Annales UMCS Informatica AI XIV, 2 (2014) 7–30
DOI: 10.2478/umcsinfo-2014-0013

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Adaptable Context Management Framework for Secure Network Services

Zbigniew Kotulski[1,2*], Mariusz Sepczuk[1†], Albert Sitek[3‡], Marcin Alan Tunia[4§]

[1] *Faculty of Electronics and Information Technology,*
Warsaw University of Technology, Poland.

Abstract – Last decades the contextual approach became an important methodology of analysing information processes in the dynamic environment. In this paper we propose a context management framework suitable for secure network services. The framework allows tracking the contextual information from its origin, through all stages of its processing up to application in security services protecting the secure network application. Besides the framework's description, an example of its application in constructing secure voice call network service is given.

1 Introduction

Present security systems are considered as context-dependent [**1**]. Providing contextual information to a security system is a process which is continuous in time (since the security system works permanently) and which undergoes external disturbances. Therefore proper usage of contexual information needs management analogous to common information management systems but having its specific properties resulting from application of the contextual information.

Context can be considered from two points of view:

- As a subject of investigation in a distributed environment,
- As an object providing extended information to the security system.

To use the contextual information in the security system operating in a dynamic environment effectively, one must consistently unify the above two approaches into one context management system drawing the framework of the contextual information flow. Such information

*zkotulsk@tele.pw.edu.pl
†msepczuk@tele.pw.edu.pl
‡asitek@tele.pw.edu.pl
§m.tunia@tele.pw.edu.pl

goes from sensors of a multi-agent system analyzing events and cumulating required information, through identification, formal validation and filtration tools processing the information, to the security system which certifies definitely the context factors taken into account.

An additional aspect which must be taken into account constructing an information service with a context-aware security system, is proper service operability. Usually the information service should fulfil specific functional requirements guaranteeing simultaneously a sufficient security level. Such conditions can be described by Service Level Agreement [**2**], which is a part of a service contract where a service (including its security, [**3, 4**]) is formally defined.

In this paper we propose a context management framework which fulfils the above requirements and is compatible with several known applications of contextual information like risk analysis. Moreover, we allow the framework acting in two modes of operation: the training mode (TM), where context factors are being identified and evaluated for the first use in a security system and in the working mode (WM), where particular values of context factors are calculated and validated for security parameters calculation for the current use.

The rest of the paper is organized as follows. In the next section we present related work and motivation for our research. Section 3 gives general concept of the context management framework and defines its modes of operation. In Section 4 building elements of the framework and their functioning are presented in detail. Practical application of the framework like establishing context-aware secure voice call is given in Section 5 while Section 6 concludes the paper.

2 Related Work

Last decades researchers working on information security realized that traditional understanding of security, i.e., such where the strongest possible protection mechanisms are applied, does not respond to topical requirements of dynamic environments where information services operate. Thus, several approaches have been proposed where designers of the protection systems tried to unify (in different proportions) several aspects of the information system functioning: required security level, hardware and software efficiency or limitations, system performance, user friendliness, cost of implementation orfunctioning, etc. In this Section we will give a short overview of the main approaches to information security which is the background of the context-aware security approach being the subject of our research.

Possibility of implementing specific protection mechanisms in the information system is strongly constrained by financial costs of such investments. Their cost or effectiveness is the subject of economy of security [**5, 6**] which tries to find a balance between possible losses in case of an attack and price of countermeasures applied. A more complete approach is security engineering. Security engineering is a specialized field of engineering that focuses on the security aspects in the design of systems that need to be able to deal robustly with possible sources of disruption, ranging from natural disasters to malicious acts. Its primary motivation is to support the delivery of security solutions that satisfy some pre-defined functional and user requirements, but with additionally preventing misuse and malicious behaviour [**7**].

A supplementary approach is the adaptive (or scalable) security which concentrates on technological aspects of implementing security in information systems and recommends restricting the protection level due to hardware/software performance limitations [**8**]. It is a kind of Service Level Agreement (SLA) [**2**], where the assumed system performance must be achieved without a minimal security level loss. More formally adaptive security is formulated as Quality of Protection (QoP) approach. The QoP proposes a unified paradigm that places all security service classes on a continuous spectrum of protection grades (where different sub-ranges of grades map to a single protection class) [**9**]. This paradigm is based on assigning a guaranteed Quality of Protection (or QoP) to each communication connection or information service. Usually QoP defines a range of allowed system configurations with QoP levels assigned from the least to the most secure. This range of security levels must be measured, e.g., by the equivalent key length and the type of cryptographic algorithm [**8**]. The system with the implemented QoP allows reconfiguration to obtain optimal performance with sufficient (user-defined) security requirements and resources availability. In a specific application such a QoP approach in telephony and computer networks can be considered as Quality of Service (QoS) that allows the traffic transport with special requirements concerning a warranty of their quality. This is especially important in real-time services where strong security services must share resources with transmission services requiring high quality, see e.g. [**10**]. The same problem, but presented from a consumers point of view, is considered as Quality of Experience (QoE) [**11**]. QoE is a subjective measure of customer's experiences with a service (web browsing, phone call, TV broadcast, videoconference). Again, QoE is strongly connected with security methods applied during service protection [**12**]. Since cryptography engages computer resources, it can negatively affect the service quality and disturb users perception. All the above approaches focus on the effect we want to obtain: good quality of information service (that is with low financial cost, sufficient performance, expected quality of users perception, etc.) but not based on external constraints leading to optimal solution. The context-aware security approach takes these environmental factors into account.

Context-aware security involves usage of supplemental information (called the context) to improve security decisions at the time they are made, which results in better decision-making capable of supporting information services in the dynamic environment [**1**]. The context information concerns entities that participate in the service, their activities or, generally, situations including several entities and activities; it can be internal, connected with the service provider of user, or external, connected with the environment. Basic context information for security of information services come from the IT stack, including IP, device, URL and application reputation. The context information can be incorporated into various security services like authentication (access control), encryption (information confidentiality), and availability (expressed as information system performance) [**13, 14**].

In literature one can find many publications where context-aware security services are proposed for applications in different information systems. A very recent state-of-the-art analysis can be found in paper [**15**] where the authors described and classified more than twenty particular approaches according to context category used and a security service where the context information was applied. Also a very interesting survey on the context-aware systems has

been presented in paper [**16**]. The authors show many different approaches to context-aware systems definitions available in literature. They also give a complex review of the history of context-based research and methods of presenting and storing context information. Besides finding specific solutions, some authors made an attempt to construct formal models of using context for specific classes of security services. For example, in [**17**] the authors proposed a framework for context-aware authentication possible to use in different information services. However, besides specific applications of context in security solutions, one can find papers concerning more fundamental problems of context-aware security. For instance, in paper [**15**] a conceptual model for security context is described including social aspects. There is described sample scenario, which includes diametrical context changes. It was assumed that users may apply their terminals for both personal and professional purposes. The model includes three levels of abstraction: entity, activity, and situation and defines relations between them. The approach presents how to model the context situation and precisely transmit contextual information for further processing (storage, transmission, updating, retrieval, etc.). However, this approach does not define clear data flow and context reasoning stages in the security context aware system as well as it does not consider the effect of context application on the information system quality. In contrast, many publications focus on further stages of the context management. For example, in paper [**18**] the authors present life-cycle of context in the context-aware systems including three steps: discovery of context information, acquisition of context information and reasoning about context information. In this model a context-aware system finds context information providers (CIP) which deliver context data. Next, the system collects the data from CIPs and put it in a context data store for further reasoning. The last step is reasoning about data which have been collected. A practical application of this life-cycle scheme is The Smart Floor [**19**] mechanism for natural user identification and tracking.

Recent studies make an attempt to manage contextual information in a way dedicated to information services in general or to specific ones. Here we give a brief overview of the results that are representative of the variety of solutions available in literature. Thus, in [**20**] the authors present The Contextual Service Adaptation Framework, which provides a platform for supporting the adaptation of services to different kinds of contextual information. The main idea focuses on three aspects: Context Management, Context Recognition and Service Adaptation based on Context. The Context Management supports the process of contextual data capturing and storing in Context Repositories. The Context Recognition classifies the context data from the repositories and the Service Adaptation refers to adaptation of services based on available contextual data at the particular moment. In paper [**21**] a Classification Framework for Storage and Retrieval of Context is presented. The document shows three phases which should be present in each context-aware system: Environment monitoring, Interpretation though context model and System adaptation based on Context information. The paper focuses on a classification for the modeling and retrieval of context and gives examples of the framework usage. In [**22**] we can find a very similar idea to the previous frameworks; it includes such stages as: Context Retrieval, Context Storing and Context Manipulation. Another Context Management Framework is shown in [**23**]. The CoBrA system is a context broker, which uses ontologies and maintains model of context with cooperation from other service

agents. An interesting approach of context management is described in paper [**24**]. The paper discusses architectural issues that are important in a context management system for Personal Networks. It is noteworthy that the authors list requirements for a context management system, which include conditions associated with security, like: privacy of the user, authentication, data integrity and confidentiality as well as data freshness and non-repudiation. Based on these requirements a user can get access to some data or not.

A flexible and extensible technology-neutral information model, which can represent multiple-domains context, and a hierarchical context management architecture, which can understand and manage complex interactions among multiple contextual entities, has been proposed in paper [**25**]. In such a system, called U-CoUDE (which stands for User-centric Context manager for Ubiquitous and Distributed Environments.), it is assumed that context management can be categorized into three levels of management coverage: Domain level (collects, infers and provides context data from a specific domain like smartphone and smarthome), User level (collects and aggregates context data from multiple user-related domains) and Social level (considers the context obtained and generated by means of interactions between a user and socially related other users). This approach can be related to other social-related approaches generally connected with SLA, particularly with QoE. The corresponding term Quality of Context (QoC) has been clearly presented in paper [**26**]. The authors suggest that Internet of Things (IoT) will contribute to context-awareness of numerous sensitive applications because context data will be also gathered from things connected to the network. In the paper the authors also point out that context data can carry sensitive data. Thus, access to those data should be properly secured. In paper [**27**], the authors present the context management framework that exploits the agent technology in mobile communications and services of 4G networks. They focus mainly on the seamless secure handover which uses context information regarding location privacy, security, network environment, and QoS priority. They designed QoS Broker that can perform the autonomous decision making for context-aware handover without the direct intervention of users. They also designed the context model for the seamless vertical handovers so that the Brokers could ensure the right context is available in the right place at the right time.

A lot of papers concentrate on utilizing context for security applications. For example, paper [**28**] presents a scheme of deriving important information for security from context data and a framework for security-relevant context. This approach, however, does not include risk management and reputation. There is no description how to validate results of the security adaptation. A decision-support framework concerning context data has been presented in paper [**29**]. The authors presented two stages of context-based decision-making: the preliminary stage and the decision-making stage. The first one is responsible for modeling components, knowledge gathering and linkage with the environment. The second one includes knowledge integration, problem modeling and solving. The framework works as a supporting mechanism used on demand. In the context aware security there is a need to introduce a model which would constantly respond to the environment changes. Paper [**30**] presents the context-aware framework for information delivery. It supports business processes identifying information necessary for the parties of the processes. It is a useful solution in complex environments

where huge amount of data is connected with one process and only its small part is required at a time. However, this is not a strictly security-oriented solution and it does not include risk consideration and securing the service on the basis of context information. It only delivers data to adequate participants of the process.

Security context, which is extensively studied for specific applications protecting information systems and which is often considered in general models describing security services, can be also effectively used in abstract formal languages applicable for security protocols modeling. The QoP-ML modeling language for cryptographic protocols [**31**], which is originally dedicated to reflect Quality of Protection constraints in security protocols operation, could be a formal languages level modeling tool context-aware security protocols.

Brief analysis of the above models shows that none of them is a complete context management system which could provide contextual information identification, acquisition and processing to optimize functioning of a secure information system. They focus on possibly complete description of contextual information, reliable collecting such information and its precise processing but without taking into account the superior purpose which is optimal functioning of the secure information system. The aim of our paper is to propose the new context management framework for secure information systems which provides contextual information to context-aware security services implemented in a secure information system in such a way that it optimizes its functioning. The system we propose is in its main idea similar to risk analysis methods where, due to uncertainty and dynamics of the environment, the context becomes their immanent part [**32, 33**]. Moreover, the process of risk management always starts with context establishing, so the risk management (or risk analysis) is a kind of a context-aware process. This occurs especially in the context-aware information security [**34**], where one assumes high dynamics of processes, which results in a quick circulation of the risk management circle and, what follows, in a very intensive context establishing process. In our model we concentrate on the optimal context establishment where the optimality measure is imposed by expected optimal functioning of the secure information system in which the context-aware security services are implemented.

3 General Concept

3.1 Framework definition

To manage efficiently the contextual information for security systems purposes we propose a layered structure of the management model. As usual in such a stack of layers, a preceding (lower) layer serves as a consecutive (higher) layer. This makes definitions of processes in each layer more transparent and their functioning more productive. Our model consists of four layers and two intelligent communication channels as presented in Figure 1. The first layer is the Context Data Acquisition one which corresponds to the information system environment including all entities, their relations and actions. The second layer is the Context Identification one which defines context important for the security services applied for protection of the information system. The next (third) layer is the Context Adaptation one which includes

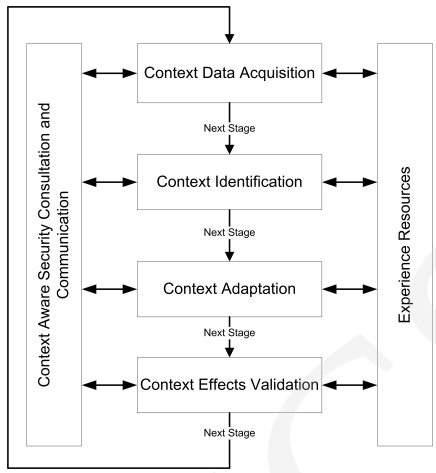

FIG. 1. Building layers of the context management framework

security services and their context-awareness. The last, fourth layer is the Context Effects Validation one which covers the state of the information system where the context-aware security services are applied and where the service providers and the end-users operate. Besides the four hierarchical layers that constitute the framework, two additional parallel layers support communication and information exchange. The Consultation and Communication layer plays a role of intelligent communication channel and it closes feedback between neighbouring hierarchical layers. The Experience Resources layer manages information exchange between the active part of the framework and the knowledge repositories, both internal and external. More detailed description of each layer is given in the Elements of the framework section.

3.2 Modes of Operation

The framework presented in Figure 1 is expected to work in two modes of operation. The modes correspond to a stage of the context-aware security system life cycle. Thus, the framework must be trained and tuned for the first use. This stage of its exploitation is the Training Mode. Next, the framework starts its regular life providing context information functions to the security system. This is the Working Mode. The outline of functioning of the context management framework in the above two modes is as follows.

The framework information flow in the Training Mode is presented in Figure 2. In this mode each layer fixes its basic components and verifies formally their choice. Next, the settings of each layer are validated according to some local layers quality criteria. After tuning the whole layered system (according to criteria implemented in the Context Effects Validation layer), the Training Mode switches to the Working Mode.

In the Working Mode presented in Figure 3 each layer has its settings already fixed. The framework performs operations at all layers estimating values of context factors and validating their quality or validating effect of these values on functioning of the security system and, generally, on security of the information system. In the case of decreasing the QoP parameters

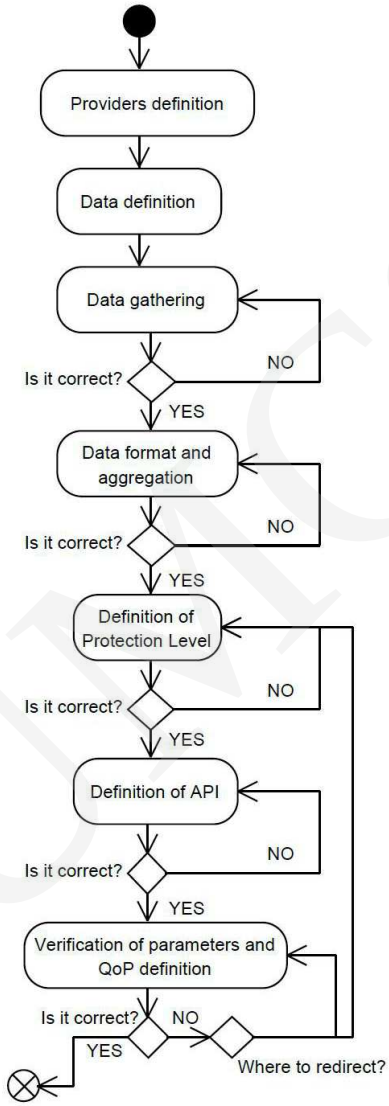

FIG. 2. Information flowchart for the Training Mode

below a certain level and a lack of solution for actual framework settings, the Working Mode temporarily switches to the Training Mode.

More detailed description of functioning of the framework and the framework layers in the two modes of operation is presented in Section 4.

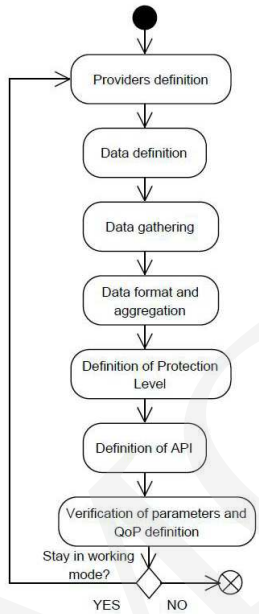

FIG. 3. Information flowchart for the Working Mode

4 Elements of the Framework

4.1 Context Data Acquisition

The first layer in our framework is Context Data Acquisition. Before we can use context in a specific way, we should discover sources of context information, collect data from them and take advantage of gathering information. In this stage our framework searches raw data and stores it for future identification. The important fact is that at the beginning all data are collected without any preferences, according to sensing possibilities. After data acquisition, raw context information is sent to the next layer, which is the Context Data Identification layer, for further processing.

As it was assumed in Subsection 3.2, the framework can work in two modes. The specific actions performed in this layer provide the framework with all-embracing, high-quality observations useful for identification of context factors. Thus, in the Training Mode of the Context Data Acquisition layer:

- The framework searches and discovers the Context Information Providers,
- The Context Data Acquisition layer periodically communicates with the Context Data Identification layer to obtain searching and accuracy recommendations,
- Reports about problems and unforeseen circumstances to the Context Aware Security Consultation and Communication layer and the Experience Resources layer.

Next, in the Working Mode the Context Data Acquisition layer:

- Gathers fresh context data to the context-aware system,
- Stores collected data in proper repository,
- Can stop data gathering from the selected CIP, depending on the feedback information from the Context Effects Validation layer.

The Context Data Acquisition layer has three interfaces to communicate with other layers, (see Figure 1) as follows:

- To communicate with other layers through the Context Aware Security Consultation and Communication layer in the case of errors or other unexpected events when additional framework recommendations are required,
- To exchange historical data with the Experience Resources layer or to obtain external knowledge,
- To pass gathered data to the consecutive Context Identification layer.

It also inputs feedback information from the Context Effects Validation layer about changing recommendations for contextual data acquisition and further processing.

Proper validation of data is performed at the end of this stage, before sending data to the next stage (the Context Identification layer) and (optionally) to the Experience Resources layer. The preliminary validation of the raw data quality concerns:

- Checking formal correctness of the gathered data (e.g., strings are strings, numbers are numbers, etc.),
- Estimating measurements accuracy it applies to the data,
- Checking syntax correctness of the gathered data.

4.2 Context Identification

The second layer in our framework is the Context Identification layer. The purpose of this layer is to transform raw context data received from the Context Data Acquisition layer and gained from the Experience Resources into high quality, well-formatted context data which can be passed directly to the Context Adaptation layer. This stage is responsible for:

- Raw context data validation: checking integrity, accuracy, usefulness, completeness, non-redundancy, etc.,
- Context data acquisition control with respect to the above properties,
- Context data categorization,
- Context information extracting and formatting.

In order to extract useful context information from raw data, we must perform specific processing. Data received from the preceding layer may be:

- Not valid: some unexpected, potentially invalid situations can be revealed based on data gained from history stored in the Experience Resources layer, for example, if air temperature is monitored and its value fluctuates in a wide range in a short period of time, it may indicate sensors failure and should be reported to the preceding layer,
- Redundant: there is a possibility that data gathered in the preceding layer consists of correlated factors, for example, in the case of localization data, current country and

GPS coordinates are dependent and only one of them should be analyzed. In such a situation, a proper alert should be sent to the preceding layer,
- Useless: if data received from the Context Data Acquisition layer do not have impact on the result of this processing or certain request has been received from the preceding stage (to stop sending a type of context information), collecting of such data should be stopped,
- Insufficient: it is possible, that raw context data is not complete and proper context information cannot be generated, for example, if information about users device is required and there is no data about current mobile network signal strength. In that situation, proper request should be sent to the preceding layer to start gathering missing data,
- Missing: there may be such a situation that completely new context information should be generated. For example, if this block received a request from preceding layer, that there is a need to provide information about neighborhood, a proper request should be propagated to the preceding layer.

A normal workflow at this stage (in the Working Mode) is as follows:

- Receive context data from the preceding layer,
- Validate received data,
- Gather needed information from the Experience Resources layer,
- Prepare and format proper context information,
- Validate generated context information,
- Store it in the Experience Resources layer, if required,
- Pass it to the consecutive layer.

In the Training Mode, which usually initiates the framework, the Context Identification layer performs analogous actions but with deeper analysis and with an additional context factors selection stage. During the framework regular work, the Working Mode switches into the Training Mode in such situations:

- The validation process failed, and a proper alert has been sent to the preceding layer,
- A request from the consecutive layer has been received and certain type of information should not be delivered anymore,
- A request from the consecutive layer has been received and a new type of context factor or contextual information should be generated.

As it is shown in the block scheme presented in Figure 1, this layer communicates with other layers through four interfaces:

- To receive raw data from the Context Acquisition layer,
- To communicate with preceding and consecutive layers through the Context Aware Security Consultation and Communication layer, in the case of extraordinary situations,
- To exchange historical data and obtain reference information and extra recommendations from the Experience Resources layer,
- To pass the product of this stage to the consecutive Context Adaptation layer.

As we mentioned in the manifest of this stage, context information as a result of this layer should be high quality and well formatted, which means that:

- Should contain only necessary data to deliver the essence of contextual information,
- Should be presented in a format which is accepted by the next layer, for example, should be well categorized: as external (or internal, combined) information about the user (or: device, environment, neighbourhood, etc).

Proper data validation should be performed at the end of this stage before sending information to the Experience Resources and the Context Adaptation layers.

4.3 Context Adaptation

On the basis of security risk identified from thr results of the Context Identification layer and recommendations obtained from the Experience Resources, there should be defined countermeasures to respond to the context-dependent threats. The Context Adaptation layer defines actions and controls their execution. The set of available solutions depends on the network service which we want to protect. Usually, the context-aware protection system and its usage conditions are defined during the Training Mode of the framework when we "teach" the framework how to operate with a certain network service. After defining the actions, there is a phase of adapting the network service according to the previously defined actions.

Firstly, the framework determines the set of countermeasures that should be implemented in order to maintain a desired security level, which may be defined by some assumed SLA or internal service settings. The resultant security level depends on the security services implemented in the network service. On the basis of the received parameters, the framework filters the context factors which influence the protection level provided by each security service. After that, new parameters for the security services are established to maintain a desired security level.

First of all, the context-awareness can be incorporated into the basic security services which are: data confidentiality, integrity and availability, see [**35, 36**]. However, in modern information systems more security services should be implemented, most of them with context factors affecting their functioning. They concern not only data security but also security of users, services, devices, networks, etc. Such security services are, for example, privacy and confidentiality, data integrity, authentication and identification, authorization, access control, anonymity and non-repudiation, see [**37**]. Thus, once the countermeasures to perform are defined, the framework activates their execution. As it was written before, there are two modes of the framework operation. In each mode the framework either uses the countermeasures with previously defined parameters to operate, or learns new parameters which will be used in future operation. Below we describe two modes of work for this layer.

In the Training Mode the framework learns:

- What types of actions (countermeasures, security services) can be performed to protect the network service,
- How to associate actions with contextual parameters that come from the preceding layer,

- How to execute previously defined actions.

During the training proper security services are selected (those that influence security level for the context service). Each security service has its model that depends on the context service requirements. During this mode an applicable model for each security service is created. There are also defined types of errors and situations in which there is no sufficient information to determine a set of actions or method to execute them. During this step the framework also learns how to report errors and problems to the Context-Aware Security Consultation and Communication layer. In the Training Mode there is no experience collection but rather using the existing experience and recommendations.

In the Working Mode the framework:

- Determines the set of actions in response to input parameters received from the Context Identification layer and takes into account the current service status,
- Determines current service status on the basis of data coming from the two layers: the Context Aware Security Consultation and Communication and the Experience Collection,
- Adapts the security services according to the previously determined actions,
- Uses adequate mechanisms to switch between the Working and Training modes, according to actual security requirements.

During the Working Mode there is also a reporting process performed in favour of two layers:

- Errors and problems reported for the Context-Aware Security Consultation and Communication layer,
- Experience data exchange with the Experience Resources layer.

In the Working Mode the framework validates how each security service is working with new parameters to confirm if it is well-defined. In the case of problems the model of security service should be updated to work properly with new parameters. Once the framework finishes the Context Adaptation phase, the service should be working with appropriate security level regarding current context situation.

After the Context Adaptation layer finishes its work, a set of data will be generated and passed to the next layer: The Context Effects Validation. Before it leaves the Context Adaptation layer, there is a need to locally validate data with the following conditions:

- Contradictions in the data set,
- Format compliance with the next layer interface,
- Format compliance with the Context-Aware Security Consultation and Communication layer interface, in the case of errors or a lack of possibility to redirect control to the next layer.

4.4 Context Effects Validation

Final validation of the effect of contextual information on the security systems is through the observation of the network information service where the context-aware protection mechanisms are being incorporated. The end-user (the information service provider) decides whether

the security system satisfies the Service Level Agreement (in particular, the Quality of Protection) requirements:

- It guarantees a sufficient security level,
- The cost of its functioning is in the desired frames[1],
- It satisfies other conditions assumed, e.g., sufficient performance of the system (QoS), its user-friendly functioning (QoE), etc.

Generally, this stage is responsible for:

- Monitoring the network information service where the context-aware security services are implemented, both, from the end-users and the network service providers points of view,
- Registering and validating security incidents connected with functioning of the information system,
- Providing data for estimation of reputation of all service users, reputation of internal subsystems and external contractors (functional services and communication services providers, external data resources, etc.),
- Verifying data required for optimal functioning of the network information service,
- Following the quality of protection conditions related to specific contract requirements of the service (QoS, QoE, etc.),
- Deciding whether the whole context management system works in the Training Mode or in the Working Mode.

The Context Effects Validation layer performs the functions listed above both in the training mode and in the working mode but in each case with different intensity. Thus, in the Training Mode the purpose of validations is defining the settings of the system in such a way that for all expected threats the system can realize security aims in some pre-defined context frames. Moreover, this also means that the QoP conditions must be satisfied. The context management system remains in the Training Mode until a training stop condition is not satisfied. If it is satisfied, the system switches into the Working Mode.

In the Working Mode the Context Effects Validation layer verifies if the present values of the parameters of the context-aware security system satisfy some pre-defined optimum criteria of system functioning. If not, the layer generates requests of parameter modification. Such a request is transmitted through the Context-Aware Security Consultation and Communication channel to an appropriate layer to modify the parameters. If in the frames of present settings it is impossible to achieve the aims of the system, the Context Effects Validation layer switches the context management system into the Training Mode to modify system settings and/or the QoP (SLA) conditions.

As it is presented in Figure 1, the Context Effects Validation layer has three interfaces. It is also the output of the whole framework with the feedback loop to the origin of the context management process. First, the layer receives the context-aware security services from the Context Adaptation layer to integrate them with the whole information system. It contacts the Context-Aware Security Consultation and Communication layer to send requests of changes

[1]As the cost of the security system functioning we assume the direct cost of implementation of protection mechanisms and some indirect costs, e.g., extra energy consmuption.

to the preceding layers in the case of such recommendations of the validation process. It also contacts the Experience Resources layer receiving external information needed in validation and historical reputation data. This layer also stores actual reputation recommendations in the Experience Resources databases.

4.5 Context-Aware Security Consultation and Communication. Experience Resources

The hierarchical layered structure of context management is associated with two parallel layers playing a role of the cross-layers intelligent information channels. They are indispensable in our system. The Context-Aware Security Consultation and Communication layer allows repeating operations of certain layer(s) in the case of a need of changing its settings or changing the values of the parameters for fixed settings. It not only transmits requests but also decides how deep the feedback loop is, so that, actions of which layers must be repeated. The Experience Resources layer delivers any information from outside the framework that can be useful in the context-aware security system. It can be standardization information, experts knowledge and actual information about threats. The Experience Resources also store and retrieve internal information of the context management framework, including processes history, reputation scores of the entities and actual trust recommendations. This means that the Experience Resources layer together with the Context Effects Validation layer build reputation of all entities participating directly in the context management system and stores the reputation recommendations for future applications.

5 Practical Use Case

The voice call is one of the most frequently used network services. By definition, it is a connection over a telephone network between the calling party and the called party. Nowadays, this service can be served over several types of networks, like PSTN, GSM, UMTS, IP etc., and using different protocols, see [**38**]. However, besides for the technology, in the voice calls the human factor must be taken into account. People communicate with each other several times a day, in different situations and from different localizations. Each time, they expect that the connection will be established quickly and convenient for the users, the voice will be (at least) understandable, and the whole connection will be protected properly. In this section, as an example, we present how our framework can be utilized for context-aware security mechanisms management during a voice call, especially over IP networks.

5.1 Context Data Acquisition

To properly establish the connection let us begin with the requirements which voice call has to fulfil. Main issues, usually included in SLA, are as follows:

- QoS: low delay during transmission, a lack of signal during call, perspicuity of voice during transmission,
- QoE issues: the voice call should be nearly like a natural conversation between people (if QoS issues are achieved, the QoE is achieved too),

- QoP issue: protection level should be adapted to the current moment (voice call should be confidential without sniffing possibility).

Let us assume that two workmates, situated in one office, initialized voice call over the IP network using their cell phones. In such a situation, following context information could be processed through our framework during the service initialization procedure:

- Localization,
- Type of connected network (private, public),
- Type of used device,
- Neighborhood,
- Position in the company (do they have access to confident information?).

At the beginning let us assume that both employees are situated in the same building, their devices are connected to a private secure company network, they are closed in their rooms, and they do not have access to vulnerable information. In that situation our framework decided not to protect the whole communication, because there is no such a need.

During the whole call, our framework monitors the current context in order to react properly for a probable change. Let us assume that suddenly one employee left his office and went to the shop. The context has changed diametrically: the network connection switched to UMTS (so part of communication is routed through an unsecured, public network), and other people appeared in the employees neighborhood. Because of that, our framework decided to renegotiate the whole session and to establish a secure connection between employees. The whole environment is still being monitored till the end of the call. Finally, the connection is ended and the framework finishes its work. For better clarity of the use case description, we will describe only the frameworks reaction to geographical coordinates change in detail.

5.2 The Framework Use

First, during the Training Mode we have to prepare the framework to operate correctly in a production environment, so we have to define initial parameters for each layer. On the basis of training data the framework should be prepared, passing through its all parallel layers and using its subsidiary layers, with the following parameters and actions:

- Context Data Acquisition:
  - Define providers for employee's current coordinates (for example GPS or GLONASS) and agents that will provide users' coordinates on framework's demand.
- Context Identification:
  - Define essential information for the user context definition: current position (inside or outside the office),
  - Define aggregation rules and proper data format for the user context: in this case coordinates are translated into information the user is inside or outside the office using office geographical borders definition. Localization information is formatted as follows: value "inside" - the user is in the office and value "outside" - the user is outside the office.

- Context Adaptation:
  - Define needed security services to assure voice call required protection: in this case we use privacy and confidentiality service,
  - Define which parameters of security services have to be changeable: in this case we have to change encryption strength,
  - Define methods for getting current status and altering voice call service working parameters: in this case we use the service API which delivers the following methods: `getCurrentParameters()` and `setParameters()`.
- Context Effects Validation:
  - Define all possible threads to the voice call and assign context frames to them: in this case we assume that the call may be eavesdropped by unauthorized entity while the user is outside the office,
  - Define principles for evaluating each implemented security service level of protection: we assume that the level of protection is correctly set if encryption strength is always the same as defined in SLA for current user's position.
- Experience Resources:
  - Define structure of stored experience information: in this case we will store information about adaptation time according to changed parameters and record responses for typical parameters in order to optimize adaptation time.
- Context Aware Security Consultation and Communication:
  - Define principles for handling problems reported by each layer: in the case there is a problem connected with a lack of information we put back processing to the previous layer, if there is another problem we put back processing to the first layer.

In the Working Mode the framework attempts to adapt the voice call service security parameters as accurately as possible considering a current context situation. The framework should perform the following actions during each layer processing:

- Context Data Acquisition:
  - Gather data about the users localization.
- Context Identification:
  - Aggregate received context data in order to produce the users localization context.
- Context Adaptation:
  - On the basis of data from the previous block define current risk connected with the user's context: in this case we can map outside office situation to high risk and inside office situation to low risk,
  - Decide if there should be change in currently working security services parameters by questioning service API (with `getCurrentParameters()` method) for current working parameters and defining which risk level is currently handled. If current production supported risk level is the same as determined according to a new localization there is no need to perform change, otherwise there is a need to perform an adaptation procedure,

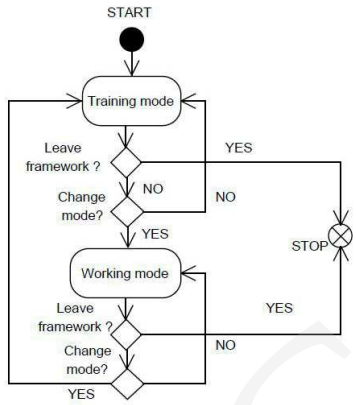

FIG. 4. Switching between Working and Training Modes

  - Perform an adaptation procedure if needed: invoke `setParameters()` service API method.
- Context Effects Validation:
  - Evaluate current privacy and confidentiality service level of protection,
  - Check if current level of protection is consistent with expectations,
  - Generate appropriate validation results.
- Experience Resources:
  - Gather experience data provided by each layer and external sources,
  - Convert received data to the internal, previously defined format.
- Context Aware Security Consultation and Communication:
  - In the case of an error reported from one of the layers decide to which layer the processing should be redirected (how deep the feedback loop should be).

Finally, the framework is obliged to switch between the modes in specific situations. Transitions between working and training modes (Figure 4) are possible when the environment, which we use to deliver the voice service, has changed.

In the use case presented in this section, when two employees are connected with voice call in the office, their transmission initially should not be encrypted, because this place is considered to be safe, without a possibility of sniffing. The decision not to encrypt data transfer was based on the information about the localization. Neighbourhood was omitted as redundant and not important information. Parameters of the voice call in this environment were monitored by the framework in the Working Mode. When one employee went to the shop and still continued conversation, the framework detected localization change. The shop is located in a shopping center, so we assume that there may be malicious people who want to sniff all transmitted data. In this case, knowledge about neighbourhood is required. The framework changed its mode to Training mode to search for new context information, like neighbourhood. Moreover, based on localization of a shop, protection level of a voice call must be increased to provide confidentiality of conversation. At the same time security conditions must be changed.

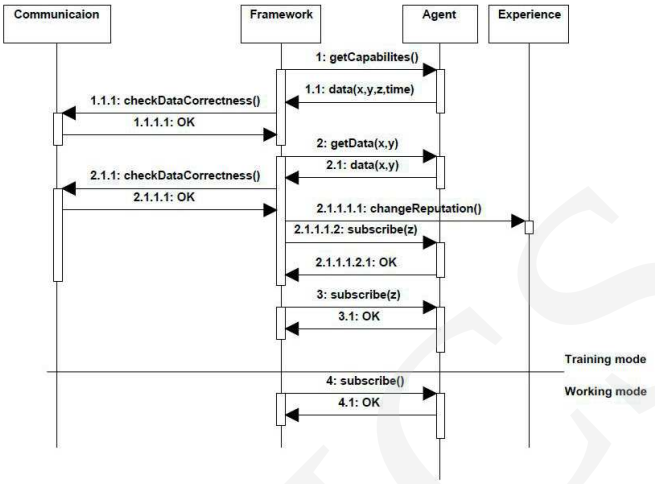

FIG. 5. Framework communication flow diagram for localization of context information gathering and processing

There is also possibility of leaving or suspending a framework process with the "leave framework" decision as depicted in Figure 4. This can place when we want to reset provided service (in situation when we change an exploitation area of service, with saving a general core of service, we have totally different environment where the framework must start learning from the beginning) which means that data from the experience resources and other historical information would be deleted.

5.3 The Framework Communication Flow Diagrams

During its work, the framework must communicate through several interfaces in order to gather context information and send its output to the service API. To simplify our diagrams, we present the messages flow for the localization context information gathering and processing, see Figure 5. The messages flow for any other context information would look similarly.

Communication steps in the Training Mode and the Working Mode are presented in Figure 5. There are four parties involved in the information flow:

- Communication, which represents the Context Aware Security Consultation and Communication layer,
- Framework, which represents the main decision process of the framework,
- Agent, which represents a context agent,
- Experience, which represents the Experience Resources layer.

The following functions are performed during the Training Mode, as presented in Figure 5:

- `getCapabilities()` is used for obtaining types of context data, which an agent is able to deliver (in our case the agent is able to deliver current time and three coordinates: x,y,z),

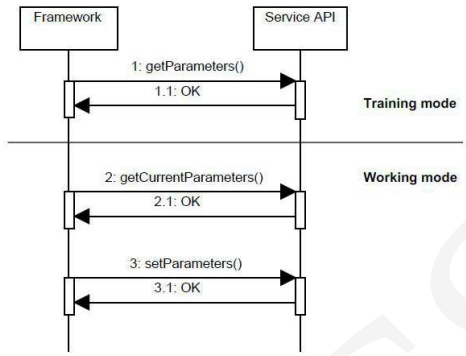

FIG. 6. Framework - Service API communication flow diagram

- `getData()` orders an agent to collect certain types of context information (in our case the agent has to deliver x and y coordinates),
- `checkDataCorrectness()` verifies if the obtained data is correct,
- `changeReputation()` updates reputation data,
- `subscribe()` orders an agent to deliver certain types of context information.

During the Working Mode only one function, `subscribe()`, is used.

For communication with the Service API we propose to use only three commands, see Figure 6:

- `getParameters()` to get all commands service API is able to perform,
- `getCurrentParameters()` to get current security parameters of the service,
- `setParameters()` to set security parameters according to the framework output.

5.4 Frameworks Results

Practical example, shown in this section, presents a lot of benefits which are the result of using the proposed framework. Some of them are as follows:

- Detection, fast reaction and adjustment to context change due to existence of two modes: Training Mode, Working Mode and transitions possibility between them,
- Optimal adaptation of security mechanisms level based on actual context information,
- Possibility of working in every environment in which there are context information providers,
- Achieving required QoP level contained in the SLA contract.

6 Conclusions and Future Work

In response to dynamic changes in contextual ubiquitous computing services and fast changing security requirements we presented context management framework, which can be utilized for various cases and implemented with various network services. It covers full context analysis, from raw context data gathering to context-dependent security adaptation and effects validation. Moreover, it can be used during the whole lifecycle of a service delivered for the end user and adapted according to changing requirements. The framework assures protection required at a certain time and situation in compliance with SLA and other agreements. There is also possibility of integration with business processes characteristic of network provider, service provider or end user. Each layer can be developed individually to meet each entity's expectations.

In this paper we presented our framework quite generally, so this topic can be expanded in future research easily. One may try to design and describe each block of the framework in detail, so that practical implementation would become more probable. This could also enable researchers to analyze each block separately, so that such factors like performance, accuracy etc. can be measured and optimized easily. This may be achieved, for example by creating a quantity and quality model for each block. On the other hand, it would be worth trying to adapt the framework to the requirements of the specific network provider, service provider or end user. Such research could be a great opportunity to gain sponsors and clients for real implementation.

References

[1] MacDonald, N., The Future of Information Security is Context-Aware and Adaptive, Gartner RAS Core Research Note G00200385, p. RA341601022011, 14 May 2010

[2] Keller, A., Ludwig, H., The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services, Journal of Network and Systems Management 11(1) (2003): 57–81.

[3] Henning, R.R., Security service level agreements: quantifiable security for the enterprise?, Proceedings of the workshop on New security paradigms, Caledon Hills, Ontario, Canada, September 22-24 (1999).

[4] Bernsmed, K., Jaatun, M., Meland, P., Undheim, A.: Security SLAs for Federated Cloud Services, Sixth International Conference on Availability, Reliability and Security, ARES (2011).

[5] Moore, T., Anderson, R., Economics and Internet Security: a Survey of Recent Analytical, Empirical and Behavioral Research, TR-03-11, Computer Science Group, Harvard University, Cambridge, Massachusetts (2011).

[6] Anderson, R.J., Economics and Security Resource Page, http://www.cl.cam.ac.uk/ rja14/econsec.html (2014).

[7] Anderson, R.J., Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley (2008).

[8] Ksiezopolski, B., Kotulski, Z., Adaptable security mechanism for dynamic environments, Computers & Security 26(3) (2007): 246–255.

[9] Gerstel, O., Sasaki, G., Quality of Protection (QoP): A Quantitative Unifying Paradigm to Protection Service Grades, OptiComm 2001, Optical Networking and Communication Conference (2001).

[10] Xue, G., Chen, L., Thulasiraman, K., Quality-of-Service and Quality-of-Protection Issues in Preplanned Recovery Schemes Using Redundant Trees, IEEE Journal on Selected Areas in Communications 21(8) (2003): 1332–1345.

[11] Fiedler, M., Hossfeld, T., Tran-Gia, P., A generic quantitative relationship between quality of experience and quality of service, Network, IEEE 24(2) (2010): 36–41.

[12] Ciszkowski, T., Mazurczyk, W., Kotulski, Z., Hossfeld, T., Fiedler, M., Collange, D., Towards Quality of Experience-based Reputation Models for Future Web Service Provisioning, Telecommunication Systems Journal 51(4), 283-295 (2012)

[13] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P., Towards a Better Understanding of Context and Context-Awareness, Handheld and Ubiquitous Computing, LNCS, vol. 1707, Springer, Berlin (1999): 304–307.

[14] Weiser, M., The Computer for the 21 st Century, ACM SIGMOBILE Mobile Computing and Communications Review 3 (1999): 3–11.

[15] Jovanovikj, V., Gabrijelcic, D., Klobucar, T., A conceptual model of security context, International Journal of Information Security (2014).

[16] Baldauf, M., Dustdar, S., Rosenberg, F., A survey on context-aware systems, Int. J. Ad Hoc and Ubiquitous Computing 2(4) (2007): 263–277.

[17] Hayashi, E., Das, S., Shahriyar, A., Owusu, E., Han, J., Hong, J., Oakley, I., Perrig, A., Zhang, J., CASA: A Framework for Context-Aware Scalable Authentication, SOUPS'13: Ninth Symposium on Usable Privacy and Secrecy (2013).

[18] Wrona, K., Gomez, L., Context-aware security and secure context-awareness in ubiquitous computing environments, Annales UMCS Informatica AI 4 (2006): 332–348.

[19] Orr, R.J., Abowd, G.D., The Smart Floor: A Mechanism for Natural User Identification and Tracking, Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands (2000).

[20] Contextual Service Adaptation Framework, http://soa4all.eu/contextmanagement.html (2014).

[21] Siljee, B., Bosloper, I., Nijhuis, J., A Classification Framework for Storage and Retrieval of Context, KI-04 Workshop on Modelling and Retrieval of Context, CEUR 114 (2004).

[22] Preuveneers, D., Berbers, Y., Adaptive Context Management Using a Component-Based Approach, Distributed Applications and Interoperable Systems, LNCS, vol. 3543, Springer, Berlin Heidelberg (2005): 14–26.

[23] Chen, H., An intelligent broker architecture for context-aware systems, A PhD. Dissertation Proposal in Computer Science at the University of Maryland, Baltimore County (2003).

[24] Bauer, M., Olsen, R., Jacobsson, M., Sanchez, L., Lanza, J., Imine, M., Prasad, N., Context Management Framework for MAGNET Beyond, Workshop on Capturing Context and and Context-Aware Systems and Platforms, Proceedings of IST Mobile and Wireless Summit (2006).

[25] Won-Ki Hong, J., Han, Y., Kang, J.-M., Seo, S., Context Management for User-centric Context-aware Services over Pervasive Networks, 14th Asia-Pacific IEEE Network Operations and Management Symposium (APNOMS 2012) (2012): 1–4.

[26] Chabridon, S., Desprats, T., Laborde, R., Marie, P., Marquez, S., Oglaza, M., A survey on addressing privacy together with quality of context for context management in the Internet of Things, Annales of Telecommunications-Annales des telecommunications 69(1-2) (2014): 47–62.

[27] Lee, M., Park, S., A Secure Context Management for QoS-Aware Vertical Handovers in 4G Networks, Communications and Multimedia Security, LNCS, vol. 3677, Springer, Berlin Heidelberg (2005): 220–229.

[28] Fischer, K., Karsch, S., Modelling Security Relevant Context - An approach towards Adaptive Security in Volatile Mobile Web Environments, WebSci11, June 14-17, 2011, Koblenz, Germany (2011).

[29] Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Operational Decision Support: Context-Based Approach and Technological Framework, Proceedings of the 5th International and Interdisciplinary Conference CENTEXT (2005).

[30] Michelberger, B., Mutschler, B., Reichert, M., A Context Framework for Process-Oriented Information Logistics, Business Information Systems, LNBIP 117, Springer, Berlin Heidelberg (2012): 260–271.

[31] Ksiezopolski, B., QoP-ML: Quality of Protection modelling language for cryptographic protocols, Computers & Security 31(4) (2012): 569–596.

[32] ISO/IEC 27005 Second edition 2011-06-01, Information technology Security techniques Information security risk management.

[33] NIST Special Publication 800-30, Revision 1, Risk Management Guide for Information Technology Systems (2012).

[34] Shoniregun, C.A., Impacts and Risk Assessment of Technology for Internet Security. Enabled Information Small-Medium Enterprises (TEISMES), Advances in Information Security, Volume 17, Springer Science+Business Media, Inc. (2005).

[35] NIST Special Publication 800-12, An Introduction to Computer Security: The NIST Handbook (1995).

[36] FIPS PUB 199, Standards for Security Categorization of Federal Information and Information Systems (2004).

[37] Menezes, A., van Oorschot, P., Vanstone, S., Handbook of Applied Cryptography, CRC Press, Boca Raton (1996).

[38] Freeman, R.L., Telecommunication System Engineering, Fourth Edition, John Wiley & Sons, Hoboken, New Jersey (2004).