in a plain form; but if it is encrypted, then we need a password-to-password), it must be entered by a user during workstation startup. This leads to an important question whether the workstation's environment can be trusted – that no malware is waiting to intercept the password (and store it somewhere for further retrieval, as in the evil maid case [7], [8]).
As one can see, for our secure workstation, we only selected an existing encryption package and had no plans to develop dedicated software, nor modify the selected one. However, the actual problem we face here is how a user could securely provide the password, or, how we could assure that the workstation has not been modified in any way. That is to say, we need to assure a secure logon. Or, we require the workstation to authenticate itself to a user before a user authenticates in the workstation [8].
The above mentioned evil maid attack is less likely in a secure (physically protected) external environment, where a workstation is intended to be installed and used. However, as we have mentioned previously, our project also considers a "mobile workstation" in the form of a laptop, and is trying to deliver a flexible solution for both cases.

5. Requirements

Drive encryption, correlated with secure logon, should fulfill a number of requirements. Some of them are more obvious, and some sound less apparent. The paramount requirements are as follows:

- efficient and effective data encryption, resistant to crypto attacks;
- workstation configuration (virtual machines and users) protected from disclosure;
- workstation environment (both software and data) shielded from tampering.

Additional, less apparent requirements are listed below:

- authentication of the workstation to a user: the workstation should convince the user that its environment is original and untampered;
- authentication of a user in the workstation, preferably, involving additional hardware and/or biometric elements;
- support for multi-access: we assume a number of user accounts, even in the case of a laptop;
- effective means to limit access to virtual machines for users that have no accounts there;
- easy and effective means to withdraw a user's access to the workstation.

The scope of encryption includes the whole hard drive without the content of `/boot` directory of the host system (domain 0). This directory contains the boot image, the initial file system (`initramfs` [9]) and user account data, packed and encrypted as a single *blob* file (refer to the next Section for details).

6. The Boot Process Step-By-Step

During the boot process, a number of software pieces are executed in a sequence (or, a *chain*). The boot may be considered secure if all the pieces are original (not modified by an attacker). To ensure this, every executing element measures its successor in the chain before passing the control to it, and writes the measurement to a Trusted Platform Module (TPM [10]) register. The first element in the chain, the Core Root of Trust for Measurement (CRTM) is a hardware-protected boot block code (e.g., the BIOS boot block code). The CRTM is considered trustworthy. The last element in our case is `initramfs`, measured by a bootloader. If all the measured values are correct and match the values of the data the TPM has *sealed* – i.e., when workstation boot software has been prepared and configured [11], the TPM enters the desired state and is able to perform cryptographic operations.
Before we proceed, we describe the content of a USB stick, which acts as a hardware key, and is prepared for each user during account creation. A BESTSTICK (BEST is an acronym from the Polish title of the project) contains four files, residing in an archive encrypted by the TPM:

- `login`: the file contains the user name;
- `password`: this is the random part of a password (to password, see below), generated using the `/dev/random` device; note that `/dev/random`, unlike the `/dev/urandom` pseudo-random device, may be considered as a true random generator, as it uses environmental "noise" (e.g., mouse movements or keyboard strokes);
- `phrase`: this is the authentication phrase;

– `image`: this file contains the authentication image, also selected by a user during the account creation. In practice, the image should be copied from another portable storage medium during the stick preparation.

The boot process, after `initramfs` has been loaded and the control passed to its main executable – `/init`, proceeds in the following steps:

1. A RAM drive is created and mounted. All further file operations are performed using the RAM drive.
2. A user's BESTSTICK is mounted and the archive it contains is copied to the RAM drive.
3. The archive is decrypted using the TPM, and the individual files are extracted. The stick is no more needed, so it is unmounted.
   Note that the TPM will switch to the desired state only if the workstation boot software chain (from CRTM to `initramfs`) has not been modified (also assuming that the software is bug-free). Otherwise, the attempt to unseal (decrypt) the secret will fail and the subsequent steps will not execute.
4. Now, the splash screen (plymouth [12]) is configured to display the phrase and picture, and the user is asked for his part of drive encryption password (`userpass`) – see Fig. 1.
5. The file containing encrypted password to the drive, `username.dom0.aes`, is extracted from the encrypted archive `/boot/best/users`; again, the TPM chip performs the decryption.
6. The file *`username`*`.dom0.aes` is decrypted using the AES algorithm, with a password composed of two parts – the random part and the user-provided part: (`userpass || password`).
7. The password to the hard drive is passed to `cryptsetup`; the RAM drive's content is overwritten with zero bytes, and then gets unmounted.
8. The user can log in to the workstation.

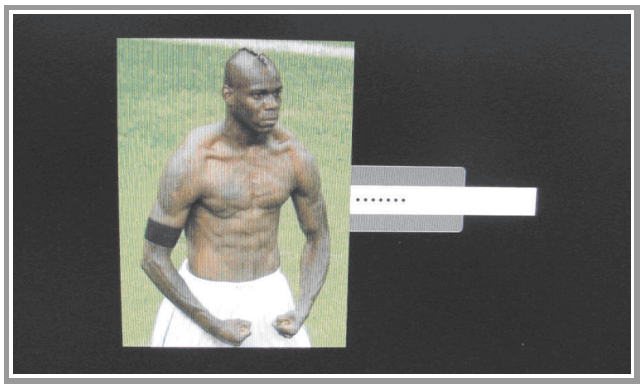


Fig. 1. The login screen (the phrase is visible after switching to console view).

In case of a "virgin" run (with no configured users), the above procedure is skipped and a special script is executed that allows to create the first (administrator) account.
The main components of the process are summarized in Fig. 2.

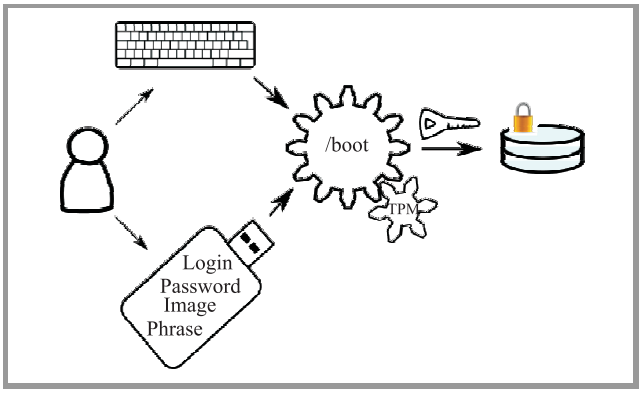


Fig. 2. Key components of the login process.

The process is executed by quite a large number of shell scripts. Most of them are embedded into `initramfs` (we use `dracut` [13] to automate `initramfs` generation). Some additional scripts perform administrative tasks (e.g., create a new user account and configure a USB stick, etc.), some are used for testing. Additional software installed on the workstation included TrustedGRUB [14] (the GRUB bootloader with TPM support), and two packages providing access to TPM: TrouSerS [15] and tpm-tools.
Note that the final effect we have reached is not completely satisfying. It is quite difficult to manage the display of the picture (a large picture happens to partially cover the password area, which looks bad). Also, we would prefer the phrase to be shown directly on the login screen. Unfortunately, while it was possible to modify and recompile `plymouth` code, we failed, for an unknown reason, to make it work after re-installation. Perhaps it requires some specific approach at some step.

7. Drive Encryption Revisited

We performed a broad set of tests to assess the efficiency of drive encryption of our Seagate Constellation 1000 GB HDD drive. A few jumbo files and a lot of small files were copied between encrypted and unencrypted partitions. The tests required some care as the system "cheats" by caching data read from the drive (for example, we rebooted the system often enough to prevent this behavior). Also, during the experiments we observed that the physical location on the drive where data are read from (or, written to) significantly influences performance, probably due to differences in drive access times. Moreover, these times often dominate the encryption. As a result, we can issue a conclusion that, while the overhead of encryption is difficult to measure precisely, it is quite small and does not lead to a negative user experience. We also performed tests for

a double encryption case, assuming that virtual domains may use additional encryption – see Section 8. This time, the performance degradation was observable but the overall system performance was still regarded as acceptable.

8. Security Assessment

In this Section, we assess the security of our architecture – the strong and weaker points and possible improvements. Note that we generally assume the worse usage scenario of a laptop stored and used in a (potentially) insecure environment.

CRTM and TPM. First, we rely on the Core Root of Trust Management and the Trusted Platform Module – that the whole boot process could be trusted. While some first successful attacks on TPM have been reported [16], they are enormously time- and resource-consuming (one could argue that with sufficient time and unlimited technical means every security mechanism could be compromised), and it seems we can assume that boot elements are secure enough.
At the same time, we assume the whole **boot software chain** is correct and immune to attacks (such as buffer overflows). Otherwise, an attacker could modify its behavior to prepare subsequent hostile actions while supplying original hash values to the TPM.

Phrase. This is an additional and quite a weak mechanism as the phrase could be intercepted relatively easily (e.g., by a hidden camera in a hotel room). Having said that, we can add that the mechanism could be augmented using a set of phrases, with a user selecting a phrase to display using some identifier (a phrase number or a few first letters). Assuming the user is careful enough and changes phrases during subsequent logins, the mechanism would be greatly strengthened.

Picture. At present, we assume it is much more difficult to "steal" the picture (e.g., with a precise photo). Unfortunately, even an imperfectly copied picture could suffice if the user is not careful enough (he might not notice some subtle differences at the pixel level). As above, a set of pictures could be employed to make an attack harder to prepare.

Password typing. A simple attack could be performed using a hidden camera in order to observe the password during typing. For increased security, if a user is able to type without looking at the keyboard, some form of a keyboard cover would prevent the snooping.

Hardware key (USB stick). The key must not be kept together with the workstation. The secrets (phrases and pictures) used for the workstation to authenticate itself to a user are decrypted automatically at startup, so they are revealed to any user, legitimate or not, who is powering the workstation with the hardware key present. There could be an additional (preliminary) password applied to decrypt the stick's content (together with TPM, of course – that is, both the preliminary password and TPM would be necessary to decrypt the stick). This would yield increased security at the cost of increased complexity of the login process.

Also, a plain USB stick may easily be copied (without leaving any trace), which also constitutes a security hole. Thus, we consider a smart card as a replacement for the stick.

Biometrics. Most modern laptops are already equipped with some sort of biometry device (e.g., fingerprint scanner), which would help in user identification.

Access withdrawal. This is relatively easy, as it is sufficient to remove a user's account by deleting the `username.dom0.aes` file (containing the encrypted password to domain 0). Even if a user refuses to return the USB stick (or, makes a copy of it), there is no way the user can log in to the workstation, and the stick becomes unusable.

Workstation configuration: users. User accounts (their presence and number) are protected; their data are encrypted.

Workstation configuration: virtual domains. Domains could be stored either as files within the encrypted file system, or using separate (encrypted) partitions (see the next paragraph). In the latter case, it seems that the number of domains could be observed but, having a sufficiently capacious hard drive, one could configure a large (sufficient in all scenarios plus some spare ones) number of partitions; some of them would be used by actual virtual domains, while other would contain random data. In this way, the presence and the number of domains would be concealed. However, this is not the full story yet: An attacker could observe changes in raw data and infer which partitions are used and which lie dormant, so some additional effort (altering the bytes from time to time) would be necessary.

Virtual domains encryption. Although our proposal concerns logging to domain 0, we assume that user domains

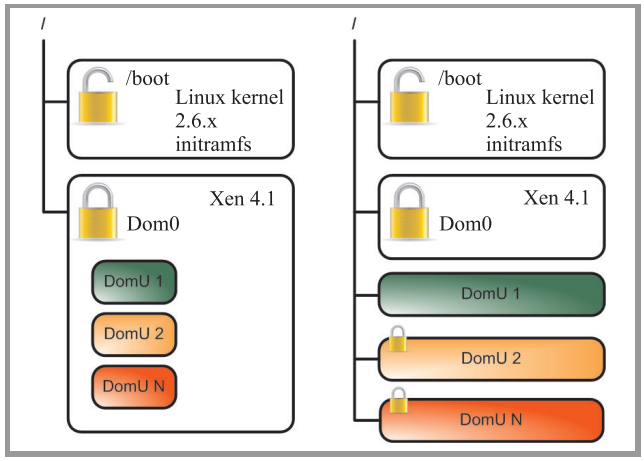


Fig. 3. Virtual domains as files within domain 0 (left) or separate drive partitions (right).

will be additionally encrypted, each using a separate password. The reason is that in case a malicious user (or, a malware that infected one of his virtual machines) is able to successfully break into domain 0, he will not be able to immediately attack inactive domains (lying dormant on the drive), as he does not know the decryption password. This is not a perfect protection mechanism (the attacker could install password interception malware), but it makes the attack on the other domains harder and more time-consuming.
We consider two possible approaches. The virtual domains may reside in files within the domain 0, or be stored in separate disk partitions (see Fig. 3).

Summary. As we have shown above, the logon process offers a reasonable level of security. This level could be increased by some additional mechanisms (multiple phrases, additional password, etc.). However, this also results in increased complexity. As [17] demonstrates, a user (human being) is the weak point; if a user finds the login process as unacceptably complex, he will also discover a way to simplify it in a way that definitely will also relax the security. Thus a sensible compromise must be found.
For an in-depth discussion of security of a system with encrypted drive in the context of the evil maid attack, possible prevention methods and ways to bypass them, refer to [7], [8], [18].

9. Related Work

The work [8], [19] concerns the same problem. We learned a lot from that work (especially the discussion of possible attacks), although technically our solution is different and offers some additional features required in our project – with multi-access (support for multiple user accounts) being probably the most important issue. In our case, the stick is not bootable; it serves for authentication purposes only. Other differences between [8], [19] and our solution are listed in Table 1. Note that by showing the comparison, we are not going to argue that our work is better in some way. We believe that it should be treated as a sort of continuation, and the table is presented for a reader's convenience.

Table 1
Comparison of features between [8] and our work

	Anti-evil maid	Our solution
Location of system authentication data	stick	stick
System authentication data	phrase	phrase and picture
Ability to use SRK password	yes	no
Part of the disk password on the stick	no	yes
Part of the password stored locally	no	yes
Location of the boot files (kernel, initramfs, ...)	stick	hard drive

10. Future Work

The main issues we would like to take into account in future involve the application of smart cards (as a replacement for the current USB sticks) and experiments with on-board biometric devices (in laptops). Also, we would like to make yet another attempt to modify `plymouth` and make it display both phrase and picture in a neat way.

11. Conclusions

In the paper, we have presented our proposal to implement a secure login to a Linux-based workstation with an encrypted hard drive. We believe it offers a reasonably high level of security; this level may additionally be increased at the cost of making things more complex (and, probably, less tolerated by a user). As a final remark we would like to note that our work remains valid also for a non-virtualized Linux system.

Acknowledgments

The authors would like to thank Joanna Rutkowska for a review and many insightful comments, which helped to improve the quality of the paper.

References

[1] "Xen" [Online]. Available: http://xen.org/

[2] "Kernel Based Virtual Machine" [Online]. Available: http://www.linux-kvm.org/page/Main_Page

[3] "Dm-crypt" [Online]. Available: http://www.saout.de/misc/dm-crypt/

[4] "LUKS + dm-crypt" [Online]. Available: http://code.google.com/p/cryptsetup/

[5] "TrueCrypt" [Online]. Available: http://www.truecrypt.org/

[6] "eCryptfs" [Online]. Available: http://launchpad.net/ecryptfs

[7] "The Invisible Things Lab's blog: Joanna Rutkowska: Evil Maid goes after TrueCrypt!" [Online]. Available: http://theinvisiblethings.blogspot.com/2009/10/evil-maid-goes-after-truecrypt.html

[8] "The Invisible Things Lab's blog: Joanna Rutkowska on Anti-Evil Maid" [Online]. Available: http://theinvisiblethings.blogspot.com/2011/09/anti-evil-maid.html

[9] "Initramfs" [Online]. Available: http://en.gentoo-wiki.com/wiki/Initramfs

[10] "Trusted Computing Group, TPM Main Specifications" [Online]. Available: http://www.trustedcomputinggroup.org/resources/tpm_main_specification

[11] "Trusted Computing: TCG proposals" [Online]. Available: http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/TrustedComputingTCG.html

[12] "Plymouth" [Online]. Available: http://www.freedesktop.org/wiki/Software/Plymouth

[13] "Dracut" [Online]. Available: https://dracut.wiki.kernel.org/index.php/Main_Page

[14] "TrustedGRUB – How does it work?" [Online]. Available: http://projects.sirrix.com/trac/trustedgrub/wiki/HowDoesItWork

[15] "TrouSerS: The open-source TCG Software Stack" [Online]. Available: http://trousers.sourceforge.net/

[16] "The H Security – Hacker extracts crypto key from TPM chip", 10 Feb. 2010 [Online]. Available: http://www.h-online.com/security/news/item/Hacker-extracts-crypto-key-from-TPM-chip-927077.html

[17] K. D. Mitnick, W. L. Simon, *The art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

[18] "Bruce Schneier's blog: Schneier on Security: Evil Maid Attacks on Encrypted Hard Drives" [Online]. Available: http://www.schneier.com/blog/archives/2009/10/evil_maid_attac.html

[19] "Joanna Rutkowska: Anti-Evil Maid installation script" [Online]. Available: http://git.qubes-os.org/?p=joanna/antievilmaid.git;a=blob;f=antievilmaid_install

Marek Małowidzki received his M.Sc.Eng. from Warsaw University of Technology. At present, he works at the Military Communication Institute where he carries out research on IT systems integration and security, and finalizes his Ph.D. on modeling traffic in TCP/IP networks. His main professional interests include distributed systems and technologies, programming languages and technologies, and TCP/IP networks.
E-mail: m.malowidzki@wil.waw.pl
Military Communication Institute
Warszawska st 22A
05-130 Zegrze Płd., Poland

Tomasz Dalecki received his M.Sc.Eng. from Warsaw University of Technology. He has been working at the Military Communication Institute since 2003. His research interests cover management systems design and implementation and security in IP-based systems.
E-mail: t.dalecki@wil.waw.pl
Military Communication Institute
Warszawska st 22A
05-130 Zegrze Płd., Poland

Michał Mazur received his M.Sc.Eng. from Military University of Technology, Warsaw. At present, he works at the Military Communication Institute where he is a programmer. His main professional interests include .NET and Java programming platforms as well as others programming languages and technologies.
E-mail: m.mazur@wil.waw.pl
Military Communication Institute
Warszawska st 22A
05-130 Zegrze Płd., Poland

Paper

A Survey of Energy Efficient Security Architectures and Protocols for Wireless Sensor Networks

Krzysztof Daniluk[a] and Ewa Niewiadomska-Szynkiewicz[a,b]

[a] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[b] *Research and Academic Computer Network (NASK), Warsaw, Poland*

Abstract—Data security and energy aware communication are key aspects in design of modern ad hoc networks. In this paper we investigate issues associated with the development of secure IEEE 802.15.4 based wireless sensor networks (WSNs) – a special type of ad hoc networks. We focus on energy aware security architectures and protocols for use in WSNs. To give the motivation behind energy efficient secure networks, first, the security requirements of wireless sensor networks are presented and the relationships between network security and network lifetime limited by often insufficient resources of network nodes are explained. Second, a short literature survey of energy aware security solutions for use in WSNs is presented.

Keywords—energy aware security architectures, routing protocols, security protocols, wireless sensor networks, WSN.

1. Introduction

A Wireless Sensor Network (WSN) is a distributed system composed of hundreds or thousands small-size, inexpensive, embedded devices deployed densely over a significant, often hostile area [1]. Each device can run applications and participate in transferring data to recipients within its range. The lack of fixed network infrastructure components in WSN allows creating unique topologies and enables the dynamic adjustment of individual nodes to the current network structure in order to execute assigned tasks.

WSNs have been identified as one of the most important technologies of this century. Due to their sensing capabilities, CPU power and radio transceiver plenty of sensor devices can be deployed in a sensing area, hence they can be used in applications, in which traditional networks are inadequate. However, nodes comprised by the network are often small battery-fed devices, which means their power source is limited [1]–[3]. The network's throughput is also limited. Moreover, the quality of wireless transmission depends on numerous external factors, like weather conditions or landform features. Part of those factors change with time.

Conventional networks with fixed infrastructure require protection against injection or modification of disseminated data packets and eavesdropping. Most applications of WSNs require the same protection. All well known attacks including traffic analysis, node replication, Denial of Service (DOS) and physical manipulating should be concerned. The security threads and attacks for all layers of the OSI model are discussed in [4]. Moreover, due to the spontaneous nature and shared wireless medium, sensor networks are more vulnerable to security attacks than wired ones. Using a computer with a wireless network adapter, anyone can gain an access to an unprotected network. Hence, the outsider can monitor the network, participate in the communication and easily launch attacks.

The main contribution of this paper is to point out the problems concerned with energy aware security architectures and protocols for IEEE 802.15.4 based WSN. It is a topic that has been a subject of intensive research in the recent years. The question is how to ensure the expected security level taking into account scarce resources of devices (network nodes). In Sections 2 and 3, we briefly summarize security requirements and security issues in WSN. Next, we present energy aware security architectures and protocols (Section 4), and energy efficient secure routing protocols (Section 5). The paper concludes in Section 6.

2. Security Requirements of WSN

Security for wireless sensor networks should focus on the protection of the data itself and the network connections between the nodes [5]–[8]. In general, security requirements often vary with application. In WSNs we can distinguish the following important requirements of security capabilities: authentication and authorization, availability, confidentiality, integrity and freshness. Thus, we need some mechanism for access authorization and protecting a mobile code. In many applications we need to protect fair access to communication channels and at the same time we often need to hide the information about physical location of our sensor node. Moreover, we need to secure routing and we have to defend our network against denial of service, malicious flows, node capturing and node injection, etc.

Authorization. Data authorization specifies access rights to resources and is strongly related to access control. Access control should prevent unauthorized users from participating in network resources. Hence, only authorized

users can join a given network. Access control relies on access policies that are formalized, like access control rules in a computer system. Most modern operating systems include access control.

Authentication. Message authentication implies a sender verification using cryptographic key. Authentication mechanisms are used to detect maliciously or spoofed packets. They are especially important in WSNs which use a shared wireless medium. In case of unicast transmission, an authentication can be guaranteed by symmetric key cryptography, using Message Authentication Code (MAC) in IEEE 802.15.4. Broadcast authentication requires more complex solutions (see [9]).

Availability. In secure network data should be safe and accessible at all times. Availability guarantees the survivability of network services against Denial-of-Service (DoS) attacks that can be launched at any layer of a wireless sensor network, and may disable a given device (network node) permanently. Moreover, DoS attack involved excessive computation and communication may exhaust battery charge of a sensor device.

Confidentiality. In WSN keeping sensitive data secret is the most important issue in case of critical applications in which highly sensitive data (secret keys, sensitive measurements, etc.) are collected and transmitted. Data confidentiality ensures that sensitive data is never disclosed to unauthorized users or entities. Hence, measurement data should not be available to neighboring nodes, and secure channels between nodes should be created. To protect a network against cyberattacks and malicious nodes, the routing information and sensor identities should remain confidential too. The standard approach to prevent end-to-end data confidentiality is to encrypt the data with a secret key.

Integrity and freshness. Data integrity is the quality of correctness, completeness, wholeness, soundness and compliance with the intention of the creators of the data. It is achieved by preventing unauthorized insertion, modification or destruction of data. In WSNs a malicious node may change messages to perturb the network functionality. Moreover, due to unreliable communication channels it is easy to inject infected packets or alerted data. In WSNs data integrity guarantees that a message being transferred is never corrupted, but providing data integrity is not enough for wireless communication. The compromised sensor nodes can listen to transmitted messages and replay attacks. Data freshness protects data against replay attacks by ensuring that the transmitted data is recent one.

3. Security in WSN

Cryptography is the common approach for defense against cyber attacks. However, maintaining an appropriate level of security and protection of sensitive information transmitted by a wireless sensor network requires solving many issues that are not present in traditional computer networks, and it is a challenging task [8], [10]. It should be underlined that the primary objective of wireless sensor networks is to make measurements for as long as possible. To do this it is essential to minimize energy use by reducing the amount of inter-node transmission and using energy aware algorithms and protocols [1], [2]. Due to limited resources of nodes forming WSN a balance between security capability and lifetime performance has to be obtained. Strong security protocols based on an asymmetric cryptography are difficult to implement. In general, asymmetric signatures are long and need high communication overhead, thus they are impractical for WSN applications. On the other side, week security protocols based on a symmetric cryptography may be easily broken. Moreover, due to a hostile deployment area, it is difficult to perform continuous surveillance of a network. To design a completely secure sensor network, security must be integrated into each node of WSN. Any network node implemented without any security could easily become a point of attack. Therefore, it is crucial to design WSN with security in mind from the very beginning. It is obvious that security usually adds some communication overhead and requires intensive computation and memory that is concerned with increased power consumption. The integration of security techniques in processing and communications simply allows for more efficient use of limited resources.

In general, three types of key management security schemes can be considered:

- *Trusted server scheme*. The symmetric key cryptography for data encryption is used. The process of establishing the key agreement between two communicating nodes is executed in the base station. Each node has to store only a single secret key. Thus, this solution is memory efficient, but energy expensive due to transmission overhead – each node has to communicate with the base station many times.
- *Self enforcing scheme*. The public key cryptography for communication between sensor nodes is used – DSA or RSA cryptography schemes. The disadvantage is that both DSA and RSA require complex computations (computing and energy expensive solution).
- *Key-predistribution scheme*. The symmetric key cryptography with limited number of keys stored in each sensor node is proposed. This solution is energy efficient – it does not introduce any additional transmission overhead for key exchange.

In many secure architectures and routing protocols, the clustering schemes for grouping all network nodes into disjoint and mostly non-overlapping clusters are applied to WSN [11], [12]. Generally, a cluster formation in WSN is based on the following characteristics: every node has to be connected to some clusters, nodes in a cluster must be able to communicate with others, often maximum diameter of all clusters in the network is the same. Most algorithms form clusters in distributed way through local broadcasts

with a maximum one or several (not many) hops. The cluster size is adapted to network capabilities and objectives. The cluster head is usually pre-assigned or picked randomly from the deployed set of nodes. Finally, we obtain a hierarchical communication structure: base station, cluster heads (various levels) and the lowest level formed by members of clusters (remaining nodes).

4. Energy Efficient Security Architectures and Protocols

In this section, we survey some of more and less common security solutions for IEEE 802.15.4 based networks. We start from the short description of the IEEE 802.15.4 security implementation. Next, we present various energy efficient architectures that can be employed in physical, data link, network, and middleware layers of the OSI communication model.

4.1. Security in IEEE 802.15.4

IEEE 802.15.4 is one of the first standards defining the radio and the medium access control layer for a low-power wireless sensor networks. ZigBee [13] is an industry alliance working on the 802.15.4 and upper protocol layers. Medium Access Control (MAC) protocols guarantee efficient access to the communication media while carefully managing the energy allotted to the node. This goal is typically achieved by switching the radio to a low-power mode based on the current transmission schedule. The comprehensive summary of MAC protocols for WSNs, and results of simulations that show their capabilities and efficiency in terms of the energy consumption are presented in [14]. The IEEE 802.15.4 network standard specification provides several security suits [15], [16]. The security suite specification defines the algorithms and operations that will be performed depending upon the security services to be provided. Each node can operate in secured or unsecured mode. A globally shared secret cryptographic key to message encryption and authentication is implemented. Eight security suites are defined in the IEEE 802.15.4 standard, and presented in Table 1. Each suit means a kind of cryptographic algorithm, the mode of block cipher, message authentication code, and the size of message authentication code. We can classify these suits based on provided properties, i.e., no security, encryption only (AES-CTR), authentication only (AES-CBC-MAC), and both encryption and authentication (AES-CCM). Thus, confidentiality is achieved through Advanced Encryption Algorithm (AES) in Counter mode (CTR), integrity through AES in Cipher Block Chaining Message Authentication Code (CBC-MAC) mode. The combination is offered with AES in the CTR with CBC-MAC mode (CCM).

Table 1
IEEE 802.15.4 security suite

	Security suite	Description
#0	Null	No security (default)
#1	AES-CTR	Encryption only, CTR mode
#2	AES-CBC-MAC-32	32 bit MAC
#3	AES-CBC-MAC-64	64 bit MAC
#4	AES-CBC-MAC-128	128 bit MAC
#5	AES-CCM-32	Encryption and 32 bit MAC
#6	AES-CCM-64	Encryption and 64 bit MAC
#7	AES-CCM-128	Encryption and 128 bit MAC

4.2. SPINS: Security Protocol for Sensor Network

The SPINS protocol developed by A. Perrig *et al.*, is described in [17]. It consists of two secure building blocks, i.e., Secure Network Encryption Protocol (SNEP) and micro version of Timed Efficient Stream Loss-tolerant Authentication (μTESLA). SNEP is used to provide confidentiality using encryption, and authentication, integrity and freshness of data using Message Authentication Code (MAC). In this approach all cryptographic primitives are constructed from a single block cipher for code reuse. Thus, the communication overhead is limited.
μTESLA is used for broadcasted data authentication. μTESLA requires that the base station and network nodes are loosely time-synchronized, and each node knows an upper bound on the maximum synchronization error. It generates authenticated broadcast message using symmetric key, and introduces asymmetric cryptography by delaying the disclosure of the symmetric keys. Therefore, μTESLA provides stronger security for networks with constrained resources. The implementation of SPINS requires about 220 bytes of RAM and 1580 to 2674 bytes of program space. An increase of energy consumption for security is about 20%.

4.3. TinySec: Link Layer Security Architecture for Wireless Sensor Networks

The problem with SPINS is that it has not been yet fully specified and implemented. TinySec is a link layer security architecture designed by Ch. Karlof *et al.*, and presented in [18]. Similarly to the SNEP protocol, it provides authentication, message integrity and confidentiality services. Replay protection has been intentionally omitted – the authors argued that this service belongs to the higher layers of the OSI model. The message authentication and integrity is provided using MAC, message confidentiality using encryption. Two security modes are possible – authentication only and authenticated encryption. In case of the first mode, the entire packet is authenticated using MAC, but the payload data is not encrypted. In case of the second mode, the payload data is encrypted and then authenticated with a MAC. Any keying mechanisms can be employed (single network-wide keys, per-link keys, group keys, etc.). TinySec is designed as a lightweight, energy efficient security package. It can be easily integrated into any WSN application. The implementation of TinySec requires about

728 bytes of RAM and 7146 bytes of program space. An increase of energy consumption depends on the mode and network technology, and is about 3% to 9,1% higher in compare to a normal TinyOS packet transmission.

4.4. LLSP: The Link-Layer Protocol

A Link-Layer Protocol (LLSP) was designed by L. E. Lighfoot *et. al*., and is described in [19]. The aim was to develop a protocol with less energy requirements than TinySec. LLSP guarantees various security requirements but focuses on three security services: message authentication, message confidentiality, and replay protection. AES-CBC mode of operation as the data encryption scheme is implemented in LLSP. The unique design of AES-CBC provides semantic security, i.e., encrypting the same plaintext twice will produce two different ciphertexts. A synchronous 4-byte counter between the sender and receiver pair is proposed to replay protection. Feedback Shift Register (FSR) is used to update this counter. The LLSP packet format is based on the TinySEC one (see Fig. 1). The difference is in a size – two byte counter values (Ctr) are removed from the security overhead in LLSP. As it was mentioned above both sender and receiver maintain a synchronous counter. Hence, the counter value has not to be transmitted, so the counter bytes are eliminated from each message packet. Thus, the LLSP security protocol reduces the energy usage without decreasing the security level.

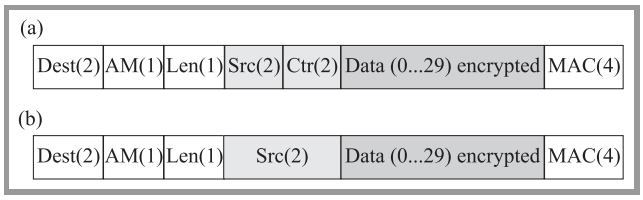


Fig. 1. Packet format in TinySec (a) and in LLSP (b).

The LLSP secure protocol was evaluated via simulation and compared with the TinySec protocol. Both applications were executed in the TOSSIM simulator (docs.tinyos.net/index.php/TOSSIM). The results are presented in [19]. From these results we can see that similar to most security protocols, the computational and energy costs increase for each packet transmission. It is concerned with extra computations and the larger packet size due to the security overhead. However, the authors of the LLSP protocol claim that using their solution the energy consumption is about 15% smaller than for TinySec, and latency reduction is about 3%.

4.5. LEAP/LEAP+: Localized Encryption and Authentication Protocol

LEAP [20] and LEAP+ [21] are lightweight, energy efficient security protocols for large scale sensor networks. They provide confidentiality and authentication services. LEAP was designed as a key management protocol to provide secure communication in WSNs. Due to various security requirements for different types of messages four types of keys for each network node are established: an individual key shared with a base station, a pairwise key shared with another node, a cluster key shared with a group of neighboring nodes, and a group key globally shared with all nodes in a network. The implementation of LEAP requires about 17.8 KB of program space. The RAM usage and energy costs depend on the number of nodes in a network.

4.6. Security Protocol Based on NOVSF

The cluster-based security protocol proposed in [22] uses a symmetric cryptography algorithm to guarantee security. To reduce the drawbacks of a symmetric cryptography and provide complete security, it employs the code-hopping technique using the Non-Orthogonal Variable Spreading Factor (NOVSF) codes. The NOVSF is an implementation of the non-blocking transmission of CDMA. In NOVSF codes, each OVSF code has 64 time slots, and any number of these time slots can be assigned to a channel. In NOVSF, the data blocks are assigned to time slots using different permutations in every session, Fig. 2. Hence, the blocks

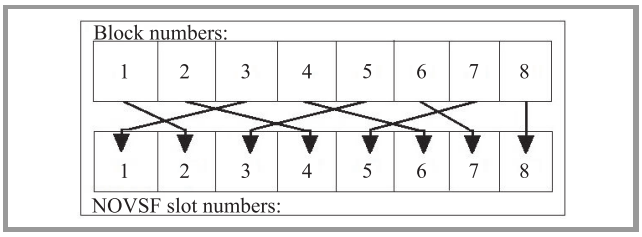


Fig. 2. Code-hopping technique.

of data are finally mixed, and such reordering method supports security. The algorithm operates as follows. First, it is assumed that all network nodes are grouped into disjoint and mostly non-overlapping clusters. As a result, a hierarchical communication structure consisting of a base station, cluster heads and the lowest level formed by members of clusters is obtained. Secondly, the following steps of the algorithm are performed:

Step 1: A base station periodically broadcasts the session key.

Step 2: Sensor nodes generate their cryptographic keys.

Step 3: The encrypted data are transmitted from sensor nodes to cluster heads using NOVSF code-hopping technique.

Step 4: Each cluster head appends its identifier number (ID) to this data and then forwards such data to the higher level cluster heads.

Step 5: The message is decrypted and authenticated by the base station.

To sum up, the transmission between nodes and cluster heads is encrypted. Based on periodically changed user specific session keys and NOVSF codes assigned to each node the authentication of messages is performed. Moreover, changing encryption keys from time to time guarantees data freshness in a network. The CBC-MAC protocol is used to provide data integrity. The total memory space for applied cryptographic primitives are about 2 KB. Hence, applying the NOVSF code-hopping technique increases security capabilities without requiring additional energy.

4.7. LSec: Lightweight Security Protocol

The Lightweight Security Protocol for distributed wireless sensor network (LSec) is described in [23]. It is the energy and memory efficient technique that assumes grouping network nodes into clusters. LSec provides following security capabilities: authentication, authorization, confidentiality of data, and protection against intrusions and anomalies. Both symmetric and asymmetric security schemes are used.

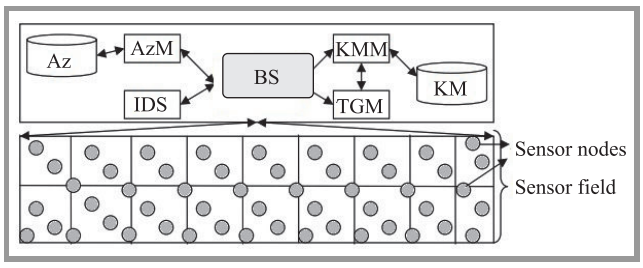


Fig. 3. LSec system architecture.

The LSec architecture consists of the following modules (see Fig. 3):

- KMM key management module: stores public and shared secret key of each node with a base station (BS) to the database (KM),
- TGM token generator module: generates the tokens for the requesters,
- AzM authorization module: checks whether a particular node is allowed to communicate with other node or a group of nodes,
- IDS intrusion detection; cluster heads send alert messages to IDS (lightweight mobile agents are installed in cluster heads).

LSec combines the features of trusted server scheme and self enforcing security scheme described in Section 3. It is assumed that the base station is the trusted party that never is compromised. Only the base station has an access to the public keys of all nodes in the network, and communicating nodes know each other's public keys only during the time of connection establishment. For every session, new random secret key is used. Each node has to store six keys (public key of node, private key of node, public key of BS, group key, public key of other node, session key). 72 bytes of memory are needed to store these keys. An asymmetric scheme is used for sharing ephemeral secret key between communicating nodes. Data is encrypted by using symmetric schemes. LSec is employed in the middleware layer of the communication model. It is scalable and memory efficient solution.
Authors claim that LSec is highly scalable and memory efficient – it introduces only 74.125 bytes of transmission and reception cost per connection. It provides stronger security and has the advantage of simple secure defense mechanism against compromised nodes.

4.8. HASF: The Hybrid Adaptive Security Framework

Hybrid Adaptive Security Framework (HASF) is a security architecture developed by T. Shon *et al.*, and described in [24]. This framework provides security capabilities with less extra energy usage than TinySec. In HASF, security functions are embedded to the network layer and the link layer (MAC) of the OSI model separately. The main idea is to provide hybrid adaptive security suite to each packet transmitted in a given WSN. The Hybrid Adaptive Security Suite (HASS) proposed in HASF is almost the same as the security suite proposed for IEEE 802.15.4, and presented in Table 1. The difference to commonly used architectures in HASS are as follows:

- *null* security is not provided,
- security suite is dynamically applied to MAC frame due to a type of a given WSN.

Three network characteristics are distinguished: *public*, *commercial*, *private*. Various security capabilities are provided to these groups of network. None confidentiality is guaranteed for public networks, more security capabilities are provided in commercial networks, and the strongest security is provided in private networks. All data are divided into control and application. *Control data* means a message or signal to manage the network operation. *Application data* means a kind of data concerned with WSN services. The attributes of these data are: periodic, urgent-periodic, on-demand, event-driven. The decisions on security levels in case of different network characteristics are presented in Table 2. In [24] authors discuss the results of application of their framework to a testbed network formed by the devices using HASS approach. They compared three kinds of nodes: IEEE 802.15.4 based system with no security, HASS based system with the AES encryption algorithm,

Table 2
Hybrid Adaptive Security Suite decision table

Feature		#1	#2	#3	#4	#5	#6	#7
Public (32)	App	+	+					
	Ctrl	+				+		
Commercial (64)	App			+		+		
	Ctrl	+					+	
Private (128)	App				+		+	
	Ctrl	+				+		+

Table 3
Summary of selected security architectures for WSN

Architecture	Security services	Properties
SPINS	Authentication, authenticated broadcast, confidentiality, integrity, freshness.	Consists of SNEP and μTesla (secure building blocks). Symmetric cryptography support. Encryption (CTR mode), Block Cipher (RC5). Not fully implemented and specified. Requires 2674 bytes of program space (max). Transmission overhead to 20%.
TinySec	Authentication, confidentiality, integrity, replay protection.	Link layer architecture easily integrated into WSN. Symmetric cryptography support. Encryption (CBC mode), Block Cipher (Skipjack). Requires 728 bytes of RAM, 7146 bytes of program space (max). Transmission overhead to 9.1%.
LLSP	Authentication, confidentiality, replay protection.	Link layer architecture. Symmetric cryptography support. Semantic security. 2 bytes less packet format (energy cost reduction without security decreasing). Transmission overhead to 7.7%.
LEAP/LEAP+	Authentication, confidentiality, intrusions protection, anomalies protection.	Symmetric cryptography support. Encryption (RC5), Block Cipher (RC5). Four types of keys available for each sensor node: individual, pairwise, cluster, group. Defence against: HELLO Flood, Sybil, Wormhole attacks. Requires about 17.8 KB of program space. RAM usage and transmission overhead depend on the number of nodes.
NOVSF-based	Authentication, confidentiality. integrity, freshness.	Works partially in the physical layer. Symmetric cryptography support. The security increased via code-hoping technique using NOVSF data blocks (assigned to time slots using permutations in every session). User specific session keys (periodically changed). Clustering-based algorithm. Requires about 2 KB of memory space.
LSec	Authentication, authorization, confidentiality, replay protection, intrusions protection, anomalies protection.	Both symmetric and asymmetric cryptography support. Public Key cryptography support. Base station – the trusted party – a single point of failure. Implemented in the middleware. Clustering-based algorithm. Simple Secure key exchange scheme: 6 keys that takes only 72 bytes of memory. Transmission overhead to 8.33%.
HASF	Authentication, confidentiality, integrity.	Provides Hybrid Adaptive Security Suite. Security functions embedded to network and link layer separately. Security mechanism dynamically applied to MAC frame. Three network types with different security (public, commercial, private). Transmission overhead to 4.8%.

and the Crossbow device based on TinySec architecture and the RC5 encryption algorithm. In the case of described experiments, the extra energy usage due to providing security functionalities was about 4.8% in case of HASS based system and 5.2% in case of TinySec based Crossbow system. The results confirmed that HASF outperforms the other common security techniques.

4.9. Summary of Security Architectures

The Table 3 presents the summary of our survey – security architectures, provided services and their main properties.

5. Secure Energy Efficient Routing Protocols

Security architectures using a globally shared key are ineffective in presence of insider attacks or compromised nodes. Therefore, more sophisticated defense mechanisms are necessary to provide reasonable protection against wormholes and insider attacks, and detect malicious nodes. Secure routing protocols can be used to improve WSN security. In this section, selected routing protocols for secure networks are presented. Similarly to the solutions described in previous sections we focus on energy aware solutions.

5.1. SERP: Secure Energy Efficient Routing Protocol

The secure energy efficient routing protocol for wireless sensor networks (SERP) is described in [25]. The main idea of this protocol is to provide a robust transmission of authenticated and confidential data from the source sensor with limited energy budget to the base station. It is dedicated to WSNs with densely deployed relatively static sensor devices.

Three main objectives were considered during design of SERP:

– energy aware organization of the network to ensure energy efficient transmission, and finally maximum lifetime of the network,

– secure transmission; nodes should have the capability to detect falsely injected reports,

– robust and resilient transmission; any node failure would not greatly hamper the performance of a network.

The protocol operates in two main phases: creating a backbone network and secure data transmission. A sink rooted tree structure is created as the backbone of the network taking into consideration balanced energy consumption. Next, a minimum number of forwarding nodes in the network is selected. The backbone network is restructured periodically. It is used for authenticated and encrypted data delivery from the source sensors to the base station. A one way hash chain and pre-stored shared secret keys are used for ensuring secure data transmission. An optional key refreshment mechanism that could be applied depending on the application is introduced for data freshness.
The energy saving mechanism is based on disable the radio transceivers of selected nodes. The nodes in a network can operate in two main states: *non-forwarding* – the transceiver is switched off, *forwarding* – both transceiver and sensing devices are switched on. It is assumed that after the backbone structure is constructed, all nodes are either in forwarding or non-forwarding states. Nodes with the non-forwarding state turn off their radio transceivers while keeping the sensing device active. On the other hand, forwarding nodes keep both radio and sensing device active. All nodes sense the environment, and after detecting any event the non-forwarding nodes turn on their radios and transmit data towards the base station via nodes in a selected path.
The SERP protocol was evaluated via simulation. Ns-2 simulator (www.isi.edu/nsnam/ns/) was used for performance analysis. SERP was compared with two popular energy aware routing protocols – LEACH [26] and EAD [27]. The simulation results are presented and discussed in [25]. The authors claim that SERP is a very competitive solution compared to the LEACH and EAD protocols w.r.t. energy requirements. Moreover, SERP provides security functionalities.

5.2. EENC: Energy Efficiency Routing with Node Compromised Resistance

A novel energy efficiency routing protocol with node compromised resistance (EENC) was developed by K. Lin *et al.*, and described in [28]. EENC bypasses the compromised nodes and improves the accuracy of packets under the condition of balancing the energy consumption. The reinforcement learning based on the ant colony optimization is used to complete routing tables. The trust values are assigned to all nodes of a network. The trust value is computed and based on the multiple behavior attributes such as: packet drop rate, forwarding delay rate, etc. These values are used to detect the malicious nodes. Each node in a WSN computes the trust values of its one hop neighbors. The idea of EENC was to provide security with minimal energy consumption. To achieve this, each node storages trust values of all its neighbors and manages its energy resources.
The EENC protocol operates as follows. To transmit data the secure and energy efficient route is computed. The calculation process consists of many rounds, each divided into three phases.

- Routing detecting phase. A certain number of forward ants are generated to search for route leading to the sink. Each ant records the information about the minimum amount of energy and minimum trust value for nodes along the path, and the hop number for each node.

- Pheromone updating phase. The sink node generates a backward ant, which carries all data collected by the forward ant. These data are used to update the pheromone value concerned with each node in a path.

- Routing maintaining phase. The route for a given source and sink nodes is established based on trust values and updated pheromone values of the nodes carried during the pheromone updating phase.

The EENC protocol was evaluated via simulation. The considered performance metric included lifetime of a network and a packet correctly received ratio. The EENC performance was compared with two other routing algorithms, i.e., DRP and MTRP described in [29]. Simulation results presented in [28] confirm that the routing established via EENC can bypass most compromised nodes in the transmission path and EENC has high performance in energy efficiency. It was observed in the experiments that the calculated lifetime and the successful packet delivery ratio were much higher for EENC than those obtained for DRP and MTRP.

5.3. REWARD Routing Protocol

The REceive WAtch ReDirect (REWARD) routing protocol for WSNs is described in [30]. This algorithm can be used to detect black hole attacks [4]. In such attacks, a malicious node acts as a black hole to attract all the traffic in a WSN through a compromised node. A compromised node is usually placed in the center and looks attractive to surrounding nodes and collect most traffic destined for a base station.
In REWARD, the distributed database including suspicious nodes and areas is created. Two types of broadcast messages, i.e., MISS (Material for Intersection of Suspicious Sets) and SAMBA (Suspicious Area, Mark a Black-hole

Attack) are used to organize this database. MISS is used to detect identifiers of malicious nodes, and SAMBA is used to identify physical locations of suspicious nodes.
The operation of the REWARD protocol is as follows. In case of demand-driven routing protocols, the query for path establishing is sent to the destination node. The destination node sends its location and waits for a packet. The destination node broadcasts a MISS message if a packet does not arrive within a specified period of time. It copies the list of all the involved nodes from the query to this MISS message – these nodes are under suspicion. The ratings for the nodes are introduced, and path metrics are calculated by averaging the node ratings in the path. The path with the highest value of a metric is selected – in this way the suspicious nodes are avoided. If a node attempts a black hole attack and drops a package, it is detected by the next node in the path. After a predefined time period, the node transmits the packet changing the path and broadcasts a SAMBA message that provides the location of the black-hole attack.
REWARD is the energy aware protocol and can be applied to networks formed by devices that can tune their transmit power. Different levels of security with less and more overhead according to a network capabilities are provided. The performance of the protocol is discussed in [30]. The authors compared the energy overhead of two variants of REWARD.

6. Summary and Conclusions

Many challenges arise from application of wireless ad hoc networking. We focused on one of them that is very important in wireless sensor networks – secure data protection and data transmission in WSN with limited resources. The paper provides a short overview of some representative energy efficient security techniques. We briefly discussed the security requirements of WSNs and showed the relationships between techniques for forming secure networks, and energy aware WSNs. Next, we described and compared based on literature survey selected energy aware architectures and protocols in WSNs that can be implemented in the physical, data link, network, and middleware layers of the OSI model.
In summary, we can say that due to scarce resources, unique properties of wireless sensor networks, and often hostile environments it is a challenging task to protect sensitive information transmitted by nodes forming a WSN. Due to limited resources of nodes that form WSN many solutions providing strong security are impractical in this type of network. Therefore, we can find many security considerations that should be investigated in the nearest future.

Acknowledgment

This work was partially supported by National Science Centre grant NN514 672940.

References

[1] M. C. Vuran, I. F. Akylidiz, *Wireless Sensor Networks*. Wiley, 2010.

[2] E. Niewiadomska-Szynkiewicz, P. Kwaśniewski, and I. Windyga, "Comparative study of wireless sensor networks energy-efficient topologies and power save protocols", *J. Telecom. Inform. Technol.*, no. 3, pp. 68–75, 2009.

[3] A. Tiwari, P. Ballal, and F. L. Lewis, *Energy-efficient wireless sensor network design and implementation for condition-based maintenance*, *ACM Trans. Sensor Netww (TOSN)*, vol. 3, no. 1, pp. 1–23, 2007.

[4] K. Sharma, M. K. Ghose, D. Kumar, "A comparative study of various security approaches used in wireless sensor networks", *Int. J. Adv. Sci. Technol.*, vol. 17, pp. 31–44, 2010.

[5] M. Ahmad, M. Habib, and J. Muhammad, "Analysis of security protocols for Wireless Sensor Networks", in *Proc. 3rd Int. Conf. Comp. Res. Develop. ICCRD 2011*, Shanghai, China, 2011, vol. 2, pp. 383–387.

[6] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks", *J. ACM Trans. Sensor Netw. (TOSN)*, vol. 5, no. 3, 2009.

[7] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, S., *Energy efficient schemes for wireless sensor networks with multiple mobile base stations*, in *Proc. IEEE Global Telecom. Conf. GLOBECOM'03*, San Francisco, USA, 2003, vol. 1, pp. 377–381.

[8] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: a survay, in *Security in Distributed Grid, Mobile and Pervasive Computing*, Y. Xiao, Ed. Auerbach Publication, 2007.

[9] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu, "Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards", *Comp. Commun*., vol. 30, no. 7, pp. 1655–1695, 2007.

[10] H. Kumar and A. Kar, "Wireless sensor network security analysis", *Int. J. Next-Generation Netw. (IJNGN)*, vol. 1, no. 1, 2009.

[11] S. K. Singh, M. P. Singh, and D. K. Singh, "A survay of energy-efficient hierarchical cluster-based routing in wireless sensor networks", *Int. J. Adv. Netw. Appl.*, vol. 2, no, 2, pp. 570–580, 2010.

[12] S. K. Singh, M. P. Singh, and D. K. Singh, "Energy-efficient homogenous clustering algorithm for wireless sensor networks", *Int. J. Wirel. Mob. Netw.*, vol. 2, no. 3, pp. 49–61, 2010.

[13] ZigBee Alliance, "ZigBee Specification v1.0", New York, USA, 2005.

[14] M. I. Shukur, L. S. Chyan, and V. V. Yap, "Wireless sensor networks: delay guarantee and energy efficient MAC protocols", *World Academy of Sci., Engin. Technol.*, vol. 50, pp. 1061–1065, 2009.

[15] N. Sastry, D. Wagner, "Security consideration for IEEE 802.15.4 networks", in *Proc. 5th Int. Conf. Web Inform. Sys. Engin. WISE 2004*, Brisbane, Australia, 2004, pp. 32–4.

[16] R. Struik and G. Rason, "Security and security architectural recommendations for the IEEE 802.15.4 Low-Rate WPAN", Certicom Corp., 2002.

[17] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. Culler, "SPINS: security protocols for sensor networks", *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.

[18] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks", in *Proc. 2nd Int. Conf. Embedded Networked Sensor Sys.*, Baltimore, MD, USA, 2004, pp. 162–175.

[19] L. E. Lighfoot, J. Ren, and T. Li, "An energy efficient link-layer security protocol for wireless sensor networks", in *Proc. IEEE Int. Con. Elec.-Infor. Technol. EIT 2007*, Chicago, IL, USA, 2007, pp. 233–238.

[20] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", in *Proc. 10th ACM Conf. Comp. Commun. Secur. CCS 2003*, Washington, DC, USA, 2003, pp. 62–72.

[21] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks", *ACM Trans. Sensor Netw. TOSN*, vol. 2, no. 4, pp. 500–528, 2006.

[22] H. Cam, S. Ozdemir, D. Muthuavinashiappan, and P. Nair, "Energy efficient security protocol for wireless sensor networks", in *Proc. IEEE 58th Veh. Technol. Conf. VTC 2003*, Orlando, Florida, USA, 2003, vol. 5, pp. 2981–2984.

[23] R. A. Shaikh, S. Lee, M. A. U. Khan, and Y. J. Song, "LSec: lightweight security protocol for distributed wireless sensor network", *Lecture Notes in Computer Science*, vol. 4217, 2006.

[24] T. Shon, B. Koo, H. Choi, and Y. Park, "Security architecture for IEEE 802.15.4-based wireless sensor network", in *Proc. 4th Int. Symp. Wirel. Pervasive Comput. ISWPC 2009*, Melbourne, Australia, 2009, pp. 1–5.

[25] A. K. Pathan and C. S. Hong, "SERP: secure energy-efficient routing protocol for densely deployed wireless sensor network", *Annales des Telecomm.*, pp. 529–541, 2008.

[26] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in *Proc. 33rd Annual Hawaii Int. Conf. Sys. Sciences HICSS'00*, Maui, Hawaii, USA, 2000, pp. 3005–3014.

[27] B. Azzedine, C. Xiuzhen, and J. Linus, "Energy-aware datacentric routing in microsensor networks", in *Proc. 6th Int. Symp. Model. Analys. Simul. Wirel. Mobile Sys. MSWiM 2003*, San Diego, CA, USA, 2003, pp. 42–49.

[28] K. Lin, Ch. F. Lai, X. Liu, and X. Guan, "Energy efficiency routing with node compromised resistance in wireless sensor networks", *Mob. Netw. Appl.*, vol. 17, pp. 75–89, 2012.

[29] Z. Yu and Y. Guan, "A dynamic en-route scheme for filtering false data injection in wireless sensor networks", in *Proc. 25th IEEE Int. Conf. Com. Commun. INFOCOM 2006*, Barcelona, Spain, 2006, pp. 1–12.

[30] Z. Karakehayov, "Using REWARD to detect team black-hole attacks in wireless sensor networks", in *Proc. Worksh. Real-World Wirel. Sensor Netw. REALWSN'05*, Stockholm, Sweden, 2005, pp. 1–5.

Krzysztof Daniluk received his M.Sc. in Computer Science from the Trinity College Dublin, The University of Dublin, in 2010 and Polish-Japanese Institute of Information Technology, Warsaw, in 2010. Currently he is a Ph.D. student in the Institute of Control and Computation Engineering at the Warsaw University of Technology. His research area focuses on wireless sensor networks, energy-efficiency together with security issues, computer networks.
E-mail: K.Daniluk@stud.elka.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Ewa Niewiadomska-Szynkiewicz – for biography, see this issue, p. 38.

Paper

Improvement of the Performance of Database Access Operations in Cellular Networks

Mustafa Vahabzadeh Dolama and Akbar Ghaffarpour Rahbar[1]

Computer Networks Research Lab, Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

Abstract—Reducing the traffic volume of location updating is a critical issue for tracking mobile users in a cellular network. Besides, when user *x* wants to communicate with user *y*, the location of user *y* must be extracted from databases. Therefore, one or more databases must be accessed for updating, recording, deleting, and searching. Thus, the most important criterion of a location tracking algorithm is to provide a small database access time. In this paper, we propose a new location tracking scheme, called Virtual Overlap Region with Forwarding Pointer (VF), and compare the number of database accesses required for updating, deleting, and searching operations for the proposed scheme and other approaches proposed for cellular networks. Our VF scheme like Overlap Region scheme reduces the updating information when a user frequently moves in boundaries of LAs. Unlike Overlap Region, the VF can reduce number of database accesses for searching users' information.

Keywords—Cellular networks, deleting cost, GSM, searching cost, tracking mobile users, UMTS, updating cost.

1. Introduction

In mobile communications (e.g., GSM, UMTS, 3G,...), the location of users is not fixed and may change in time. Therefore, to make a communication between user *x* and user *y*, the system must first find the location of user *y*. Therefore, the location of users must be tracked from time to time [1]. In mobile communications, a small geographical area (called cell) is served by a Base Station (BS). Several cells are grouped into a Location Area (LA) and several LAs make a mobile communications network. The Mobile Terminals (MTs) in a cell directly communicate with the BS of the cell. Several BSs are connected to a Base Station Controller (BSC) and several BSCs are connected to a Mobile Switching Center (MSC) [2], [3].

In a typical telephone system, we have one database that stores all users' information permanently. Therefore, the location of each user can be found easily by searching the database. However, in cellular networks the location of users is not fixed. When a user enters a new location, the information of this user must be updated. With the increase of the number of mobile users in cellular networks, the database access time becomes a bottleneck because more database access operations (for updating, deleting, searching, and recording new information) are necessitated in time [4]. Thus, choosing a good algorithm for tracking users in cellular networks depends on the number of needed database accesses.

The objective of this paper is to propose a new method for tracking mobile users and compare the number of database accesses for the proposed method with other available methods. The proposed scheme uses the concept of virtual overlap region and forwarding pointer, but with a different policy for updating the information when a user frequently moves in boundaries of LAs in one overlap region. Indeed, the number of database accesses for searching users' locations and updating their information can be reduced efficiently. To the best of our knowledge, this is the first time mobile tracking schemes have been compared based on their database access operations (updating, searching, deleting, etc.), except our recent work in [5].

Our contributions in this paper are proposal of the Virtual Overlap Region with Forwarding Pointer (VF) location tracking scheme, and comparison of location tracking schemes based on database access operations.

The remainder of this paper is organized as follows. Location management schemes are explained in Section 2. The proposed VF method is described in Section 3. In Section 4, we compare location management methods. Finally, a brief conclusion is presented in Section 5.

2. The Schemes Proposed for Tracking Mobile Users

Many strategies have been proposed to reduce the overhead of database accesses in mobile communications networks [6]–[13]. In this section, we will briefly describe and compare some location management approaches such as two-tier architecture [7], Forwarding Pointer [6], Virtual Layer [8], Virtual Layer with Forwarding Pointer [9], and Overlap Region [10].

2.1. Two-Tier Architecture

Two-tier architecture [7] uses a two-level database system: HLR that maintains all permanent information of each user and a pointer to another database; and Visitor Location Register (VLR) that stores temporary location information

[1] Corresponding Author.

of users. The VLR database is maintained at each LA. Therefore,

- When mobile user x enters the mobile communications network (i.e., user turns the mobile on), a new record is created in both HLR and VLR in order to store the information of user x. Thus, one HLR and one VLR accesses are required.
- When mobile user x moves from LA_i to LA_j, the information of the user x in VLR_i is deleted and a new record is created in VLR_j. In addition, a message is sent to HLR by VLR_j in order to update the user x pointer from VLR_i to VLR_j. Therefore, one HLR access and two VLR accesses are necessary.
- When mobile user x decides to call mobile user y:
  - if both user x and user y are in the same LA_i, the location of user y is found from VLR_i. Thus, one VLR access is needed;
  - if both user x and user y are not in the same LA_i, first, the location of user y is searched in VLR_i. Since the information cannot be found in VLR_i, the relevant VLR_j can be found from HLR. Finally, the location of user y is found from VLR_j. Therefore, one HLR access and two VLR accesses are required to find the location of user y.

  Since the access of the HLR database takes more time than the access of a VLR database due to the large size of the HLR database, the two-tier architecture can reduce the search cost when both user x and user y are in the same LA. However, when user x and user y are not in the same LA, the HLR, the new VLR, and old VLR all must be accessed for appropriate functions. This, in turn, increases the number of database accesses.
- Finally, when user x turns his/her mobile off or exits from the mobile communications network, the information of user x in HLR and VLR should be deleted. To delete the information of user x, one HLR and one VLR accesses are necessitated.

2.2. Forwarding Pointer

When a user frequently moves in a boundary between LAs, more HLR accesses are required for updating in the two-tier architecture and HLR may likely become a bottleneck. The Forwarding Pointers scheme [6] has been proposed to efficiently reduce the volume of HLR accesses required for updating. In this approach, the main idea is to set up a forwarding pointer from an old database to a new database when a user leaves the old LA toward a new LA. Therefore,

- When mobile user x enters a mobile communications network (i.e., user x turns his/her mobile on), a new record is created in both HLR and VLR in order to store the information of user x. Thus, one HLR and one VLR accesses are needed.
- When mobile user x moves from LA_i to LA_j, a new record is created in VLR_j and a pointer is set to VLR_j from VLR_i. Therefore, two VLR accesses are only needed.
- When mobile user x calls mobile user y:
  - If both user x and user y are in the same LA_i, the location of user y is either directly found from VLR_i, or is following the pointers chain. Thus, l VLR accesses are necessitated where l is the length of the pointers chain. We have $l = 1$ if the information is retrieved directly from VLR_i.
  - If both user x and user y are not in the same LA_i, the location of user y is first searched in VLR_i and the relevant pointers chain. Since the information cannot be found, the relevant VLR_j can be found from HLR. Finally, the location of user y is either directly found from VLR_j or by following the pointers chain. Therefore, one HLR access and $2 \times l$ VLR accesses are needed to find the location of user y.
- Finally, when user x turns his/her mobile off or exits from the mobile communications network, the information of user x in HLR and VLR must be deleted. One HLR and l VLR accesses are needed to delete the information of user x.

Since no update is required in the HLR database, the update cost goes down. When the length of the pointer chain is less than five, according to analytical estimation in [6], this scheme can reduce the total cost by 20% to 60%. Although, this scheme can reduce the total cost, the frequent updates problem still exists when a user moves back and forward in the boundary of an LA.

2.3. Virtual Layer Scheme

The virtual layer scheme [8] has been proposed to construct a new location database architecture (see Fig. 1). The bold lines in Fig. 1 represent the original layer and the dotted lines represent the virtual layer. For every virtual layer, one VLR is needed (i.e., subVLR).
In this scheme, one SubMSC is necessitated for each virtual layer. The SubMSCs are connected to the covered MSC. For example in Fig. 1, consider MT_x moves from position A to B, B to C and then comes back to position A. Initially in position A, HLR and VLR1 have created an entry for MT_x. When MT_x moves to position B, the SubMSC4 creates a new entry for MT_x and VLR1 must be updated. Then, when MT_x moves from position B to C and C to A, no update is needed because the virtual layer has not changed.

The goal of this scheme is to reduce both location updating rate and location updating cost, especially when the MTs

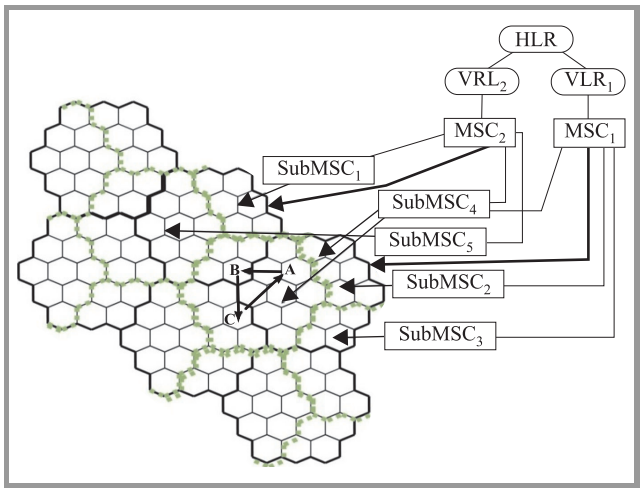


Fig. 1. The demonstration of the virtual layer.

reside near the boundaries of LA and frequently cross through the boundary to another LA.

- When mobile user x enters the mobile communications network (i.e., user x turns the mobile on), a new record is created in both HLR and VLR in order to store the information of user x . Thus, one HLR and one VLR accesses are necessitated.
- When VLR is active: mobile user x moves from one LA_i to LA_j:
  - If the information of user x already exists in $subVLR_k$, VLR_i must be deactivated and $subVLR_k$ must be activated. Therefore, one VLR and one subVLR accesses are necessary.
  - If the information of user x does not exist in $subVLR_k$, a new record is created in $subVLR_k$ and the information in previous subVLR must be deleted. Besides, VLR_i must be deactivated and $subVLR_k$ must be activated. Therefore, one VLR and two subVLR accesses should be done.
- When subVLR is active: mobile user x moves from virtual layer i to virtual layer j:
  - If the information exists in VLR_k, the information in VLR_k is updated. VLR_k must be activated and $subVLR_i$ must be deactivated. Therefore, one VLR and one subVLR accesses are required.
  - If the information does not exist in VLR_k, a new record is created in VLR_k and the previous record must be deleted. Hence, a message is sent to HLR by VLR_k in order to update the user x VLR pointer (from previous VLR to VLR_k). Besides, VLR_k must be activated and $subVLR_i$ must be deactivated. Thus, one HLR access, one subVLR, and two VLR accesses are required.
- When mobile user x calls mobile user y:
  - If both user x and user y are in the same LA_i, the location of user y is found from VLR_i. Thus, one VLR access is needed.
  - If both user x and user y are not in the same LA_i, the location of user y is first searched in VLR_i. Since the information cannot be found from VLR_i, the relevant VLR_j can be found from HLR. Finally, the location of user y is found from VLR_j. Therefore, one HLR access and 2 VLR accesses are necessitated to find the location of user y.
- Finally, when user x turns his/her mobile off, or exits from the mobile communications network, the information of user x in HLR and VLR should be deleted. For this purpose, one HLR and one VLR accesses are necessary.

2.4. Virtual Layer with Forwarding Pointers

Chang and Lin have proposed an improved scheme [9] that uses forwarding pointers in virtual layer to reduce the update cost. The possible state of a user in this scheme is:

- When mobile user x enters the mobile communications network (i.e., user x turns the mobile on), a new record is created in both HLR and VLR in order to store the information of user x. Thus, one HLR and one VLR accesses are necessitated.
- When VLR is active: mobile user x moves from one LA_i to LA_j:
  - If the information of user x already exists in $subVLR_k$, VLR_i must be deactivated and $subVLR_k$ must be activated. Therefore, one VLR and one subVLR accesses should be performed.
  - If the information of user x cannot be found in $subVLR_k$, a new record is created in $subVLR_k$ and the information in previous subVLR must be deleted. Besides, VLR_i must be deactivated and $subVLR_k$ must be activated. Therefore, one VLR and two subVLR accesses are required.
- When subVLR is active: mobile user x moves from virtual layer i to virtual layer j:
  - If the information exists in VLR_k, the information in VLR_k is updated. VLR_k must be activated and $subVLR_i$ must be deactivated. Therefore, one VLR and one subVLR accesses are needed.
  - If the information does not exist in VLR_k, a new record is created in VLR_k and a message is sent by VLR_k to previous VLR to set a pointer to VLR_k. Furthermore, VLR_k must be activated

and subVLR$_i$ must be deactivated. Thus, one subVLR and two VLR accesses are required.

- When mobile user x calls mobile user y:
  - If both user x and user y are in the same LA$_i$, the location of user y is directly found from VLR$_i$ or by following the pointers chain. Thus, l VLR accesses are needed.
  - If both user x and user y are not in the same LA$_i$, first the location of user y is searched in VLR$_i$ and relevant pointers chain. Since the information cannot be found, the relevant VLR$_j$ can be found from HLR. Finally, the location of user y is directly found from VLR$_j$ or by following the pointers chain. Therefore, one HLR access and $2 \times l$ VLR accesses are required to find the location of user y.
- Finally, when user x turns his/her mobile off, or exits from the mobile communications network, the information of user x in HLR and VLR should be deleted. To do this, one HLR and l VLR accesses are required.

2.5. Overlap Region

The Virtual Layer scheme [8] requires the reconstruction of the mobile communications network architecture. The architecture requires extra equipments. To overcome the reconstruction of the mobile communications network, the Virtual Overlap scheme [10] with time stamp has been proposed. Figure 2 depicts the structure of the Virtual Overlap [10]. Each Overlap Region (OR) has seven LAs. The bold line in Fig. 2 represents the Overlapping Region for LA$_5$, and therefore, we have OR$_5$ = {LA$_1$, LA$_2$, LA$_4$, LA$_5$, LA$_6$, LA$_9$, LA$_{10}$}. In Fig. 2, the OR for LA$_6$ is OR$_6$ = {LA$_2$, LA$_3$, LA$_5$, LA$_6$, LA$_7$, LA$_{10}$, LA$_{11}$}. Each LA has an associated MSC and VLR.

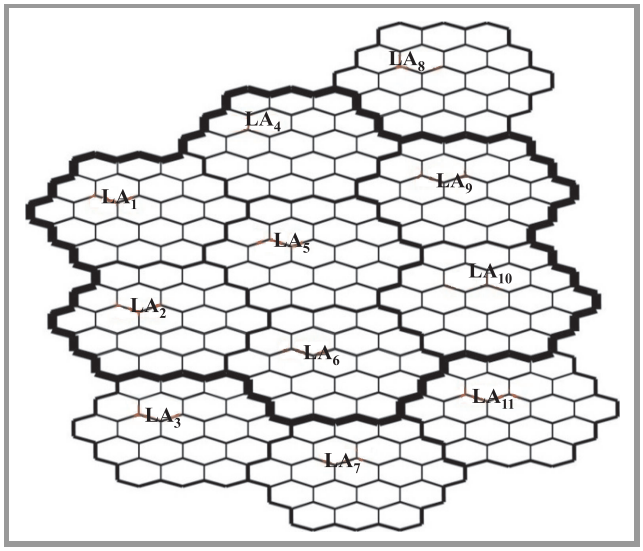


Fig. 2. The structure of virtual overlap in mobile communications network.

In the Virtual Overlap scheme [10], each VLR has two fields: TS which indicates the time that a mobile user enters the associated LA, and OR which indicates the Overlap Region in which the mobile user has registered last time. Therefore,

- When mobile user x enters the mobile communications network (i.e., user x turns his/her mobile on), a new record is created in both HLR and VLR in order to store the information of user x. Thus, one HLR and one VLR accesses are necessary.
- When mobile user x moves from LA$_i$ to LA$_j$:
  - If LA$_i$ and LA$_j$ are in the same virtual overlap region, a new record is created in VLR$_j$ and the TS field of VLR$_j$ records the current time. Therefore, one VLR access is necessitated.
  - If LA$_i$ and LA$_j$ are not in the same virtual overlap region, a new record is created in VLR$_j$ and the TS field of VLR$_j$ records the current time. Thus, a message is sent to HLR in order to update the user x data. Furthermore, the information of user x in previous OR (with seven VLRs) must be deleted. Therefore, one HLR and eight VLR accesses are required.
- When mobile user x calls mobile user y:
  - If both user x and user y are in the same LA$_i$, the location of user y is found from seven VLRs in the relevant OR. Thus, seven VLR accesses are needed.
  - If both user x and user y are not in the same LA$_i$, first the location of user y is searched in VLR$_i$. Since the information cannot be found from VLR$_i$, a message is sent to HLR by VLR$_i$ and then the relevant VLR$_j$ can be found in HLR. Finally, the associated overlap region is found from the OR field of VLR$_j$, and then the location of user x is searched in seven VLRs in the relevant OR. Therefore, one HLR access and eight VLR accesses are required to find the location of user y.
- Finally, when user x turns his/her mobile off or leaves the mobile communications network, the information of user x in HLR and seven VLRs on the OR that user has resided before should be deleted. To delete the information of user x, one HLR and seven VLR accesses are necessary.

3. The Virtual Overlap Region with Forwarding Pointer Scheme

In this section, we shall propose a new approach for location updating based on the concepts of the virtual overlapping region and forwarding pointers. The goal of our VF is to

reduce the number of database accesses for updating and searching the information.

3.1. The Architecture of VF

Now, we detail the VF scheme. The VLR database that maintains current user location information keeps two fields as the LA ID and pointer PO. The LA ID field indicates the identification number of a LA and the PO field is a pointer to another VLR. Note that each LA has a unique identifier number. If the LA ID is –1, the PO field is used to find the LA ID in another VLR.

3.2. The Procedure of Location Registration

When a new mobile user (i.e., mobile user x) resides in a location area LA_i, the associated database VLR_i will create a new entry for mobile user x and will record the LA identification number. Then, the system gives the LA ID to the mobile user x and sends a message to the HLR to record the current location of mobile user x. When mobile user x moves, the procedure of the location registration is as follows:

- When mobile user x detects a new LA_j, the mobile user x sends the LA ID that assigned previously in LA_i to the associated service switch through its BS.
- Determine if the new LA_j and previous LA_i belong to the same overlap region.
- If yes, the VLR_i will update its LA ID to LA_j ID.
- If no, the VLR_j will create a new entry for mobile user x and sends a message to VLR_i in order to set a pointer to VLR_j and change the LA ID field to –1. Then, a new location number is sent by VLR_j to mobile user x.

3.3. The Procedure of Call Delivery

When mobile user x wants to call mobile user y in LA_i, the following steps are required for the call delivery as:

- The system first searches mobile user y in VLR_i.
- Determine whether mobile user y can be found in VLR_i.
- If yes, the mobile user y's LA is retrieved from the LA ID field:
  - According to the location information of VLR, the service switch MSC_j can be found.
  - The service switch MSC_j determines the cell location of the mobile user y and assigns a Temporary Location Directory Number (TLDN) to mobile user y. Then, the TLDN is returned from the MSC_j to the MSC_i. By this way, MSC_i knows where to send the information relevant to mobile user x.
- If no, a message is sent to HLR:
  - From the HLR, the associated VLR can be found.
  - According to the location information of VLR, the service switch MSC_j can be found.
  - The service switch MSC_j determines the cell location of the mobile user y and assigns a TLDN. Then, the TLDN is returned from the current VLR to the HLR.
  - Upon receiving the TLDN, if the current VLR is different from the last VLR registered, the HLR updates the relevant pointer to point to the current VLR, and deletes the chain of forwarding pointers.
  - The HLR sends the TLDN to the original switch (i.e., MSC_i) and the connection between the caller user and the called user is set up using the TLDN.

According to the above details, the numbers of database accesses in VF are as follows:

- When mobile user x enters a mobile communications network (i.e., user x turns his/her mobile on), a new record is created in both HLR and VLR in order to store the information of user x. Thus, one HLR and one VLR accesses are necessary.
- When mobile user x moves from LA_i to LA_j:
  - If LA_i and LA_j are in the same virtual overlap region, the LA ID field in VLR_i is updated. Therefore, one VLR access is necessitated.
  - If LA_i and LA_j are not in the same virtual overlap region, a new record is created in VLR_j and a pointer is set up from VLR_i to VLR_j. therefore, two VLR database accesses are needed.
- When mobile user x calls mobile user y:
  - If both user x and user y are in the same LA_j, the location of user y is found from LA ID in the relevant VLR. Thus, one VLR access is needed.
  - If both user x and user y are not in the same LA_j, first the location of user y is searched in VLR_j. Since the information cannot be found from VLR_j, a message is sent to HLR by VLR_j and then the relevant VLR_i can be found in HLR. Finally, the location of user x is obtained from LA ID in the relevant VLR. Therefore, one HLR access and 2 VLR accesses are required to find the location of user y.
- Finally, when user x turns his/her mobile off or leaves the mobile communications network, the information of user x in HLR and l VLRs must be deleted. To delete the information of user x, one HLR and l VLR accesses are necessary.

Table 1
Comparison of database accesses

Scheme		Two-tier architecture [7]		Forwarding Pointer [6]		Virtual Layer [8]		Virtual Layer with Forwarding Pointers [9]		Overlap Region [10]		VF	
Operation		HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR
User is turned on		1	1	1	1	1	1	1	1	1	1	1	1
User is turned off		1	1	1	l	1	1	1	l	1	7	1	l
Searching a user	minimum access	0	1	0	1	0	1	0	1	0	7	0	1
	maximum access	1	2	1	$2\times l$	1	2	1	$2\times l$	1	8	1	$2\times l$
User moves from one LA to another LA	minimum access	1	2	0	2	0	0	0	0	0	1	0	1
	maximum access	1	2	0	2	1	3	0	3	1	8	0	2

Table 2
Ranking of schemes when $l < 7$

Scheme	Two-tier architecture [7]	Forwarding Pointer [6]	Virtual Layer [8]	Virtual Layer with Forwarding Pointers [9]	Overlap Region [10]	VF
User is turned on	1	1	1	1	1	1
User is turned off	1	2	1	2	3	2
Searching a user	1	2	1	2	3	2
User moves from one LA to another LA	5	1	3	2	4	1

4. Performance Evaluation

In this section, we shall compare VF scheme with the schemes stated in Section 2. First, the number of database accesses under different schemes will be illustrated for each possible action of a user. Then, we shall discuss the number of databases by an example.

4.1. Performance Evaluation of Database Accesses

In mobile communications, tracking mobile users could be the most important issue. Therefore, a good scheme must provide a small database when a user moves from one LA to another LA. Table 1 shows the comparison of different schemes in terms of the number of database accesses for possible status of a user. In this table, VF, Virtual Layer and Virtual Layer with Forwarding Pointers have small database accesses, but Virtual Layer and Virtual Layer with Forwarding Pointers need reconstruction of the mobile communications network. Furthermore, when the length of the chain in Forwarding Pointer and Virtual Layer with Forwarding Pointer schemes goes up, the number of database accesses increases. Since the access of HLR database takes more time, Overlap Region and VF reduces an update cost when an user goes back and forth in boundary of LAs (just need one VLR access) which is comparable with the two-tier architecture (that needs one HLR and two VLRs accesses). While searching the user location, the VF, two-tier architecture and Virtual Layer always provide small number of database accesses, and Overlap Region has more database accesses than other schemes.

Based on the number of database accesses, Table 2 ranks the proposed schemes when the Forwarding Pointer chain length is $l < 7$. When a user mobile is turned on, a new record is created in HLR and VLR databases. Therefore, the number of database accesses for all schemes are the same. When a user mobile is turned off, all information must be deleted. In this case, the overlap region scheme is the worst. For searching user information, again, the overlap region scheme has more database accesses. For a movement from one LA to another, which is more important in cellular networks, VF and Forwarding Pointer schemes are the best candidates and the two-tier scheme is the worst.

On the other hand, Table 3 depicts the ranking of schemes when the Forwarding Pointer chain length is $l \geq 7$. In this

Table 3
Ranking of schemes when $l \geq 7$

Scheme	Two-tier architecture [7]	Forwarding Pointer [6]	Virtual Layer [8]	Virtual Layer with Forwarding Pointers [9]	Overlap Region [10]	VF
User is turned on	1	1	1	1	1	1
User is turned off	1	3	1	3	2	3
Searching a user	1	3	1	3	2	3
User moves from one LA to another LA	5	1	3	2	4	1

situation, the number of database accesses for deleting and searching the information in VF, Forwarding Pointer and Virtual Layer with Forwarding Pointer goes up.

4.2. Impact of Users' Mobility

Figure 3 shows an example in which user x moves from position A to position F through positions B, C, D, E, and F.

- Initially, user x enters LA_5 or is turned on in LA_5. The following procedures are performed:
  - VLR_5 creates a new entry for user x.
  - VLR_5 sends a registration message to HLR to create an entry and to set a pointer to VLR_5.
- When user x moves from A to B:
  - **Two-tier architecture**. VLR_9 creates a new record for user x and sends a message to HLR to update information. Then, the information in VLR_5 is deleted.

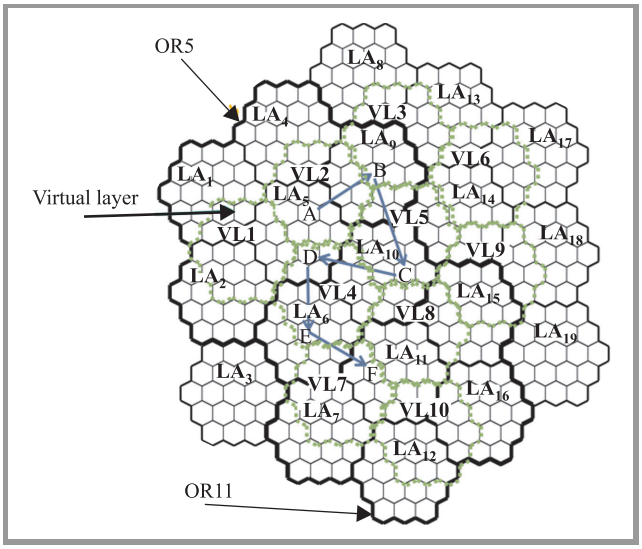


Fig. 3. An example of user movement in mobile communications network.

  - **Forwarding Pointer**. VLR_9 creates a new record for user x and sends a message to VLR_5 to set a pointer to VLR_9.
  - **Virtual Layer**. When user x enters LA_9, a new record is created in $subVLR_2$ and VLR_5 is deactivated. Then, user x enters the virtual layer 3 from virtual layer 2. Therefore, a new record is created in VLR_9 and the information in VLR_5 is deleted. Hence, a message is sent to HLR by VLR_9 to update relevant information.
  - **Virtual Layer with Forwarding Pointer**. When user x enters LA_9, a new record is created in $subVLR_2$ and VLR_5 is deactivated. Then, user x enters the virtual layer 3 from virtual layer 2. Therefore, a new record is created in VLR_9 and the information in VLR_5 is deleted. Hence, a message is sent to VLR_5 to set a pointer to VLR_9.
  - **Overlap Region with Time Stamp**. When a user enters LA_9, because LA_9 is in OR_5 a new record is created in VLR_9 and the TS field of VLR_9 stores the time that user has entered LA_9. Moreover, the OR field of VLR_9 stores the user x overlap region number (OR_5).
  - **VF**. When a user enters LA_9, the LA ID field in VLR_5 is updated from 5 to 9 since LA_9 is in OR_5.
- Movement from position B to position C:
  - **Two-tier architecture**. VLR_{10} creates a new record for user x and sends a message to HLR to update information. Then, the information in VLR_9 is deleted.
  - **Forwarding Pointer**. VLR_{10} creates a new record for user x and sends a message to VLR_9 to set a pointer to VLR_{10}.
  - **Virtual Layer and Virtual Layer with Forwarding Pointer**. When user x enters LA_{10}, a new record is created in $subVLR_5$ and VLR_9 is deactivated. Furthermore, the information of user x is deleted from $subVLR_2$.

  - **Overlap Region with Time Stamp**. When a user enters LA_{10}, because LA_{10} is in OR_5 a new record is created in VLR_{10} and the TS field of VRL_{10} records the time that user has entered LA_{10}. In addition, the OR field of VLR_{10} stores the user *x* overlap region number (OR_5).
  - **VF**. When a user enters LA_{10}, the LA ID field in VLR_5 is updated from 9 to 10 because LA_{10} is in OR_5.

- When user *x* moves from position C to position D:
  - **Two-tier architecture**. VLR_5 creates a new record for user *x* and sends a message to HLR to update information. Then, the information in VLR_{10} is deleted.
  - **Forwarding Pointer**. VLR_5 updates user *x* information, because the information already exists in VLR_5. Then, a message is sent to VLR_{10} to set a pointer to VLR_5.
  - **Virtual Layer**. When user *x* crosses the boundary of virtual layers in the direction of C to D, VLR_{10} creates a new record and sends a message to HLR to update information. Then, the information in VLR_9 is deleted. When a user reenters LA_5 again, a new record is created in $subVLR_4$ and VLR_{10} is deactivated. Furthermore, the information of user *x* is deleted from $subVLR_5$.
  - **Virtual Layer with Forwarding Pointer**. When user *x* crosses the boundary of virtual layers in the direction of C to D, VLR_{10} creates a new record and sends a message to VLR_9 to set a pointer to VLR_{10}. Then, when a user enters LA_5, a new record is created in subVLR4 and VLR_{10} is deactivated. Furthermore, the information of user *x* is deleted from $subVLR_5$.
  - **Overlap Region with Time Stamp**. When a user enters LA_5, because LA_5 is in OR_5 and the information already exists in VLR_5, the TS field of VLR_5 is only updated.
  - **VF**. When a user enters LA_5, because LA_5 is in OR_5 the LA ID field in VLR_5 is updated from 10 to 5.

- When user *x* moves from position D to position E:
  - **Two-tier architecture**. VLR_5 creates a new record for user *x* and sends a message to HLR to update information. Then, the information in VLR_{10} is deleted.
  - **Forwarding Pointer**. VLR_6 creates a new record for user *x* and sends a message to VLR_5 to set a pointer to VLR_6.
  - **Virtual Layer and Virtual Layer with Forwarding Pointer**. Since the movement is in the same virtual layer, no update is required.
  - **Overlap Region with Time Stamp**. When a user enters LA_6, because LA_6 is in OR_5 a new record is created in VLR_6 and the TS field of VLR_6 stores the time that user has entered LA_6. Furthermore, the OR field of VLR_6 stores the user *x* overlap region number (OR_5).
  - **VF**. When a user enters LA_6, because LA_6 is in OR_5 the LA ID field in VLR_5 is updated from 5 to 6.

- Finally, user *x* moves from position E to position F:
  - **Two-tier architecture**. VLR_{11} creates a new record for user *x* and sends a message to HLR to update information. Then, the information in VLR_6 is deleted.
  - **Forwarding Pointer**. VLR_{11} creates a new record for user *x* and sends a message to VLR_6 to set a pointer to VLR_{11}.
  - **Virtual Layer**. When user *x* crosses the boundary of virtual layers in the direction of E to F, VLR_6 creates a new record and sends a message to HLR to update information. Then, the information in VLR_{10} is deleted. Then, when user enters LA_{11}, a new record is created in $subVLR_7$ and the information of user *x* is deleted from $subVLR_4$.
  - **Virtual Layer with Forwarding Pointer**. When user *x* crosses the boundary of virtual layers in the direction of E to F, VLR_6 creates a new record and sends a message to VLR_{10} to set a pointer to VLR_6.Then, when a user enters LA_{11}, a new record is created in $subVLR_7$ and VLR_6 is deactivated and $subVLR_7$ is activated.
  - **Overlap Region with Time Stamp**. When a user enters LA_{11}, because LA_{11} is not in OR_5, a new record is created in VLR_{11} and the TS field is set to the current time. In addition, the OR field of VLR_{11} is set to OR_{11}. Then, a message is sent to HLR by VLR_{11} to update the information. After all, the information of user *x* is deleted from all VLRs in OR_5.
  - **VF**. When a user enters LA_{11}, because LA_{11} is not in OR_5 a new record is created in VLR_{11} and a pointer is set up from VLR_5 to VLR_{11}. In addition the LA ID field in VLR_5 is updated to –1 and user *x* takes new LA ID from VLR_{11} (i.e., 11).

Let's suppose that user *y* in LA_{14} wants to call user *x*. First, VLR_{14} is queried, but the relevant information cannot be found. Hence, a message is sent to HLR by VLR_{14}.

  - **Two-tier architecture**. From the HLR database, the associated VLR (i.e., VLR_{11}) is found and the information is retrieved from VLR_{11}.

Table 4
Comparison of database accesses under the example showed in Fig. 3

Scheme	Two-tier architecture [7]		Forwarding Pointer [6]		Virtual Layer [8]		Virtual Layer with Forwarding Pointers [9]		Overlap Region [10]		VF	
Path	HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR	HLR	VLR
A (Initial)	1	1	1	1	1	1	1	1	1	1	1	1
A → B	1	2	0	2	1	3	0	3	0	1	0	1
B → C	1	2	0	2	0	3	0	3	0	1	0	1
C → D	1	2	0	2	1	5	0	5	0	1	0	1
D → E	1	2	0	2	0	0	0	0	0	1	0	1
E → F	1	2	0	2	1	5	0	5	1	8	0	2
User y call user x	1	2	1	5	1	2	1	5	1	8	1	2
Total database access	7	13	2	16	5	19	2	22	3	21	2	9
Normalized cost $C_{U,T}/C_{U,V}$	$7\alpha+13$		$2\alpha+16$		$5\alpha+19$		$2\alpha+22$		$3\alpha+21$		$2\alpha+9$	

– **Forwarding Pointer**. From the HLR database, the associated VLR (i.e., VLR_5) is found and the information is retrieved from VLR_5 by following the chains (i.e., VLR_6, VLR_{11}).

– **Virtual Layer**. From the HLR database, the associated VLR (i.e., VLR_{11}) is found and the information is retrieved from VLR_{11}.

– **Virtual Layer with Forwarding Pointer**. From the HLR database, the associated VLR (i.e., VLR_5) is found and the information is retrieved from VLR_5 by following the chains (i.e., VLR_6, VLR_{11}).

– **Overlap Region with Time Stamp**. From the HLR database, the associated VLR (i.e., VLR_{11}) is found. Then, the information is searched in OR_{11} that consists of VLR_6, VLR_7, VLR_{10}, VLR_{11}, VLR_{12}, VLR_{15}, and VLR_{16}.

– **VF**. From the HLR database, the associated VLR (i.e., VLR5) is found and the information is retrieved from VLR5 by following the chains (i.e., VLR_{11}).

Table 4 shows the number of database accesses among different schemes for this example. We assume that all database accesses have the same cost.
Let the database access cost for HLR ($C_{U,H}$) be equal to

$$C_{U,H} = \alpha \times C_{U,V}\,, \tag{1}$$

where $C_{U,V}$ is the VLR access cost and $\alpha \geq 1$. Then, the total database access cost ($C_{U,T}$) according to VLR access cost can be obtained from Eq. (2).

$$C_{U,T} = C_{U,H} + C_{U,V}\,. \tag{2}$$

From Eq. (2), the normalized access cost value of $C_{U,T}/C_{U,V}$ can be obtained (see the last row in Table 4).

As a result for the example in Fig. 3, the VF has the smallest database accesses in total (11 accesses: 9 VLR and 2 HLR accesses). Therefore, this scheme is better than others. Virtual Layer and Virtual Layer with Forwarding Pointer need reconstruction of the mobile communications network. Overlap Region scheme reduces database access for updating, however, it needs more database accesses for searching the location of users.

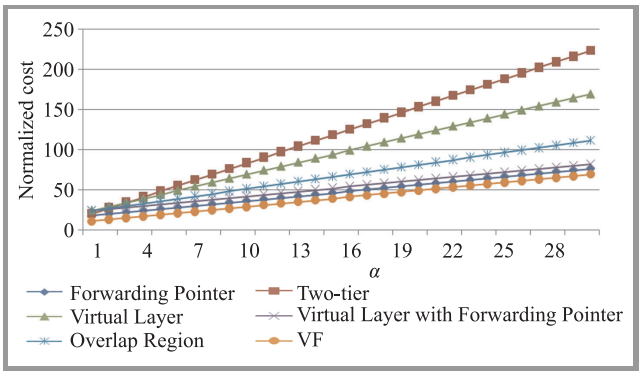


Fig. 4. Normalized cost for example of Fig. 3 for different values of α.

According to Fig. 4, with the increase of α, the two-tier architecture scheme has larger cost than the other schemes. Note that the HLR database must be accessed for every action in mobile communications network including searching, updating, deleting, and creating new record. Since in VF, Forwarding Pointer and Virtual Layer with Forwarding Pointer, access to HLR is avoided by using the forwarding pointer chain from one VLR to another VLR, the cost of these schemes is lower than others.

5. Conclusion

In this paper, we have studied five location management schemes and the number of database accesses for inserting, updating, deleting, and searching operations. When a user frequently makes a call to other users, Overlap Region

needs more database accesses than others. In Forwarding Pointer, when a user frequently moves within boundaries of LAs, the number of database accesses becomes high. Virtual Layer and Virtual Layer with Forwarding Pointer need the reconstruction of mobile communications network. With the increases of the mobile users in the mobile communications network, the size of the HLR database goes up and the two-tier architecture cannot be a good scheme at all. This is because the HLR database must be accessed for every action including inserting, updating, deleting, and searching operations.

In addition, we have proposed a new scheme (VF) and compared it with other schemes. According to our comparisons, VF has a small number of database accesses when a user frequently moves within the boundary of LAs. For searching the user location, VF still has smaller database accesses than others. Therefore, VF could be the best candidate either when a user frequently moves within boundaries of LAs or frequently makes a call to other users.

References

[1] K. T. Chen, S. L. Su, and R. F. Chang, "Design and analysis of dynamic mobility tracking in wireless personal communication networks", *IEEE Trans. Veh. Technol.*, vol.51, no. 3, pp. 486–497, 2002.

[2] R. Jain, Y. B. Lin, C. Lo, and S. Mohan, "A caching strategy to reduce network impacts of PCS", *IEEE J. Selec. Areas Commun.*, vol. 12, no. 8, pp. 1434–1444, 1994.

[3] J. Li and Y. Pan, "Dynamic database management for PCS networks", in *Proc. 21st Int. Conf. Distrib. Comput. Sys.*, Phoenix, Arizona, USA, 2001, pp. 683–686.

[4] E. Pitoura and G. Samaras, "Locating objects in mobile computing", *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 4, pp. 571–592, 2001.

[5] M. V. Dolama and A. G. Rahbar, "Performance evaluation of the number of database accesses in cellular networks," in *Proc. 2nd Int. Conf. Wirel. Mobile Networks WiMo-2010*, Ankara, Turkey, 2010, pp. 46–58.

[6] R. Jain and Y. B. Lin, "An auxiliary user location strategy employing forwarding pointers to reduce network impacts of PCS", *Cellular Networks*, vol. 1, pp. 197–210, 1995.

[7] E. Pitoura and G. Samaras, "Locating objects in mobile computing", *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 4, pp. 571–592, 2001.

[8] D. Chung, H. Choo, H. Y. Youn, and D. R. Shin, "Reduction of location update traffic using virtual layer in PCS", *The Human Society and the Internet*, LNCS 2105, pp. 398–410, 2001.

[9] C. C. Chang and I. C. Lin, "The strategy of reducing the location update traffic using forwarding pointers in virtual layer architecture", *Comp. Standards and Interfaces*, vol. 25, no. 5, pp. 501–513, 2003.

[10] C. C. Chang, I. C. Lin, and C. C. Lin, "A novel location tracking scheme for reducing location updating traffic in a personal communication system", *Wirel. Personal Commun.*, vol. 44, pp. 139–152, 2008.

[11] I. F. Akyildiz and W. Wang, "A dynamic location management scheme for next-generation multitier PCS system", *IEEE Trans. Wirel. Commun.*, vol. 1, no. 1, pp. 178–189, 2002.

[12] G. Krishnamurthi, S. Chessa, and A. K. Somani, "Optimal replication of location information in mobile networks", in *Proc. IEEE Int. Conf. Commun. ICC'99*, Vancouver, Canada, pp. 1768–1772, 1999.

[13] Y. B. Lin, and W. N. Tsai, "Location tracking with distributed HLR's and pointer forwarding", *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 58–64, 1998.

Mustafa Vahabzadeh Dolama received his B.Sc. degree in Computer Software from Azad University, Khoy, Iran in 2008, and M.Sc. degree in Computer Networks from Sahand University of Technology, Sahand New Town, Tabriz, Iran in June 2010. His current research interests include cellular networks, scheduling, and passive optical networks.

E-mail: m_vahabzadeh@sut.ac.ir
Department of Electrical Engineering
Sahand University of Technology
Sahand New Town, Tabriz, Iran

Akbar Ghaffarpour Rahbar an Associate Professor in Electrical Engineering department at Sahand University of Technology, Sahand New Town, Tabriz, Iran. Dr. Rahbar received his B.Sc. and M.Sc. degrees in Computer Hardware and Computer Architecture, both from Iran University of Science and Technology, Tehran, Iran, in 1992 and 1995, respectively. He received his Ph.D. degree in Computer Science from University of Ottawa, Canada, in 2006. He is the director of Computer Networks Research Lab at Sahand University. Dr. Rahbar is a senior member of the IEEE. He is currently on the editorial board of Wiley Transactions on Emerging Telecommunications Technologies Journal, International Journal of Advances in Optical Communication and Networks, and Journal of Convergence Information Technology. His main research interests are optical networks, optical packet switching, scheduling, PON, IPTV, VANET, network modeling, analysis and performance evaluation, the results of which can be found in over 80 technical papers (see http://ee.sut.ac.ir/showcvdetail.aspx?id=13).

E-mail: akbar_rahbar92@yahoo.com
ghaffarpour@sut.ac.ir
Department of Electrical Engineering
Sahand University of Technology
Sahand New Town, Tabriz, Iran

Paper

On the Influence of Network Impairments on YouTube Video Streaming

Arkadiusz Biernacki[a], Florian Metzger[b], and Kurt Tutschku[b]

[a] *Institute of Computer Science, Silesian University of Technology, Gliwice, Poland*
[b] *Chair of Future Communication, University of Vienna, Vienna, Austria*

Abstract—Video sharing services like YouTube have become very popular which consequently results in a drastic shift of the Internet traffic statistic. When transmitting video content over packet based networks, stringent quality of service (QoS) constraints must be met in order to provide the comparable level of quality to a traditional broadcast television. However, the packet transmission is influenced by delays and losses of data packets which can have devastating influence on the perceived quality of the video. Therefore, we conducted an experimental evaluation of HTTP based video transmission focusing on how they react to packet delay and loss. Through this analysis we investigated how long video playback is stalled and how often re-buffering events take place. Our analysis revealed threshold levels for the packet delay, packet losses and network throughput which should not be exceeded in order to preserve smooth video transmission.

Keywords—multimedia communication, network measurements, quality of service, video streaming.

1. Introduction

During the past years video sharing services like YouTube in the US, Smiley in Japan, the now defunct Megavideo in Hong-Kong, and Dailymotion in France have become very popular. YouTube users alone request millions of videos every day. Consequently, popularity of this kind results in a drastic shift in Internet traffic statistic, which reports that the share of P2P traffic is declining, primarily due to an increase in traffic from Web-based video sharing services [1]. So far, there is no indication that this trend will decrease and indeed is more likely to sustain. Thus, fulfilling the rising demand for video traffic will be a challenging task for both content providers as well as ISPs (Internet Service Providers).

Video streaming in the above mentioned services is either web-based or HTTP-based, therefore being transported using the TCP. The TCP is currently the most widely used transport protocol in the Internet but conventionally regarded as inappropriate for media streaming. The primary reason lies in the TCP reliability and retransmission mechanisms which can lead to undesirable transmission delays and may violate timeliness requirements for streamed live media. In this context, coping with packet delay and loss, which can occur due to congestion or packet corruption, demands new solutions as classical transmission procedures, used in unreliable protocols, e.g. UDP, may be not sufficient. It should also be taken into account that the HTTP and TCP are general purpose protocols and were not specifically designed or optimized for streaming media delivery. Thus, attempts are being made to adapt media delivery to the Internet instead of trying to adapt the Internet to multimedia content streaming.

In our work we concentrate on YouTube which represents a service that is unlike the traditional VoD systems in several important aspects. From our perspective, the most important difference between YouTube and other more traditional VoD systems is that the latter usually offer professionally-produced video content such as movies, news, sport events, or TV series. The quality and popularity of this content are well-controlled and predictable. In contrast, YouTube videos can be uploaded by anyone with access to the Internet. The quality of these video clips vary significantly making network optimizations for specific content unreasonable.

Internet connections are characterized by a number of statistically determined characteristics including latency and reliability. These traits are not guaranteed – in fact, they can fluctuate considerably depending on the local ISP network load, remote server load, background traffic, as well as network infrastructure quality. Video delivered by more traditional channels such as satellite, DVD, cable or digital TV broadcasting requires usually not to much buffering space at a client side because data arrives at a media player with mostly deterministic delay, rate and very limited or infrequent data drops. Video delivered over the Internet is much more problematic because there is no guarantee that the data will flow to a user at a sufficient rate and determined delay. Instead, it arrives with a rate and delay that can change consistently during video file transmission. Therefore, buffering is of increasing importance for video streams when they are transmitted over the Internet, including Web-based streaming. The YouTube client software manages buffering and playing of the received content using several behaviors [2].

In our work we study the efficiency of three possible streaming client playback strategies. Our goal is to investigate how often the player buffer runs out under different network interferences, especially packet loss and delay. As a consequence, we want to investigate how these parameters influence the perceived quality of video received by end users. An ISP may have partial influence on these characteristics and therefore may be able to tune the quality of video transfer and influence its users' satisfaction.

2. Video Distribution

2.1. Protocols

Contemporary media delivery systems can be classified into two categories: systems with and systems without feedback control mechanism.

One of the options for multimedia delivery systems with feedback control is an usage of the Real-Time Streaming Protocol (RTSP). The RTSP is a stateful protocol, which means that the server keeps track of the client's state from the first time the client connects to the streaming server until the time it disconnects. The client communicates its state to the server by sending commands such as play, pause or disconnect. The server begins sending the media as a steady stream of small RTP (Real-time Transport Protocol) packets. The data is sent at the media bitrate and the client buffer is filled with just few packets before playback begins. If the client or server discover any interferences in their communication, like increasing latency or packet drops, they can renegotiate transmission parameters, e.g., the server can send the same video content but with reduced encoding rate. The transmission is usually based on unreliable transport protocols, most commonly the UDP. However, when using the UDP, data packets often have difficulty getting around firewalls and network address translators. Thus, sometimes the TCP is preferred when firewalls or proxies block UDP packets, although at the expense of potentially unnecessary reliability.

Such problems are limited when using the HTTP as a media delivery protocol because firewalls and routers know how to pass HTTP traffic through. It also does not require special proxies or caches. The HTTP is a stateless protocol. Thus, multimedia transmission based on it share this feature and behave as a system without feedback control. Basically, if an HTTP client requests data, the server responds by sending the required data, but it does not remember the client or its state which means that each HTTP request is handled completely independently.

HTTP streaming may be implemented in several ways. In our work we focus on an implementation which can be described as a progressive download. The progressive download is nothing more than a transfer of a video file from a HTTP server to a client where the client may begin playback of the file before the download is complete. Contrary to the above mentioned systems with feedback control, which rarely send more than a few seconds of video content to a client in advance, HTTP streaming (web) servers progressively push the whole video content to a client, usually does not taking into account how much data have been already sent in advance. Simultaneously, most players are capable of playing the video file while its download is still in progress. Most web-based streaming platforms, including Vimeo, MySpace, and MSN Soapbox, are based on HTTP and do not have a feedback control. However, some HTTP streaming services, e.g. YouTube, implement additional application layer flow control mechanisms that limit the transmission rate to the same magnitude as the video bitrate [3].

Currently, it is thought that HTTP media streaming is easier and cheaper to deploy because web streaming can use generic HTTP solutions and does not require specialized servers at each network node. Standard HTTP caching mechanism allow to move media content to an edge of the network, closer to users. Nonetheless, the above technology have also shortcomings. The congestion avoidance algorithm of the TCP produces a saw-tooth shaped transmission rate. Furthermore, the reliability of TCP results in variable transmission delays due to retransmissions of lost packets. As a consequence, it was commonly assumed that the TCP is not suitable for multimedia streaming, which is to some extent loss tolerant but delay sensitive. The instantaneous transmission rate and transmission delay variation of the TCP must be smoothed out by receiver-side buffering. Despite these drawbacks, currently a dominant share of multimedia traffic is being delivered using the HTTP and TCP [1].

2.2. Video Buffering

Most of HTTP players are able to concurrently play and download the same file. In the simplest case the player fills its internal buffer at the beginning of the video transmission and starts the video playback as soon as a minimum buffer level is achieved. While simultaneously playing and downloading the content, the amount of video data in the buffer is variable and depends mainly on the download bandwidth, video bitrate and video playing rate. When the download bandwidth is larger than the video rate the buffer grows. In the opposite case, the buffer will shrink and if the situation last long enough it may also run out. In such cases the video stalls and the player waits until the buffer will be refilled again.

Let's assume that $G(t)$ represents the number of data packets generated at a HTTP server by time t, Fig. 1 with a packet transmission rate limited only by the infrastructure conditions like TCP throughput, server performance, etc. The first packet is generated at time 0 and sent immediately to a client. Let $A(t)$ denote the number of packets arriving at the client by time t and $B(t)$ denote the number of packets played by the client by time t. Since the transmis-

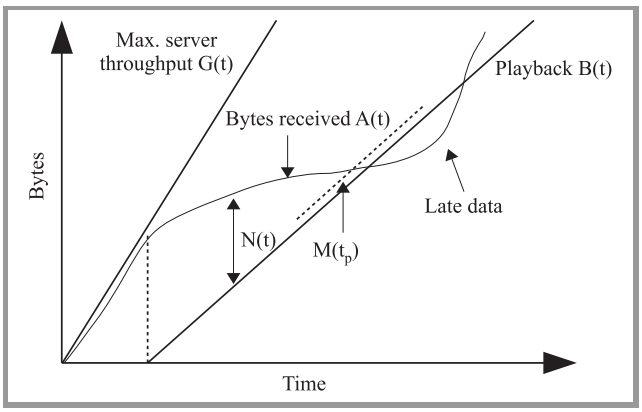


Fig. 1. Player buffer occupancy in the time function.

sion rate is constrained by the generation rate at the server, we have $A(t) \leq G(t)$. A packet arriving earlier than its playback time is referred to as an early packet. At time t, the number of early packets is counted as $N(t) = A(t) - B(t)$. A negative value of $N(t)$ indicates that the packet arrival is behind the playback by $-N(t)$ packets.
During streaming, there can be many time periods Δt_i for which $N(\Delta t_i)$ has a negative value. In our work we try to answer the question: what is the value of $\sum_i \Delta t_i$ and i, i.e. the total video stall time in a relation to video clip length and the number of stalling events for several video files transmitted from YouTube.

2.3. Video Playing Strategies

For video streaming YouTube currently uses amongst others device-dependent 3GP containers with RTSP dedicated for mobile streaming applications and Adobe Flash with HTML5 employing HTTP streaming of Flash Video. Adobe Flash, henceforth referred to as Flash, is the default container when YouTube is accessed via a PC. Users need to install a proprietary plug-in for viewing Flash videos. HTML5 supports videos that do not require any proprietary plug-ins running directly.
When streaming with the Flash Player, it basically behaves like a simple HTTP player described above i.e. it starts the video playback as soon as a minimum buffer level is achieved. However, thanks to the flexibility of the Flash authoring platform, the buffering functionality can be additionally enhanced using client-side ActionScript code. The standard buffering process is believed to be susceptible to bandwidth drops, as well as being unable to exploit a sudden increase of bandwidth. The enhancement is called a dual-threshold buffering strategy and assures a faster start and, at the same time, should provide better resilience to bandwidth fluctuations, or other adverse network conditions. Therefore, the playback of a video file starts when the first threshold in the buffer is filled with a given amount of data. But, instead of trying to keep the buffer full to this level, the modified strategy attempts to fill the buffer to a second, higher threshold. This additional data may be useful later if the network connection encounters temporary impairments like bandwidth drops or fluctuations.
In the case of HTML5 streaming, the playing strategy depends on particular video player implementation. The W3C HTML5 specification [4, Section 4.8] states, that in the case of autoplay "the user agent [...] will automatically begin playback of the media resource as soon as it can do so without stopping". To approximate this difficult to fulfil condition every implementation differs. We investigated Firefox's implementation of this spec as its code is open source and the behaviour can therefore be studied not just by observing network traces but also by reading the sources. The algorithm in the Firefox is summarized in algorithm in Fig. 2 and Table 1. Rather than using static thresholds it facilitates moving averages to estimate the development of the transmission rate. It does not differentiate between the initial video startup time and intermittent buffering events.

if $s_{MA} > v_{MA}$ **then**
    $c \leftarrow (b_b = 20s \vee b_T = 20s)$
else
    $c \leftarrow (b_b = 30s \vee b_T = 30s)$
end if

Fig. 2. Firefox playback (re-)start decision algorithm.

This implementation requires large playback buffers due to the chosen high video buffering amounts, but could also result in very few stalling events.

Table 1
Variables involved in buffering decisions

Variable	Explanation
s_{MA}	Moving average of the transmission speed.
v_{MA}	Moving average of the video bitrate.
c	Condition upon which to start/resume playback.
b_b	Amount of video data the buffer contains.
b_T	Amount of time spent in non-playing buffering state.

The HTML5 network traffic also differs from the Flash traffic. The works [5] and [2] identified YouTube's block transmission behaviour, which uses longer and client application controlled block phases for Google Chrome and no blocking at all for Firefox.

3. Previous Works

A major research area related to our work is concerned with the analysis and characterization of streaming services in the Internet. Early works in this area go back to the twentieth century and focused amongst others on the characterization of videos on the Web [6], video access statistics of users [7], developing UDP-based streaming protocols and providing mechanisms for TCP-friendliness and loss recovery, e.g. [8], [9].
When to concentrate on the HTTP video, several YouTube measurement studies have been reported in literature in the last few years. These works focused on characterizing various aspects of YouTube videos, as well as its usage patterns. On the one hand, we have work based on user traffic trace analysis including deep packet inspection, e.g. [2], [10]–[12]. Their authors operated on real world measurements obtained from, e.g., ISPs' networks and they characterized video popularity, durations, size and playback bitrate, as well as usage pattern statistics such as day versus night patterns or traffic volume. Additionally, in [10] the investigation of YouTube user sessions statistics was presented. In [2] the authors presented a traffic characterization of Netflix and YouTube, and identified different streaming strategies deriving also a model for the aggregate traffic generated by these services. Plissonneau *et al.*

in [12] described the impact of YouTube traffic on a French regional ADSL point of presence revealing that YouTube video transfers are faster and larger than other large Web transfers.

On the other hand, there are publications based on crawling the YouTube site for an extended period of time [13]–[15]. These works examined video popularity and user behaviour and found that statistics such as length, access patterns, growth trend, and active life span were quite different compared to traditional video streaming applications. Furthermore, in [13] information directly available from YouTube servers was used to analyse the characteristics of videos served by YouTube while [14] investigated social networking in YouTube videos. Also Abhari and Soraya in [15] investigated YouTube popularity distribution and access patterns through the analysis of a vast amount of data collected by crawling the YouTube API. On the basis of the observations, the authors presented essential elements of the workload generator that can be used for benchmarking caching mechanisms.

A global study of user experience for YouTube videos using PlanetLab nodes from all over the world is performed in [16]. Results from this analysis show that on average there are about 2.5 pauses per video, and on average 25% of the videos with pauses have total pause time greater than 15 seconds.

The closest work to ours is [17] where the authors evaluated the responsiveness of adaptive HTTP algorithms (taking into account YouTube amongst others) under variable network conditions. The authors claimed that the performance of the streaming algorithm increases with the decrease of network delay and by providing information to the client, particularly about the achievable throughput. It compensates for the structural noisiness of measurements and improves the ability of the client to accurately estimate the throughput.

4. Experiments

In order to observe the behaviour of YouTube, and, for that matter, any other, streaming under varying network conditions, we created a controlled, isolated environment for our tests, depicted in Fig. 3. Analyses are conducted in two phases. In phase one, Python scripts on a client computer simulate a streaming application by issuing HTTP GET requests to videos at a YouTube cache. Any video data is stored and analysed for its frame characteristics, such as the size, type and relative playback time. Furthermore, the client captures the packet trace using tcpdump/libpcap, allowing offline analysis of the packets to and from the HTTP server.

The transmission is done through a network emulation node using the built-in Linux Kernel *netem* module capable of altering the network QoS parameters, such as packet delay distribution, packet loss rate, or transmission throughput. Random packet losses are limited due to access network link layer and transport protocol retransmissions. However, through this mechanisms, loss acts as another source of

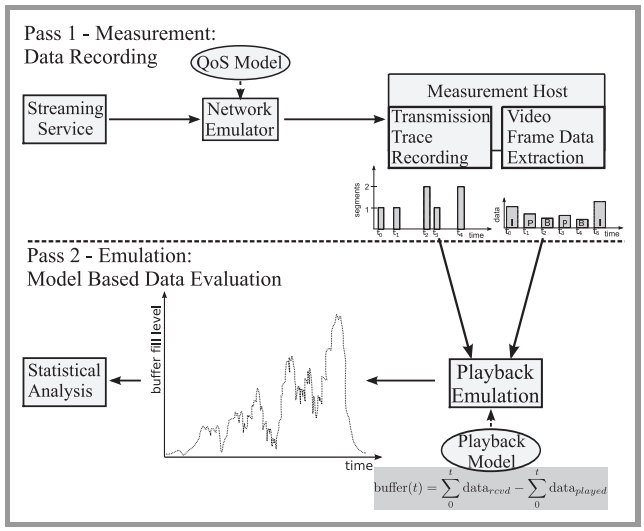


Fig. 3. Two-pass measurement environment used to capture network traces independently of playback strategies.

delay and jitter. We focus on asymmetric access networks, such as ADSL phone lines, which are assumed to form the bottleneck links.

In the second phase, the video file with the frame dataset are re-assembled. Then they are used to feed models in a media playback emulation process, which combines the transmission and video frame traces to calculate the playback buffer fill level for every point in time during the playback. Each frame has its own playing time which specifies the time at which the frame should be played in relation to the initial frame. From this analysis, we can determine how much data is required to play each frame of the video without any delay to allow for an uninterrupted playback. On this basis, we generate statistics about user-perceivable artifacts such as re-buffering events that would occur during the playback. These statistics can then be compared to the results of other models and network QoS.

For our experiment we used a 92 s video file encoded at 850 Kbit/ps.

4.1. Quality Measures

From the user's perspective, the key performance characteristic of a network is the QoS of received multimedia content. However, in the case of HTTP video the transmission is reliable, so there is no packet loss induced video degradation. Nevertheless, packet losses introduce additional delay caused by TCP retransmissions which consequently can lead to re-buffering events resulting in jerky playback. The packet delay and loss reduce also TCP throughput. When the throughput is lower than the playback rate and the buffer has drained, the video playback will pause and wait for new video data. A user expects that delays resulting from content buffering will be minimized and do not occur during normal video play.

Thus, to characterize the relationship between the network QoS and application QoS, for our purpose, we use two

measures for HTTP videos. The first measure of the application QoS takes into account relative total stalling time experienced by a user and is defined as:

$$\mathrm{SR} = \sum_i \Delta t_i / T, \tag{1}$$

where t_i are times for which $N(\Delta t_i)$ has negative value and T denotes a total duration of the video file when played without interruptions. As the above measure is the ratio of total stalling time to the the video duration, it is desirable to minimize its value by an ISP.

The application QoS defined in (1) did not differentiate between the cases in which a user can experience one long stalling period Δt^l or several shorter stalling periods Δt^s where $\Delta t^l = \sum_i \Delta t_i^s$. Thus, in our analysis we also use a second, complementary measure which value is the number of re-buffering events i associated with every stalling period.

In our experiment every video playing scenario has at least one re-buffering events which is a result of an initial buffering. The initial buffering is used to accommodate initial throughput variability or inter-packet jitters. Some streaming strategies may achieve smoother streaming with larger initial buffering, nonetheless it increases the startup latency of received video content. The re-bufferings, which take place in the middle of video playback, are usually a consequence of the congestion avoidance algorithm of the TCP.

In our analysis we compared the SR (1) and stalling frequency for the earlier mentioned buffering algorithms: Flash, HTML5 and simple buffering strategy (Simple). The last strategy assumes that the algorithm always starts playback as soon as any data is available in the buffer. This means that, if the player is currently stalling and a complete frame becomes available in the buffer, playback will immediately restart and the frame will be shown even if this means stopping playback after that frame again. This results in the lowest required buffer space. Moreover, playing the video as soon as possible, gives the fastest end. Consequently, the Simple strategy give the lowest SR and an upper limit for the number of stalls occurring. Conversely, the best way to minimize the number of stalls is to wait for the entire file to be downloaded.

5. Results

5.1. Delays

The delay in an experienced by video content consists of two components: delay introduced by network, which is the time it takes a data packet to travel from sender to receiver and TCP-level delay, which is a consequence how the TCP reacts to fluctuations in the effective network throughput. While throughput fluctuations can occur due to application-level flow control, they are primarily the result of network congestion.

As results of our experiments we obtained statistics of buffer occupancy as a function of time for the three examined playing strategies. An exemplary trace of the buffer occupancy for the Simple strategy is presented in Fig. 4.

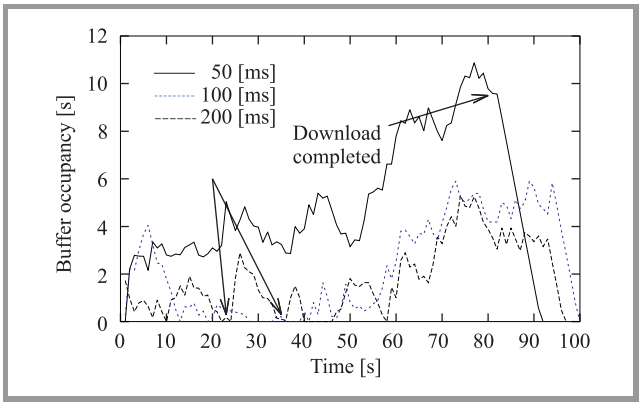


Fig. 4. Player buffer occupancy as a function of time.

We may notice that with increasing latency the buffer occupancy, measured as video playback time, is decreasing and re-buffering events happen more often. When the packet delay is 50 ms, there is only one stalling event on the beginning of the video transmission. According to the formula throughput $\sim 1/\mathrm{delay}$ describing theoretical TCP throughput, in this case the delay is the lowest, thus the theoretical connection throughput is the highest. Therefore, the download of the whole video finishes after about 80 s of the experiment. From this moment the buffer is no longer supplied. It decreases at a constant rate as the video player pulls the remaining video data and presents it to its user.

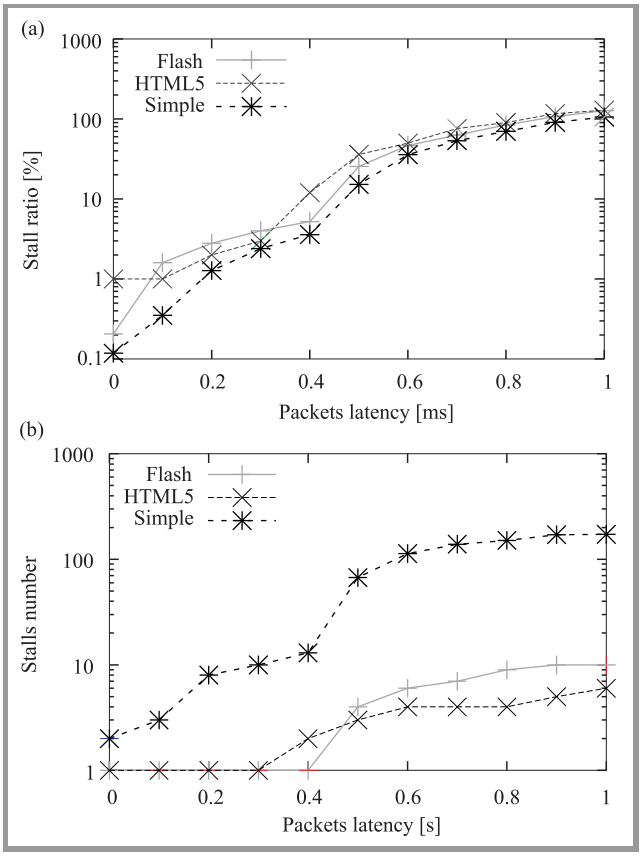


Fig. 5. The influence of packet delay: (a) stalling ratio; (b) re-buffering events.

However, when the delay rose to 100 ms or 200 ms, watching the video is quite inconvenient due to the frequent buffer under-runs. In these situations more re-buffering events occur, the total playing time exceeds the original video length and the file downloading completes after about 92 s.

Generally, we were interested not in a transient buffer analysis but in its examination in the context of application QoS for which measures were defined in the Subsection 4.1. Thus, in the further experiments, except for the packet delay, we obtained statistics of the buffering behavior in scenarios with additional packet loss and network throughput limitation.

As it is shown in Fig. 5(a), packet delay has a certain influence on application QoS which is defined as the SR in Eq. (1). Increasing gradually the packet delay up to 1000 ms caused the successive rise of the SR from less than 1% to about 100% in average. From the three examined playback strategies, the Simple strategy experienced the lowest SR while the highest SR was obtained by Firefox HTML5 strategy. Nonetheless, with increasing delay, the differences between the playing algorithms diminished.

When it comes to measuring the number of stalls, the situation looks quite different. When increasing latency up to 300 ms, a user using the Flash or HTML5 strategies usually experienced only a single re-buffering event which occurred at the beginning of the playback. When the delay exceed 500 ms the HTML5 strategy has the lowest number of stalls from all the three examined strategies. The Simple strategy was not able to successfully mitigate the network impairments which resulted in several re-buffering events during the playback, Fig. 5(b).

5.2. Packet Loss

Packet loss can be caused by a number of factors including signal degradation over the network medium due to multi-path fading, packet drop because of channel congestion, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines. When transmitting HTTP video, in the event of packet loss, the receiver asks for retransmission or the sender automatically resends any segments that have not been acknowledged. Nevertheless, retransmitting missing packets causes the throughput of the connection to decrease due to the sliding window mechanism used for acknowledgement of received packets, implemented in the TCP.

For the packet loss up to 2% the SR graph resembles *S* shape, Fig. 6(a). For packet loss below 0.8% the SR has value 1 for the Flash and HTML5 strategies, and about 0.2 for the Simple strategy. Such values of the SR can be considered relatively low and should not have much impact on the received video quality. However, for the packet loss between 1% and 1.2% the SR rises rapidly achieving values of several tens. The further increase of the packet loss rate results in a relatively small rise of the SR. Generally, the Simple strategy has the lowest value of the SR. We can also notice that for 1% packet loss there is a significant difference between the Flash and HTML5 strategies which diminishes for the other packet loss values.

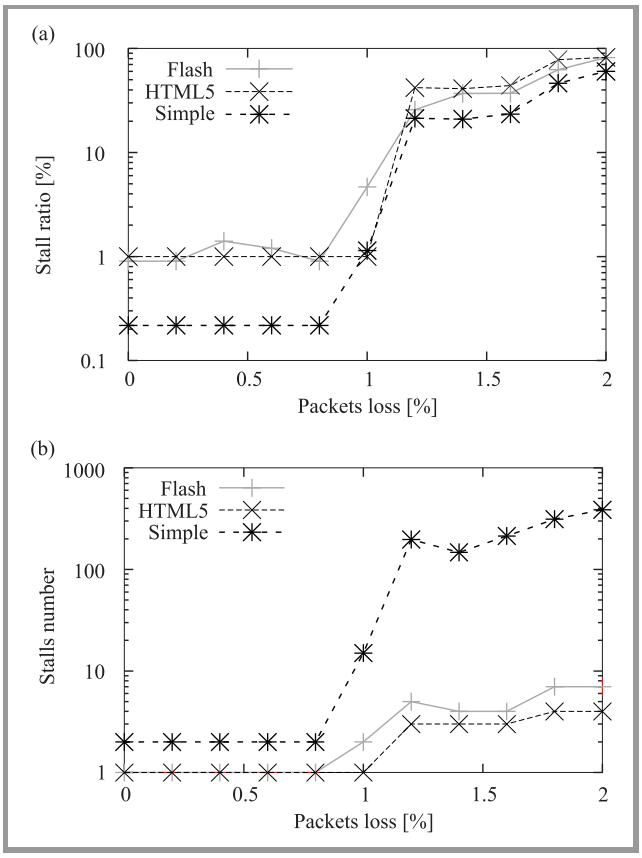


Fig. 6. The influence of packet loss: (a) stalling ratio; (b) re-buffering events.

When to measure the video streaming resilience against the packet loss as number of stalls, Fig. 6(b), the shape of the chart is similar to the shape of the SR presented in the Fig. 6(a). We can also observe here the steady rise of the stalls number when the packet loss is lower than 0.8% and higher than 1.2%. For the packet loss value between 0.8% and 1.2% the stalls number grow quite fast. Contrary to the results presented in the Fig. 6(b), this time the Simple strategy has the worst performance. The HTML5 strategy is little better than the Flash strategy for the packet loss higher than 1%.

5.3. Throughput

In this section we investigate how the download throughput limitation influence the YouTube video streaming. The upload throughput in the experiments was fixed and set to 10 Mbit/s. Figure 7(a) shows the dependency between the SR and download throughput ranging from 256 Kbit/s up to 10 Mbit/s. We can observe that the 1 Mbit/s download throughput is sufficient for our streamed video independently of the playing strategy used. Increasing the throughput beyond 1 Mbit/s does not significantly improve the SR. From the other side, even a small throttle of the network throughput results in a dramatic rise of the

SR value. Taking into account that the video encoding rate is 850 Kbit/s, such streaming behavior is common sense and the network throughput below this threshold value should be insufficient. Furthermore, the mentioned threshold will be additionally increased by network protocols overhead.

The similar situation is when we used the stalls number measure, Fig. 7(b). For the 1 Mbit/s network throughput and and above the number of stalls is 1 for the Flash and HTML5 strategies. The Simple playing strategy experiences two re-buffering events.

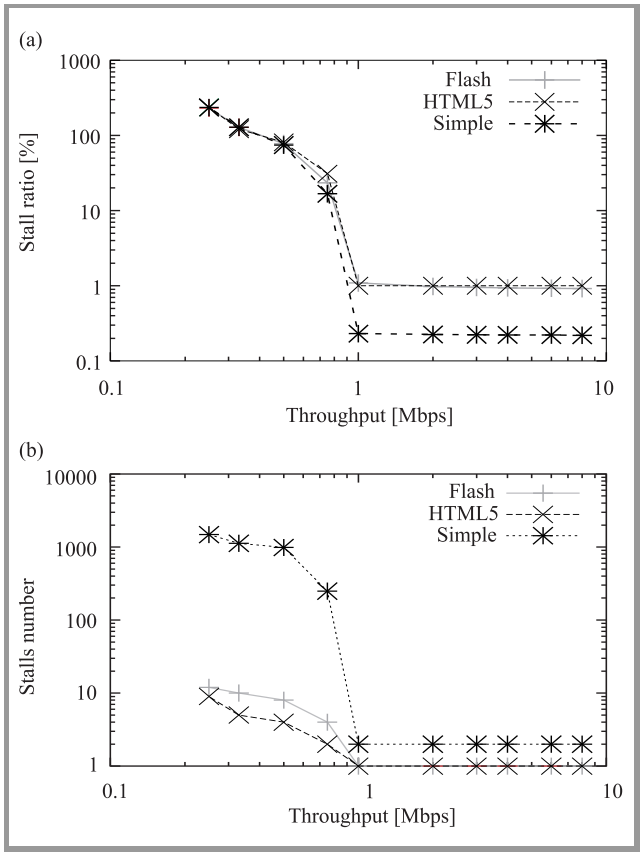


Fig. 7. The influence of throughput limitation: (a) stalling ratio; (b) re-buffering events.

Generally, we can conclude that the all three playing strategies cannot satisfactorily cope with even small limitation in the network throughput, and the initial buffering implemented by the Flash and HTML5 strategies fails in this situation.

6. Conclusions

In the paper we tried to answer how network defects, manifested as latency, packet loss and throughput limitations impacts the quality of HTTP based video playback. For this purpose, we conducted an experimental evaluation of YouTube video transmission examining the quality of experience of end user applications expressed as a function of playback buffer occupancy. Through this analysis we investigated how long the video playback is stalled and how often re-buffering events take place.

Generally, in order to watch the 850 Kbit/s video without interruptions and extensive buffering time, the packet delay introduced by the network should not exceed 200 ms. The packet loss higher than the 0.8% makes the viewing of online video very inconvenient. For smooth transmission of the video, network connection throughput should be at least 1 Mbit/s.

Our analysis revealed that there exists some small differences between the Flash and HTML5 strategies, however, in the most cases they will remain unnoticed by the end user. The buffering algorithm used in an HTML5 player showed the highest resilience against the packet delay and loss when taking into account the number of re-buffering events experienced during the video play. However, when comparing these both strategies with the Simple strategy, it is obvious that starting the video playback as soon as a minimum buffer level is achieved is insufficient. Although the Simple strategy has lower stalling time compared to the other two strategies, nonetheless, the number of re-buffering events which occur during the video streaming in unacceptable for an end user. All the three playing strategies cannot satisfactorily cope with even small limitation in the network throughput. The initial buffering mechanism implemented by the Flash and HTML5 strategies fails in that situation.

Acknowledgment

The research was partially supported by the National Science Centre (Poland) under grant DEC-2011/01/D/ST6/06995.

References

[1] "Global mobile data traffic forecast update, 2010002015", *Cisco Visual Networking Index. White Paper*, Cisco, 2011.

[2] A. Rao, Y. S. Lim, C. Barakat, A. Legout, D. Towsley, and W. Dabbous, "Network characteristics of video streaming traffic", in *Proc. 7th Int. Conf. Emerg. Netw. Exper. Technol. CoNEXT 2011*, Tokyo, Japan, 2011.

[3] S. Alcock and R. Nelson, "Application flow control in YouTube video streams". *ACM SIGCOMM Comp. Commun. Rev.*, vol. 41, no. 2, pp. 24–30, 2011.

[4] I. Hickson, "HTML5: a vocabulary and associated APIs for HTML and XHTML', April 2010 [Online]. Available: http://www. w3.org/TR/html5/

[5] A. Finamore, M. Mellia, M. Munafo, R. Torres, and S. R. Rao, "YouTube everywhere: impact of device and infrastructure synergies on user experience", Tech. Rep. 418, Purdue University, May 2011.

[6] S. Acharya and B. C Smith, "Experiment to characterize videos stored on the web", in *Proc. ACM/SPIE Multimedia Comput. Netw. MMCN 1997*, vol. 3310, pp. 166–178, 1997.

[7] S. Acharya, B. Smith, and P. Parns,"Characterizing user access to video on the world wide web", in *Proc. ACM/SPIE Multimedia Comput. Netw. MMCN 2000*, vol. 3969, pp. 130–141, 2000.

[8] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the internet", SIGCOMM Comput. Commun. Rev., vol. 29, no. 4, pp. 189–200, 1999.

[9] S. Floyd, M. Handley, J. Padhye, and JÃ¶rg Widmer, "Equation-based congestion control for unicast applications", *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 43–56, 2000.

[10] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge". in *Proc. 7th ACM SIGCOMM Conf. Internet Measur. IMC 2007*, San Diego, CA, USA, 2007, pp. 15–28.

[11] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of YouTube network traffic at a campus network-measurements, models, and implications", *Computer Netw.*, vol. 53, no. 4, pp. 501–514, 2009.

[12] L. Plissonneau, T. En-Najjary, and G. Urvoy-Keller, "Revisiting web traffic from a DSL provider perspective: the case of YouTube", in *Proc. ITC Spec. Seminar Netw. Usage Traffic*, Berlin, Germany, 2008.

[13] M. Cha, H. Kwak, P. Rodriguez, Y. Y Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system", in *Proc. 7th ACM SIGCOMM Conf. Internet Measur. IMC 2007*, San Diego, CA, USA, 2007, pp. 1–14, 2007.

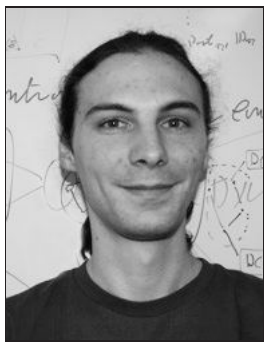
[14] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of YouTube videos", in *Proc. 16th Int. Worksh. Quality of Service IWQoS 2008*, Enschede, The Netherlands, 2008, pp. 229–238.

[15] A. Abhari and M. Soraya, "Workload generation for YouTube", *Multimedia Tools and Appl.*, vol. 46, no. 1, pp. 91–118, 2010.

[16] D. K. Krishnappa, S. Khemmarat, and M. Zink, "Planet YouTube: global, measurement-based performance analysis of viewer;'s experience watching user generated videos", in *Proc. IEEE 36th Conf. Local Comp. Netw. LCN 2011*, Bonn, Germany, 2011, pp. 948–956.

[17] S. Benno, J. O. Esteban, and I. Rimac, "Adaptive streaming: the network HAS to help", *Bell Labs Tech. J.*, vol. 16, no. 2, pp. 101–114, 2011.

Arkadiusz Biernacki received the M.Sc. and Ph.D. degree in Computer Science from the Silesian University of Technology, Poland, in 2002 and 2007, respectively. From 2007 he is an Assistant Professor at the Silesian University of Technology. From 2010 he has been collaborating with Chair of "Future Communication" (endowed by Telekom Austria) at the University of Vienna. His research interests focus on network traffic modeling and computer system simulations.
E-mail: arkadiusz.biernacki@polsl.pl
Institute of Computer Science
Silesian University of Technology
Akademicka 16
44-100 Gliwice, Poland



Florian Metzger is a research assistant at the the Chair of "Future Communication" (endowed by Telekom Austria) at the University of Vienna. He received his diploma thesis at the University of Wuerzburg, Germany, in 2009. His research interests cover signaling load in mobile networks as well as modern video streaming approaches.
E-mail: florian.metzger@univie.ac.at
Chair of Future Communication
University of Vienna
Währinger Str. 29/5.46
1090 Vienna, Austria

Kurt Tutschku holds the Chair of "Future Communication" (endowed by Telekom Austria) at the University of Vienna. Before that, he was an Assistant Professor at the Department of Distributed Systems, University of Wuerzburg. He led the department's group on Future Network Architectures and Network Management until December 2007. From February 2008 to July 2008, he worked as an Expert Researcher at the NICT (National Institute for Information and Communication Technology, Japan). He has received a doctoral degree in Computer Science from University of Wuerzburg in 1999 and completed his habilitation at the University of Wuerzburg in 2008. His main research interest include future generation communication networks, Quality-of-Experience, and the modeling and performance evaluation of future network control mechanisms and P2P overlay networks.
E-mail: kurt.tuschku@univie.ac.at
Chair of Future Communication
University of Vienna
Währinger Str. 29/5.38
1090 Vienna, Austria

Paper

Optimal Pump Scheduling for Large Scale Water Transmission System by Linear Programming

Jacek Błaszczyk[a], Andrzej Karbowski[a,b], Kamil Krawczyk[a], Krzysztof Malinowski[a,b], and Alnoor Allidina[c]

[a] *Research and Academic Computer Network (NASK), Warsaw, Poland*
[b] *Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland*
[c] *IBI-MAAK Inc., Richmond Hill, Ontario, Canada*

Abstract—Large scale potable water transmission system considered in this paper is the Toronto Water System, one of the largest potable water supply networks in North America. The main objective of the ongoing Transmission Operations Optimizer project consists in developing an advanced tool for providing such pumping schedules for 153 pumps, that all quantitative requirements with respect to the system operation are met, while the energy costs are minimized. We describe here a linear, so-called Simplified Model (SM), based on mass-balance equations, which is solved on week horizon and delivers boundary conditions for so-called Full Model (FM), which is nonlinear and takes into account hydraulic phenomena and water quality.

Keywords—linear programming, minimum cost operative planning, pump scheduling, water supply.

1. Introduction

Toronto Water System (TWS) delivering water to 4 million people is the largest potable water supply network in Canada and the fifth largest in North America. It includes the whole City of Toronto (COT) and southern portion of the Region of York (ROY). TWS is supplied by 4 water filtration plants located at the north shore of Lake Ontario, and additionally by a number of wells at southern part of ROY. The average daily water demand from TWS is 2500 ML, while the total storage of reservoirs 2200 ML. It has 1300 km of pipelines, 153 pumps in 29 pumping stations, 19 pressure districts, 28 reservoirs and elevated tanks (many with two or more cells). The annual cost of water pumping is about 36 millions CAD (data from 2007). Since the electrical tariffs and costs structure are very volatile and unstable (changes are from hour to hour, and even at 15-minutes intervals), there is a need for an automatic control system of the network, reacting in a proper way to both the changes in customers demands and the market energy prices.

The main objective of the ongoing Transmission Operations Optimizer (TOO) project consists in developing an advanced tool for providing such pumping schedules for 153 TWS pumps that all quantitative requirements with respect to the system operation are met, while the energy costs are minimized [5]. It is assumed that TOO should produce detailed optimal schedules for all pumps which will be further passed to water transmission system by a Supervisory Control And Data Acquisition (SCADA) module.

The following modules of TOO has been developed: demand forecasting module, energy rates forecasting module, pumping schedule optimizer and, finally, an assessment module consisting mainly of hydraulic, EPANET based, TWS simulator (Fig. 1).

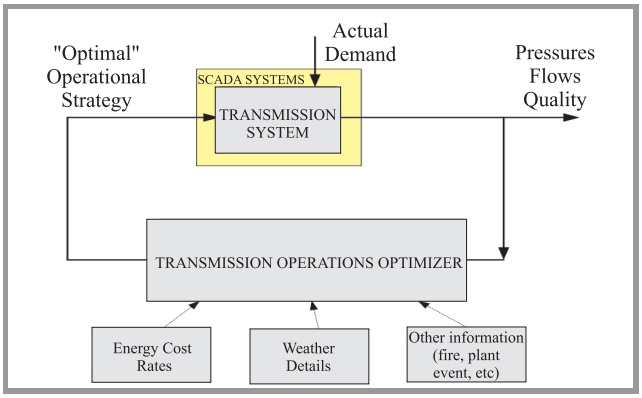


Fig. 1. TOO functionality.

This paper presents one of the key components of TOO, namely, the Simplified Model (SM), based on the solution of a linear programming (LP) problem with complex objective function expressing the cost of electrical energy consumption at pumping stations.

This component of TOO uses water distribution model based on mass-balance equations for pressure district volumes consisting of aggregated volumes of reservoirs and elevated tanks. For the 7-day control horizon with one-hour discretization and aggregation of pump's flows at pumping stations, the resulting LP problem is solved. After that, the optimal aggregated flows at pumping stations are disaggregated by a scheduler into individual pump flows and their start/stop times.

The solution obtained from SM is supposed to deliver terminal conditions for the precise, based on hydraulic de-

pendencies, Full Model (FM) at 24th or 48th hour of the control horizon, and the reference control trajectories for those obtained from the FM.
In the paper, a part of the overall system was studied. This included all the facilities within the City of Toronto which account about 97% of the energy used. The Region of York facilities (3% energy use) were not included. One-hour and 15-minute intervals were used as the discretization step. The computations were started with reservoir levels at 95%, and at the end of the 7-day period they were taken back to 95% of the total capacity (or higher).

2. Optimization Problem Formulations for the SM

2.1. Optimization Problem 1

A simplified discrete-time system model (mass-balance) for the entire system can be written as:

$$V_i(k+1) = V_i(k) + \sum_{j=1}^{N_{\mathrm{PS}}} b_{ij}\frac{T}{24}u_j(k) - d_i(k),$$
$$i = 1,2,\ldots,N_D, \quad k = 0,1,\ldots,N-1, \tag{1}$$

where:

- N_D – number of pressure districts,
- N_{PS} – number of pumping stations,
- $V_i(k)$ – volume stored in district i at time k (in ML),
- $b_{ij} = c_{ij}Q_j$ – matrix (units are ML/D, Firm Capacity matrix),
- c_{ij} – 1 if pumping station j is pumping into district i,
  -1 if pumping station j is pumping out of district i,
  0 otherwise,
- $Q_j = \sum_{m=1}^{N_{jp}} Q_{jm}$ – total station capacity (in ML/D),
- Q_{jm} – pump capacity for the m-th pump at the j-th pumping station,
- $u_j(k)$ – accumulated "control vector" at time k, $u_j(k) \in [0,1]$ (a continuous variable),
- $d_i(k)$ – demand/consumption from the i-th pressure district in period between k and $k+1$ time instant (in ML),
- T – time interval (typically one-hour),
- N – number of time intervals (= 168 for 7-days and $T = 1$ hour).

We must take into account time varying minimum and maximum reservoir levels:

$$V_{i,\min}(k) \le V_i(k) \le V_{i,\max}(k), \tag{2}$$

where $V_{i,\min}(k)$ and $V_{i,\max}(k)$ are the minimum and maximum storage volumes specified (typically these will be constants with respect to time k).

The total cost is the sum of pumping stations energy cost and water production cost:

$$J_{\mathrm{TOTAL}} = J_{\mathrm{STATIONS}} + J_{\mathrm{PLANTS}}, \tag{3}$$

where J_{STATIONS} is total cost for all stations:

$$J_{\mathrm{STATIONS}} = \sum_{j=1}^{N_{\mathrm{PS}}} J_j, \tag{4}$$

and total cost for a week (7 days) for station j is:

$$\begin{aligned} J_j = \sum_{k=0}^{N-1} \mathrm{CC}_j(k) &+ (\mathrm{DCR}_j - \mathrm{TAR}_j)\,\mathrm{MaxKVA}_j \\ &+ \mathrm{TCNR}_j\,\mathrm{PeakKW}_j + \mathrm{TCCR}_j\,\mathrm{MaxKW}_j \\ &+ \mathrm{DRCR}_j\,\mathrm{PKWHtotal}_j \\ &+ \mathrm{WOCR}_j\,\mathrm{LFactor}\,\mathrm{PKWHtotal}_j, \end{aligned} \tag{5}$$

where:

- CC_j – Commodity Charge, per kWh; flat or increasing block tariffs charge,
- DCR_j – Distribution Charge, per maximum KVA through the week,
- TAR_j – Transmission Allowance, per maximum KVA through the week,
- TCNR_j – Transmission Charge – Network, per maximum kW from 7:00 a.m. to 7:00 p.m. weekdays (referred to as "peak kW"), through the week,
- TCCR_j – Transmission Charge – Connection, per maximum kW from 7:00 p.m. to 7:00 a.m., through the week,
- DRCR_j – Debt Retirement Charge, per kWh in the week,
- WOCR_j – Wholesale Operation Charge, per kWh in the week; cost is multiplied by a loss factor (eg., 1.0376),

and

$$\mathrm{PKWHtotal}_j = \sum_{k=0}^{N-1} \mathrm{PKWH}_j(k) \tag{6}$$
$$\mathrm{PKWH}_j(k) = \mathrm{PKW}_j(k)\,T \tag{7}$$
$$\mathrm{PKW}_j(k) = P_j u_j(k) \tag{8}$$
$$P_j = \sum_{m=1}^{N_{jp}} \mathrm{PRATING}_{jm}, \tag{9}$$

where $\mathrm{PKW}_j(k)$ is used power (kW) at station j at time k, N_{jp} is the number of pumps at the j-th pumping station and $\mathrm{PRATING}_{jm}$ is the pump power rating for the m-th pump at the j-th pumping station.
Maximum KVA through the week is:

$$\mathrm{MaxKVA}_j = \max\left\{\mathrm{PKVA}_j(k)\right\}_{k=0}^{N-1} \tag{10}$$

$$\mathrm{PKVA}_j(k) = \frac{\mathrm{PKW}_j(k)}{\mathrm{PF}_j}, \tag{11}$$

where PF_j is the power factor for the j-th pumping station (eg., 0.92).

Peak KW through the week is:

$$\text{PeakKW}_j = \max\left\{\text{PKW}_j(k),\ k=7\text{ a.m. to 7 p.m. weekdays}\right\}_{k=0}^{N-1} \quad (12)$$

Maximum KW through the week is:

$$\text{MaxKW}_j = \max\left\{\text{PKW}_j(k),\ k=7\text{ p.m. to 7 a.m. weekdays}\right\}_{k=0}^{N-1} \quad (13)$$

The cost function (5) depends on the maximum values over the time period of optimization:

$$J_{\text{MAX},j} = (\text{DCR}_j - \text{TAR}_j)\,\text{MaxKVA}_j + \text{TCNR}_j\,\text{PeakKW}_j + \text{TCCR}_j\,\text{MaxKW}_j \quad (14)$$

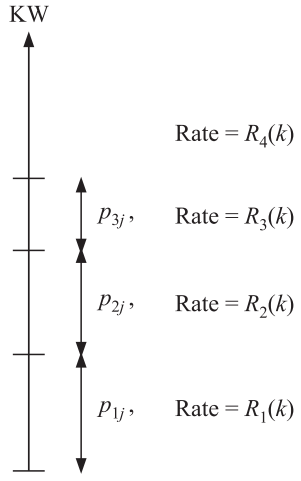
The above component can be converted into a conventional linear programming form by introducing auxiliary variables z_{1j}, z_{2j} and z_{3j} to represent peak factors. We express the transformed model as

$$\bar{J}_{\text{MAX},j} = (\text{DCR}_j - \text{TAR}_j)\,z_{1j} + \text{TCNR}_j\,z_{2j} + \text{TCCR}_j\,z_{3j} \quad (15)$$

subject to constraints:

$$\begin{aligned} &\text{PKVA}_j(k) \le z_{1j}, && k = 0,\ldots,N-1 \\ &\text{PKW}_j(k) \le z_{2j}, && k = 7\text{ a.m. to 7 p.m. weekdays} \\ &&& \text{and } k = 0,\ldots,N-1 \\ &\text{PKW}_j(k) \le z_{3j}, && k = 7\text{ p.m. to 7 a.m. weekdays} \\ &&& \text{and } k = 0,\ldots,N-1 \end{aligned} \quad (16)$$

The commodity charge (CC_j) is variable, dependent on the time of a day and the rate structure. It is assumed that there are a maximum of four blocks for the cost rates, as depicted below:



If $\text{PKW}_j(k) < p_{1j}$

$$\text{CC}_j(k) = R_1(k)\,\text{PKWH}_j(k) \quad (17)$$

Else if $\text{PKW}_j(k) < (p_{1j} + p_{2j})$

$$\text{CC}_j(k) = (R_1(k) - R_2(k))p_{1j}T + R_2(k)\,\text{PKWH}_j(k) \quad (18)$$

Else if $\text{PKW}_j(k) < (p_{1j} + p_{2j} + p_{3j})$

$$\text{CC}_j(k) = (R_1(k) - R_3(k))p_{1j}T + (R_2(k) - R_3(k))p_{2j}T + R_3(k)\,\text{PKWH}_j(k) \quad (19)$$

Else

$$\text{CC}_j(k) = (R_1(k) - R_4(k))p_{1j}T + (R_2(k) - R_4(k))p_{2j}T + (R_3(k) - R_4(k))p_{3j}T + R_4(k)\,\text{PKWH}_j(k) \quad (20)$$

The $\text{CC}_j(k)$ can be modeled as a piecewise-linear function, and because costs are non-decreasing, i.e., $R_1(k) \le R_2(k) \le R_3(k) \le R_4(k)$, it is also convex, so as a result of total modeling, we obtain large-scale linear programming model (without binary variables).

J_{PLANTS} is the cost of producing water at the four water treatment plants:

$$\begin{aligned} J_{\text{PLANTS}} = {} & \lambda_{\text{RLC}} \cdot \text{VOLtotal}_{\text{R.L.CLARK}} \\ & + \lambda_{\text{RCH}} \cdot \text{VOLtotal}_{\text{R.C.HARRIS}} \\ & + \lambda_{\text{FJH}} \cdot \text{VOLtotal}_{\text{F.J.HORGAN}} \\ & + \lambda_{\text{ISLAND}} \cdot \text{VOLtotal}_{\text{ISLAND}}, \end{aligned} \quad (21)$$

where λ_s are the production costs in \$/ML for the respective plants. The total volume from the plants is formulated in the following way, e.g.:

$$\text{VOLtotal}_{\text{R.L.CLARK}} = \sum_{\text{k=0}}^{\text{N-1}} \text{VOL}_{\text{R.L.CLARK}}(k), \quad (22)$$

and

$$\text{VOL}_{\text{R.L.CLARK}}(k) = \text{Flow}_{\text{R.L.CLARK}}(k) \cdot T \quad (23)$$

Each of the pumping station flows can be expressed at time k as:

$$\text{Flow}_j = Q_j \frac{T}{24} u_j(k). \quad (24)$$

Optimization goal is to find $u_j(k), k = 0,1,\ldots,N-1$ to minimize J_{TOTAL} Eq. (3) subject to mass-balance equations (1), bounds (2) and $0 \le u_j(k) \le 1$.

2.2. Optimization Problem 2

This optimization problem is similar to Optimization Problem 1 with the only difference that we have individual control variables for each pump $u_{jm}(k)$, $0 \le u_{jm}(k) \le 1$, for $j = 1,\ldots,N_{\text{PS}}$, $m = 1,\ldots,N_{jp}$, and $k = 0,1,\ldots,N-1$.

3. Disaggregation in Optimization Problem 1

Our disaggregation problem which is solved at every stage (every hour or every quarter of an hour; for simplicity we will omit the time index k) and for every pumping station can be described as follows:

$$\min_{u^j} \sum_{m=1}^{N_{jp}} PRATING_{jm}\, u_{jm} \tag{25}$$

$$\sum_{m=1}^{N_{jp}} \eta_{jm} Q_{jm} u_{jm} = \hat{Q}_j, \tag{26}$$

$$u_{jm} \in [0,1],\ \forall m \tag{27}$$

where:

- u_{jm} – individual pumping as a continuous variable (% of the interval T when the pump is ON) for the m-th pump at the j-th PS,
- $u^j = (u_{j1}, u_{j2}, \ldots, u_{j\overline{\overline{PS_j}}})$ – vector of all pumpings at the j-th PS,
- η_{jm} – the efficiency of the m-th pump at the j-th PS,
- $\hat{Q}_j = Q_j \frac{T}{24} \hat{u}_j$ – the desired flow of the j-th PS; it results from the solution of the Optimization Problem 1.

The cost of the energy in Eq. (25) is proportional to the power used. Because price is the same for all pumps, it is proportional to the sum of the power used by all pumps.

The number of these LP problems is not bigger than NN_{PS}. Probably, there would be much less of them, because we omit these PS-es for which $\hat{Q}_j$ equal zero (then automatically all $\hat{u}_{jm}$ equal zero too).

We will disaggregate control in such a way that we will get at every stage (of the length T) the minimal power used.

Let us replace now the components $\eta Q u$ with the new variables y:

$$y_{jm} = \eta_{jm} Q_{jm}\ u_{jm} \tag{28}$$

Hence:

$$u_{jm} = y_{jm} / (Q_{jm} \eta_{jm}) \tag{29}$$

Let us denote:

$$\alpha_{jm} = PRATING_{jm} / (Q_{jm} \eta_{jm}) \tag{30}$$

In the new variables we will have the problem:

$$\min_{y} \sum_{m=1}^{N_{jp}} \alpha_{jm} y_{jm} \tag{31}$$

$$\sum_{m=1}^{N_{jp}} y_{jm} = \hat{Q}_j, \tag{32}$$

$$y_{jm} \in [0, \eta_{jm} Q_{jm}], \forall j \tag{33}$$

This is nothing, but an auction problem.

We can get the optimal solution by sorting (before the optimization) for every PS the elements α_{jm} from the smallest to the largest and allocate the maximum, that is $y_{jm} = \eta_{jm} Q_{jm}$ (=the pump is ON over the whole stage) until their sum reaches $\hat{Q}_j$. The last element before reaching $\hat{Q}_j$ will be usually smaller then $\eta_{jm} Q_{jm}$ (this pump will be ON over a fraction of T), the remaining pumps (with the larger coefficients α_{jm}) will be OFF during the given stage.

4. Numerical Results

The full 7-day model with discrete variables for pump switches was intractable in reasonable time period (the obtained computation times for 2-day subproblems were much longer than 5-minute time limit assumed for TOO), for two popular commercial mixed-linear optimizers: CPLEX, Xpress-MP and one mixed-nonlinear optimizer: MINLPBB, so we decided to use continuous control variables $u_{jm}(k)$ from interval $[0,1]$ for each pump individually.

We solved Optimization Problems 1 and 2 (linear) for flat and increasing energy tariffs, for full 7-day optimization horizon with a one-hour and 15-minute intervals. The cost of optimized operations, problem statistics, and solution times are summarized in Table 1. For the optimization we used commercial Xpress-MP solver (version 2008A, on evaluation license, 64-bit Linux binary) with options `barrier, barthreads=4, mipthreads=4` (multithreaded mode).

It is seen from the Table 1 that, quite surprisingly, the value of costs in both flat and increasing block energy tariffs case does not depend on the time discretization. Hence, there is no motivation to use time step equal 15 minutes instead of 1 hour. The cost of suboptimal, aggregated solution is not more than 2% higher than that of the optimal one, so the approach presented as Optimization Problem 1 is rather acceptable.

5. Conclusions and Future Work

We have implemented a simplified mass-balance based model for COT. The resulting continuous LP problem is solved very fast by both commercial and free LP solvers. The results obtained for the aggregation of pumping variables case are satisfactory. However, there are some doubts about applicability of the SM, because it neglects hydraulic phenomena in the network, such as flows and head-losses in pipes and valves, dynamics of individual reservoirs and elevated tanks, pumpage and efficiency curves of pumps, continuity laws for junctions, etc. Owing to this, the next step will be the full hydraulic model of the system. The current solvers allow for solving such problems on only shorter horizon – 24 or 48 hour long. The presented

Table 1
Optimization results with Xpress-MP 2008A solver for problems with one-hour (1h) or 15-minute (15m) intervals, and flat (F) or increasing (I) block energy tariffs (n – number of linear variables, m – number of linear constraints, $J_{\text{STATIONS}} = J_{\text{CC}} + J_{\text{OTHER}} + J_{\text{MAX}}$, T – optimization time)

Problem	n	m	J_{TOTAL} [\$]	J_{CC} [\$]	J_{OTHER} [\$]	J_{MAX} [\$]	J_{PLANTS} [\$]	T [sec]
OP1-F-1h	8651	9528	528590	219980	61070	244569	2970	1
OP2-F-1h	23603	9528	521747	214514	60151	244111	2970	1
OP1-I-1h	12179	10872	539119	226849	61960	247340	2970	1
OP2-I-1h	27131	10872	529622	222655	60866	243129	2970	2
OP1-F-15m	34307	38001	528590	219980	61070	244569	2970	9
OP2-F-15m	94115	38001	521747	214514	60151	244111	2970	19
OP1-I-15m	48356	43353	539119	226849	61960	247340	2970	8
OP2-I-15m	108164	43353	529622	222655	60866	243129	2970	10

SM model will be used to deliver for this future FM terminal conditions, as well as the reference control and state trajectories.

References

[1] J. Błaszczyk, A. Karbowski, K. Krawczyk, and K. Malinowski, "NASK w Toronto", *Biuletyn NASK*, no. 3, pp. 30–33, 2009 (in Polish).

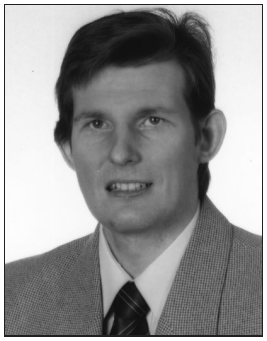
[2] J. Błaszczyk, K. Malinowski, and A. Allidina, "Aggregated pumping station operation planning problem (APSOP) for large scale water transmission system", in *Proc. PARA'2010, State-of-the-art in Scientific Computing*, *Lecture Notes in Computer Science*, K. Jonasson, Ed., vol. 7133, pp. 260–269. Berlin-Heidelberg: Springer, 2012.

[3] M. A. Brdys and B. Ulanicki, *Operational Control of Water Systems: Structures, algorithms and applications*. New York: Prentice Hall, 1994.

[4] L. W. Mays, *Optimal Control of Hydrosystems*, 1st edit. New York: Marcel Dekker, 1997.

[5] T. M. Walski *et al.*, *Advanced Water Distribution Modeling and Management*, 1st edit. Haestad Methods Solution Center, Waterbury: Haestead Press, 2003.

[6] L. A. Rossman, "EPANET 2 users manual", Tech. Rep. EPA/600/R-00/057, U.S. States Environmental Protection Agency, National Risk Management Research Laboratory, Office of Research and Development, Cincinnati, Ohio, USA, 2000.



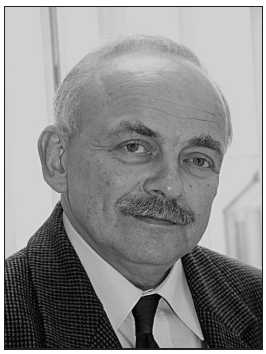
Jacek Błaszczyk received his M.Sc. and Ph.D. degrees in Automatic Control from the Warsaw University of Technology, Poland, in 2000 and 2008, respectively. Currently, he is an Assistant Professor at the Research and Academic Computer Network (NASK). His research interest include large-scale nonlinear optimization, optimal control, parallel and distributed computations, numerical software for optimization and linear algebra, and recently, modeling, simulation and optimization of large-scale water distribution systems.
E-mail: jacek.blaszczyk@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Andrzej Karbowski received the M.Sc. degree in Electronic Engineering (specialization automatic control) from Warsaw University of Technology (Faculty of Electronics) in 1983. He received the Ph.D. in Automatic Control and Robotics, in 1990. He works as adjunct both at Research and Academic Computer Network (NASK) and at the Faculty of Electronics and Information Technology (at the Institute of Control and Computation Engineering) of Warsaw University of Technology. His research interests concentrate on data networks management, optimal control in risk conditions, decomposition and parallel implementation of numerical algorithms.
E-mail: A.Karbowski@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Kamil Krawczyk received his B.Sc. from Warsaw University of Technology, Poland, in 2009, and has been working for Research and Academic Network (NASK), Warsaw, Poland, since.

E-mail: k.krawczyk.1@stud.elka.pw.edu.pl
E-mail: kamil.krawczyk@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland



Krzysztof Malinowski Prof. of Techn. Sciences, D.Sc., Ph.D., MEng., Professor of control and information engineering at Warsaw University of Technology, Head of the Control and Systems Division. Once holding the position of Director for Research of NASK, and next the position of NASK CEO. Author or co-author of four books and over 150 journal and conference papers. For many years he was involved in research on hierarchical control and management methods. He was a visiting professor at the University of Minnesota; next he served as a consultant to the Decision Technologies Group of UMIST in Manchester (UK). Prof. K. Malinowski is also a member of the Polish Academy of Sciences.

E-mail: K.Malinowski@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

E-mail: Krzysztof.Malinowski@nask.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Alnoor Allidina received his B.Sc. (Hons) degree in Electrical and Electronic Engineering in 1977, and M.Sc. and Ph.D. degrees in 1978 and 1981 respectively, from the University of Manchester Institute of Science and Technology (UMIST), Manchester, UK. He held a tenured position with UMIST before taking on various industrial positions in the UK and Canada, focusing on the practical application of control theory. In 1991 he started a consulting and system integration business in system automation, optimization and data management. The business is now part of IBI Group, and he is the Vice-President of IBI-MAAK Inc. He is responsible for technology development, business and strategic planning, and management. Current effort is focused on novel approaches to automation and system optimization with a focus on energy efficiency in real-time system control.

E-mail: alnoor.allidina@ibigroup.com
IBI-MAAK Inc.
9133 Leslie Street, Suite 201
Richmond Hill, Ontario, Canada L4B4N1

Information for Authors

Journal of Telecommunications and Information Technology **(JTIT)** is published quarterly. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review**. Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

Suitable research-related papers should hold the potential to advance the technological base of telecommunications and information technology. Tutorial and review papers are published only by invitation.

Manuscript. TEX and LATEX are preferable, standard Microsoft Word format (.doc) is acceptable. The author's JTIT LATEX style file is available:
http://www.nit.eu/for-authors

Papers published should contain up to 10 printed pages in LATEX author's style (Word processor one printed page corresponds approximately to 6000 characters).

The manuscript should include an abstract about 150–200 words long and the relevant keywords. The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.

Keywords should not repeat the title of the manuscript. About four keywords or phrases in alphabetical order should be used, separated by commas.

The original files accompanied with pdf file should be submitted by e-mail: redakcja@itl.waw.pl

Figures, tables and photographs. Original figures should be submitted. Drawings in Corel Draw and PostScript formats are preferred. Figure captions should be placed below the figures and can not be included as a part of the figure. Each figure should be submitted as a separated graphic file, in .cdr, .eps, .ps, .png or .tif format. Tables and figures should be numbered consecutively with Arabic numerals.

Each photograph with minimum 300 dpi resolution should be delivered in electronic formats (TIFF, JPG or PNG) as a separated file.

References. All references should be marked in the text by Arabic numerals in square brackets and listed at the end of the paper in order of their appearance in the text, including exclusively publications cited inside. Samples of correct formats for various types of references are presented below:

[1] Y. Namihira, "Relationship between nonlinear effective area and mode field diameter for dispersion shifted fibres", *Electron. Lett*., vol. 30, no. 3, pp. 262–264, 1994.

[2] C. Kittel, *Introduction to Solid State Physics*. New York: Wiley, 1986.

[3] S. Demri and E. Orłowska, "Informational representability: Abstract models versus concrete models", in *Fuzzy Sets, Logics and Knowledge-Based Reasoning*, D. Dubois and H. Prade, Eds. Dordrecht: Kluwer, 1999, pp. 301–314.

Biographies and photographs of authors. A brief professional author's biography of up to 200 words and a photo of each author should be included with the manuscript.

Galley proofs. Authors should return proofs as a list of corrections as soon as possible. In other cases, the article will be proof-read against manuscript by the editor and printed without the author's corrections. Remarks to the errata should be provided within one week after receiving the offprint.

A copy of the JTIT is provided to each author of paper published.

Journal of Telecommunications and Information Technology has entered into an electronic licencing relationship with EBSCO Publishing, the world's most prolific aggregator of full text journals, magazines and other sources. The text of *Journal of Telecommunications and Information Technology* can be found on EBSCO Publishing's databases. For more information on EBSCO Publishing, please visit www.epnet.com.

(*Contents Continued from Front Cover*)

INSTYTUT ŁĄCZNOŚCI
PAŃSTWOWY INSTYTUT BADAWCZY

Editorial Office

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

tel: +48 22 512 81 83
fax: +48 22 512 84 00
e-mail: redakcja@itl.waw.pl
http://www.nit.eu