Towards context-awareness security for mobile applications

Sofien BEJI, Yassine JAMMOUSSI
RIADI Laboratory
National School of Computer Sciences
Campus Manouba, Tunisia

Nabil EL KADHI
ECCE Department
Ahlia University
Bahrain

Abstract—**In standard computing, security is usually treated as a static requirement where counter-measures are mapped to attacks. In mobile computing, the heterogeneity of environment and the integration of resources lead to a dynamic adaptation of security mechanisms. This paper, firstly, attempts to define a new conceptualization of an abstract context which is relevant to security requirements. Secondly an analysis of the security mechanisms and challenges in the mobile world is given and adaptation rules are proposed in an XML based pattern. Finally security adaptation rules were integrated in a whole context-aware architecture.**

Keywords-component; Mobility; security; context-awareness;

I. Introduction

The widespread of mobile phone devices makes the security of applications taken as a guarantee for end users. Nevertheless, for mobile applications developers, securing personal data and transactions is a challenge. In fact, the mobile field is an integration of technologies ranging from hardware with device and smart card to software with OS and platforms through wireless telecommunication networks like 2G, 3G and recently 4G. Hence mobile secure applications should take care of the environment which is characterized by several types of factors such as the network type, the user, the device model and the involved technologies. All of these factors lead to the paradigm of context-awareness and especially context-aware security.

Context-awareness is an emerging aspect of future mobile systems. In particular, concepts like environment intelligence and ubiquitous computing rely on context information in order to personalize services provided to their end users [1]. Even though security is a relevant area for context-awareness applications, there are few works dealing with that domain in the mobility field. We try through this paper to contribute with a new conceptualization of context-awareness that will help the realization of security in the field of mobile applications. Our work is part of an approach that handles security during the process of mobile applications development.

The remainder of the paper is outlined as follows: Section 2 presents the state of the art of context-awareness. Section 3 is the motivation for the context-awareness in the case of mobile applications. Section 4 briefly presents our vision of the appropriate context model for mobile security. Section 5 is the adaptation layer where the main security counter-measures are analyzed. Finally section 6 and 7 deal with our architecture and process for context-aware security for mobile applications.

II. State of the art of context-aware systems

A. Context definitions

Context-aware applications are a fast growing research area involved in more than one field. However, the research seems still in its infancy. In fact, even the core definition of context is still ill defined due to the multiplicity and divergent definitions available in the literature. The originators of the term Context-awareness are Schilit and Theimer [4] who in 1994 introduced and defined Context-aware computing as 'the ability of a mobile user's applications to discover and react to changes in the environment they are situated in'[4]. The definition of Brown [7] is 'The elements of the user's environment that the computer knows about'. For Chen and Kotz [6] context is 'the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user'.

We adhere to the definition of Chen which is more explicit and relevant to our case. In this definition, the authors give special attention to states and settings of the environment which describe and characterize the context in the mobile field.

B. Categories of context

In addition to the large number of context definitions, there are some proposals for categorizing context-aware applications. A first classification given in [8] by Barkhuus and Dey deals with passive and active context-awareness. Active context-awareness adapts the device automatically whereas in the passive one, changes are communicated to the user who will decide for the application of actions. An other classification proposed by Mäntyjärvi et al in [9] gives three categories of context-aware applications: manual, semi-automated and fully automated. The manual and semi-automatic categories require user-interaction to preset device functionality and adaptation. In the third and ultimate level of automation, the device infers and proceeds actions without requiring user actions.

In [10], the authors deal with push and pull categorization. In a push type, the information is automatically delivered to the device when getting access to a given area. For the pull type, it the user activation is required for information inquiry. Table 1 summarizes the categories of context.

TABLE I. CATEGORIES OF CONTEXT

Context name	Context values
Device adaptation	Active, passive
User interaction	Manual, semi-automatic, automatic
Information delivery	Push, pull

Basically context-adaptation is performed at run-time but it could be extended to earlier stages in order to target the “write once and deploy many” paradigm. Thus, it is possible to inherit context adaptation from general to specific. Identifying possible alternatives is made at the development phase, choosing the possible ones at the installation and triggering the mechanism is made at run-time. In [14], an ontology for mobile applications security was proposed in order to assist developers during the choice of the appropriate counter-measures. The scope of the current work is turned to security context-awareness during run-time.

III. MOTIVATION OF CONTEXT-AWARENESS FOR MOBILE SECURITY

We believe that the mobile field security is one of the most areas where dealing with context-awareness is mandatory. In fact, several factors are against the regular deployment of standard security counter-measures. Firstly, we can observe a myriad of technologies that tend to arise a serious portability issue. In fact, there are different types and generations of mobile phones and networks with different performance and services. It is a challenge for a regular application development to target this wide field. Secondly, the mobility feature makes the device moving through heterogeneous environment where constraints and threats are continuously changing. Another argument is relative to users for whom several profiles are possible. All these features make the use and configuration of security mechanisms dependable of the context where they are deployed.

Before presenting our context-aware security system, we have to point out the relevant requirements, that such model should meet. Firstly, the model should implement the functional requirement of security adaptation rules at run-time according to context in a semi-automated and push mode. Secondly, our model should be independent of any specific use case. Finally, we should target the efficiency quality requirement by restricting resources to the built-in sensors and attributes of the mobile device.

According to [11], the architecture of a typical context-aware system or application should match the model of Sens, Reason and Actuate. Typically this is a four layered stack composed of the sensor layer, the semantic layer, the decision layer and the actuator layer.

The sensor layer requires the definition of the context attributes which are relevant to the security service. The semantic layer handles knowledge representation. The decision layer is the one dedicated to the choice of the appropriate actions in the identified conditions. Finally the actuate layer is the realization of the taken actions. Our contribution through this work will mainly target the abstract model of the sensor layer and the rules of the decision layer.

IV. A TYPICAL CONTEXT CONCEPTUALIZATION FOR SECURITY REQUIREMENTS

In this section we will present our conceptual model for a context-aware system that handles the security aspect for mobile applications.

The sensor layer includes the raw informations gathered from different sensors. Prior to this step, it is essential to define the categories and associated attributes of our context. Different contexts have been proposed in the literature and are summarized in [12]. A typical context sensor layer will include the category of the context, the attributes in that category, the value of the attribute and finally the confidence that describes the value probability. We believe that the selection of the appropriate categories and attributes is made up of its relevance to the application field. The dilemma of context definition is not within the identification of the most complete list of attributes but rather in the most appropriate ones. In fact, in the mobility field, resources are limited and access to external sensors is not always allowed. Ideally, scanning the environment to identify location and available networks is relevant to the security of applications but it harms processing and battery life-time. Among the context categories discussed in the literature, we believe that these are relevant to the security aspect: Device, Network, Task, User, Location, and Time. For each category, we can identify a set of attributes. Table 2 gives an overview of each category with the associated set of attributes.

TABLE II. CONTEXT-AWARE SECURITY CATEGORIES AND ATTRIBUTES.

Context Category	Context attribute
Device	Device.CPU Device.CryptographicCapabilities Device.MemorySize Device.BatteryLevel Device.InputType Device.OutputType Device.SmartCardType {SIM, UICC} Device.SmartCardSlotsNumber Device.CertificateUpdate
Network	Network.Type Network.Bandwidth Network.Delay Network.DisconnectionRate Network.Price Network.RoamingMode
Task	Task.Criticality Task.Timeliness
User	User.Language User.Age
Location	Location.Place
Time	Time.WorkingDay Time.DayOfMonth Time.Now

The device with the Smart card and the network are the typical instances of the physical context. Several relevant attributes like the device hardware configuration, the type of the smart card have been mentioned. Task, user location and time form the basis for the virtual context. We have also identified the user age as a relevant attribute due to the awareness level that could be inferred from the age. Being in public open space or inside the company buildings are different locations that can arise different threats and have an impact on the access control rules. In the same way, the day and time attributes are also important where access control depends on work-on or work-off days.

Based on the context categories of table 2, it is possible to compose a set of profiles relevant to security, each profile is identified by a context name. Thereby, a context_name is the aggregation of a set of Context_Attributes. A High_performance_Device context can be defined by a set of combined attributes like a high clock frequency, a large memory, a CPU with cryptographic features or the battery level.

V. The security adaptation layer

Since security is satisfied by a set of mechanisms, we will point out the adaptation regarding these counter-measures with the associated technologies. Mainly, we will be interested in Encryption, Public Key Infrastructure (PKI), access control models and authentication tokens as main counter-measures. Here below, the analysis of the deployment and the possible adaptation rules of each counter-measure.

A. Encryption and hash functions

Among the set of security services, symmetric and asymmetric algorithm target essentially confidentiality. A set of parameters can have a deep impact on the execution time and energy consumption of an algorithm. The developer has to play with the appropriate configurations according to the confidentiality requirements and the run-time conditions.

Among the observed features, we will be interested in the operation mode, the initialization vector (IV) and hash functions.

1) Operation modes

Operation modes specify how will encryption work. The encryption of block N could be dependent on the encryption of block N-1, or successive blocks may be independent of each other. A given cipher algorithm can be used in more than one operation mode. The widely used modes are Electronic Code Book (ECB) and Cipher Block Chaining (CBC) for block ciphering and Cipher Feedback (CFB), Output Feedback (OFB) and CounTeR (CTR) for stream ciphering.

With the variety of cipher algorithms and operation modes, a newbie software security developer is faced to a large set of combinations. DES/OFB/NoPadding for example means DES as symmetric encryption algorithm with OFB as an operation mode and with no padding for the last block. Another specification may be as follows: IDEA/CBC/ISO10126Padding.

To get back with some recommendations regarding the choice of an operation mode, we will firstly focus on cipher text errors and recovery which are frequent in wireless environment in comparison to wired infrastructure. From a survey [22] of the mentioned modes, the results recommend the use of OFB in error-prone environments like in wireless networks. Such a recommendation is due to the fact that a cipher text block error will only affect the corresponding plain text block. Moreover, it is recommended to use CTR mode for not synchronized environment because its efficiency to recover from such situation.

Regarding the execution time, a PDA benchmark [23] with a 256 key sized AES has shown that ECB is the appropriate choice. On the other hand, ECB is the weakest mode because identical plain text blocks are encrypted into identical cipher text blocks. Also CTR mode permits the parallelism but has a weakness in randomness of its counter. Consequently, ECB and CTR recommended only in case of low level confidentiality requirements.

- If Bad_Network Then Use OFB Mode
- If Timeliness And Low_Level_Confidentiality And Random_Source Then Use CTR Mode.
- If Timeliness And Low_Level_Confidentiality Then Use ECB Mode.
- If Random_Source Then Use OCB.

2) The initialization vector and randomness

Zero-filled, constant or pseudo random initialization vector can lead to dictionary attacks especially for a limited data payload. This flaw could be mitigated when supported by True number generation sources. In the context of a multimedia equipped device, sound and camera could be used as a source for randomness. According to [26], microphone and digital camera perform a high-rate sampling of physical sources which was used to produce random values with acceptable entropy.

3) Cryptographic hashing functions

Hash algorithms are one-way functions that turn an arbitrary message of bytes into a fixed-length digest. MD5, SHA-1, RIPE-MD are hashing algorithm used for integrity checking. In [23], the results show that MD5 (512 bits) is faster than SHA-512(512 bits).

For battery consumption, the benchmark in [24] shows that MD5 and SHA-1 are almost the same for messages less than 10K. Nevertheless, for large messages that are above 10K, MD5 is significantly better.

B. Access control model

Basically access control is the mechanism that accepts or rejects authorization from an entity to perform a service or get access to an object. A widely adopted access control method is the Role Based Access Control (RBAC) one. RBAC is implemented with the concepts of subject, role, object and operation. A subject belonging to a role can perform an operation on an object.

Among its large scale adoption by enterprise application developers, RBAC remains limited to the subject-oriented point of view. In fact, in the mobile field, we should take in account additional properties related to the environment like location, time and date in access control rules. A typical scenario is of an employee with a smartphone trying to get access to the enterprise resource planning system from an unknown country during a work-off day. In this situation, a context defined by the values of the Device.date, Device.time and Location.place attributes denies the access control of a subject S to an object O.

To overcome the RBAC model limitations, several context-aware access control models have been introduced, the Temporal-RBAC [28] which adds the time dimension and the dynamic RBAC model [27] based on dynamic roles.

In a larger scope, Generalized RBAC[12] which is an extension of the RBAC model, grants access to a subject role for an object role when a given context is activated. Policy access rules are hence defined in accordance to the subject, object and environment. In Attribute Based Access Control (ABAC)[29], permissions giving relies on attributes of subjects, resources, and environment

Since the mobile field is a distribute system, featured by a continuously changing context, we believe that ABAC is the appropriate access control model. A typical rule is the one that disable access to Wide Area Network in-side the company buildings or Multi-media files downloads in a network roaming mode.

C. *Authentication adaptation*

Any electronic authentication process is defined by the core concepts of factors, methods and channels. A claimant proves his identity to a verifier by proving possession of a token which is transferred through a channel. All the process is performed according to a method or a protocol. Particularly in the mobile model, there are several scenarios that can take place when dealing with authentication. Depending on the context and especially the device capabilities, a token could be a Subscriber Identity Module (SIM) Card stored key, a device certificate or even the user voice. We can resume the main authentication methods in 4 categories: password based, private key sharing, Public Key Infrastructure(PKI) and zero knowledge.

A typical password category includes: PIN, strong password, Scrambled PIN and One-Time-Password. The strength of the password is defined by the set of symbols that could be used. To avoid a key stroke attack, the 0-9 digits can be randomly arranged on a graphical screen, this is a scrambled PIN. For One-Time-Password, time-synchronization or hash function can be deployed. We can imagine two ways of adaptation regarding the token type and the authentication channel:

1) *Token type*

The main adaptation factor in this case depends on usability. Taking care of the device capabilities, we can deal with several types of token. Here are the different scenarios:

User Certificate if the device accepts certificate installation and if PKI is available.

Biometry if the device is enhanced with biometric capabilities.

A strong password if the device is a smart phone or a PDA with a large keypad.

A PIN for a limited keypad device.

2) *Token transport channel*

WAP 1.x and 2.0, SMS, Over-The-Air (OTA), Unstructured Supplementary Service Data (USSD) and Voice are different channels available for wireless network. An authentication token should be securely transferred to the verifier party. We present the list of possible solutions to transfer a password according to the available network and device capabilities:

- If device compliant with WAP 1.1 or WAP 1.2 Then Use WMLScript Cryptolib[25] with Signtext and Encryptext functions for Authentication and confidentiality.
- If device compliant with WAP 2.0 Then Use end-to-end security with TLS.
- If Not WAP coverage then Use SMS.
- If Not WAP coverage Then USE USSD. USSD is a GSM technology used to send text between the device and an application. There are push and pull modes where the latter is initiated by the device. For more information about UCCD, you can see the GSM 02.90 and 03.90 standards.
- If Not WAP coverage And high sensitive credentials then Use SIM Toolkit (STK) with Over-The-Air (OTA) protocol. The GSM 03.48 knowing as OTA can initiate a secure session with the SMS center. OTA is used by mobile network operators to update SIM Card content on the air.
- If Not WAP And Not OTA Then Use voice.

D. *Public Key Infrastructure (PKI) challenges*

PKI is the set of technologies, procedures and resources to implement and use asymmetric encryption. PKI plays a sensitive role for authentication. In this section, focus will be given to some issues regarding deployment of PKI in wireless and mobile environment. Firstly, we will be interested in certificate revocation where the size of CRL is not adequate for limited bandwidth environment. Next, we will discuss some constraints regarding key generation and store.

1) *Certificate validation*

The process of certificate validation requires at least two basic tasks. The path validation which consists in the confirmation of the identity of the public key holder and the certification path construction which consists in the building of the path of certificates from the root.

Due to device performance and network quality in a mobile context, the described process of path validation and certification path construction could be too complex. To overcome such a situation, the Server-Based Certificate Validation Protocol (SCVP) [21] could be used and the tasks of path discovery and path validation could be delegated to the server side. Hence, depending on the current context in terms of performance and network, the certificate validation could be treated locally or remotely by a relying party.

2) Certificate Revocation challenges

There is no unique method to check the validity of a certificate, among the used methods we can cite: Certificate Revocation List (CRL), Delta CRL, Certificate Revocation System (CRS) and Online Certificate Status Protocol (OCSP) [3]. Each method has advantages and drawbacks, and could be used in a given context when corresponding resources are available.

The CRL download procedure is featured by the freshness of certificate information but is tackled by its big size and thus not appropriate for bad network conditions. Delta CRL overcomes regular CRL constraints by limiting the provided data to the last CRL version updates. Anyway a first CRL download or an off-line installation is required. The main idea behind the use of CRS is to provide the requester only with the certificate status, this is an efficient and timeliness method that requires an on-line and fast network availability. Finally, the OCSP is a client-server protocol that may be used if additional informations about the certificate are required. We can briefly summarize our analysis in the list of rules below:

- If Good_Network And Good_Device Then Download CRL.
- If BAD_Network And Existing_CRL Then Download Delta_CRL.
- If BAD_Network And No_CRL Or Old_Version_CRL Then Use CRS with Certificate Revocation Status.
- If BAD_Network And Additional_Informations required Then Use OCSP.

3) Key generation challenge

Several practical studies have shown that key generation in asymmetric encryption is resource consuming especially for pervasive and mobile computing. A first local alternative to overcome the well-known RSA algorithm is the adoption of an efficient algorithm such as Efficient and Compact Subgroup Trace Representation (XTR) or the Elliptic Curve Digital (ECC) Signature Algorithm[17]. A second alternative to speed up RSA calculations is by outsourcing computing to external servers [16].

4) Key storage challenge

Private keys must be stored securely. In general, a private key should never be stored anywhere in plain text form. In standard computing, additional devices like USB drives may be used to handle the private key. In a mobile context, it is cumbersome to use additional devices with the mobile phone. Hence we will be limited to software and built-in solutions that may hold the private key.

More than one solution is possible. Firstly, we can use smart card applications such as the Wireless Identity Module (WIM) [19] which ensures that key pairs are generated inside the card and private keys never go outside. With new generation of smart cards, called Universal Integrated Circuit Card (UICC), it is possible to host multi applications on the same card. SIM, WIM or U(SIM)[18] are different javaCard applications (applets) stored on the same physical card.

Another solution is where keys are generated and stored by the device OS such as Symbian[30] or Windows Mobile[31]. A third solution slightly different from the second is where keys are not stored directly by the mobile phone OS but through a framework such as Java ME [19].

Each one of the above solutions could be deployed within some given conditions. It is knowing that a smart card is tamper resistant but a built-in applet like the WIM is required. We can summarize the key storage solutions in two classes, chip based and device based. Chip based is appropriate regarding the confidentiality of the private key and portability whereas device based is preferred where timeliness and efficiency is required.

VI. Context-aware security architecture

In the previous sections we have presented reactive behaviors in association with context-aware situations which can be represented by rules that follow the Event-Control-Action (ECA) pattern. ECA rules are also called condition rules and they have the form of if<condition> then <action>. There is more than a unique structure for expressing an ECA rule, we will adopt the structure below:

rule-name [in ruleset-name] [priority priority-val] [if condition] then action.

To be portable, extensible and cross-platform, an XML-like representation is required. The listing in Box 1 shows our Data Type Definition (DTD) of the security rules representation. The RuleSet is identified by a name and includes a set of rules. Each rule is represented in ECA pattern and has a priority. The context is nothing than a composition of conditions. Any condition is a logical expression with attributes and the associated values. For the action part, it deals with an action type, the target and pre-conditions if any.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!ELEMENT RuleSet (Rule+)>
<!ATTLIST RuleSet Name CDATA #REQUIRED>
<!ELEMENT Rule (Event?, Context, Action+)>
<!ATTLIST Rule Ref ID #REQUIRED>
<!ATTLIST Rule Priority  #IMPLIED>
<!ELEMENT Event (#PCDATA)>
<!ELEMENT Context (Log_Operator, Condition+)*>
<!ELEMENT Log_Operator EMPTY>
<!ATTLIST Log_Operator Value (OR|AND|NOT) "AND">
<!ELEMENT Condition (Context_Attribute, Comp_Operator, Attribute_Value)>
<!ELEMENT Context_Attribute (#PCDATA)>
<!ELEMENT Comp_Operator EMPTY>
```

```
<!ATTLIST Comp_Operator Op_Value (GT|LT|GE|LE|EQ|DF) "EQ">

<!ELEMENT Attribute_Value (#PCDATA)>

<!ELEMENT Action (Precondition*,(Target, Type)+)>

<!ELEMENT Precondition ANY>

<!ELEMENT Target (#PCDATA)>

<!ELEMENT Type (#PCDATA)>
```

Box 1. Security adaptation rule DTD.

To show the application of the DTD on an example, Box 2 is a snippet of a rule named CRL_Check that belongs to the rules set named PKI. The context is defined with two conditions related with an Or operator. The action part triggers the OCSP protocol.

```
<?xml version="1.0" encoding="UTF-8"?>

< RuleSet Name="PKI">

 < Rule Ref= "CRL_Check" Priority="2">

  <Event>Authentication</Event>

   <Context>

    <Log_operator Value=”Or”>

     <Condition>

       <Context_Attribute>Network.Bandwidth</Context_Attribute>

           <Comp_Operator >LT</Comp_Operator>

           <Attribute_Value>56</Attribute_Value>

     </Condition>

     <Condition>

      <Context_Attribute>Device.Processing</Context_Attribute>

           <Comp_Operator >EQ</Comp_Operator >

           <Attribute_Value>Low</Attribute_Value>

           </Condition>

        ...

       </Operator>

      </Context>

        <Action>

       <Precondition>CRS is supported</Precondition>

       <Target>CRS Protocol</Target>

       <Type>Use</Type>

...

 </Rule>

</Ruleset>
```

Box 2. XML example of security adaptation rule.

The security rules are nothing else than a component of a context-aware system architecture. To avoid building an architecture from scratch, a survey of existing context-management models leads to three categories: widget based[32], client-server[33] and blackboard[2] based. The widget based model is adopted from the architecture of graphical user interfaces. The main concept behind this design is the separation between context acquisition and its use.

In the client-server model, there is no central manager, each component has its own capabilities to sense, reason and actuate. Even tough the model is flexible and adopts standard coding and protocols, the components suffers from their complexity.

The blackboard model is a data-centric one where the main idea is the subscription of the client applications to receive messages matching a specified pattern. By this way, only relevant contexts will trigger new actions for the associated client applications. Since we target, a layered architecture, a non-domain specific solution that handles context management, the context framework proposed by Korpipää[5] seems to be the appropriate choice. In fact, our proposal is inspired from the context framework which is a blackboard based model that integrates an XML script engine for action execution. The proposed context-aware security architecture integrates a main component, the context manager which stores contexts in a database and handles the publish and subscribe mechanism of the blackboard technique. The context sources collects data from several sources. Raw data are transformed to an easily readable pattern through the context abstractor. To identify context switching, the change detector provides the components chain with relevant changes. The security adaptation rules are stored in a repository which is connected to the activator.

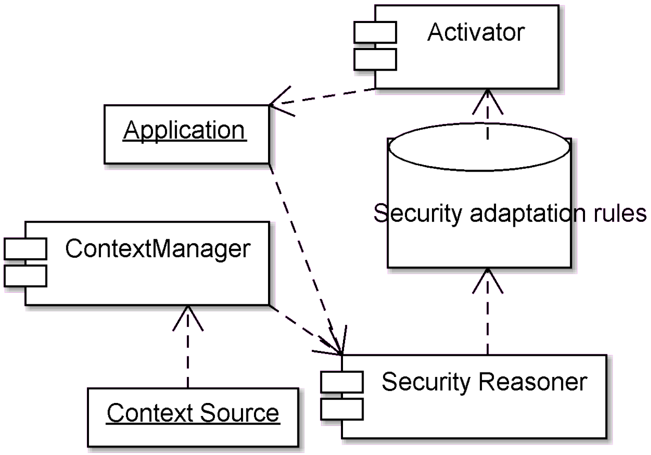

Figure 1. Context-aware security architecture.

VII. Context-aware security process

According to the architecture in fig 1, we can outline the process of security adaptation according to a sequence of steps. In fact, we can identify two distinguish triggering events, the former starts from the environment and the latter from the application. Consequently our system can act in two ways, in proactive and reactive. In the reactive way, the application requests a security service like confidentiality or authentication, the change detector will then communicate with the context manager which will fetch for the subscribed client applications. Once the application identified, the security reasoner will identify the set of rules relative to the given context with the security service required. The set of security adaptation rules will be then executed by the activator to provide the application with the appropriate security configuration. In the proactive way, a protective security configuration is applied to the device when a given context is identified.

VIII. Conclusion

Throughout this article we have firstly presented context-aware systems, than argued that mobile applications security is one of the area of context-awareness. At a second stage, we have presented a conceptualization of the mobile context security through the proposal of the relevant categories and attributes. Environment changes and mobile device limitations were analyzed and security adaptation rules were proposed through an XML representation. The presented context security rules have been integrated in a whole architecture, the context framework. Finally, the process of context-aware security has been described within the framework components. Our next step is the implementation of the architecture and its test in real conditions.

References

[1] K. Wrona and L. Gomez, "Context-aware security and secure context-awareness in ubiquitous computing environments", XXI Autumn Meeting of Polish Information Processing Society ISBN 83-922646-0-6 Conference Proceedings, pp.255-265, 2005.

[2] T. Winograd, Architectures for context. Human-Computer Interaction, Vol. 16, No. 2, 3 & 4, pp. 401-419, 2001.

[3] J.L. Muñoz, and J. Forné, “Certificate Revocation Policies for Wireless Communications,” Proceedings of the IASTED International Conference on Communication Systems and Networks. ACTA Press, 2002, pp. 427-432.

[4] B. N. Schilit and M. M. Theimer. “Disseminating active map information to mobile hosts,” IEEE Network 8(5): September/October 1994 pp. 22-32.

[5] P. Korpipaa, Blackboard-based software framework and tool for mobile device context awareness, Ph.D thesis, University of Oulu, 2005.

[6] G. Chen and D. Kotz, “A Survey of Context-Aware Mobile Computing Research,” Dartmouth Computer Science Technical Report TR2000-381.

[7] P. Brown, J. Bovey, and J.D Chen, “Context-aware Applications: From the Laboratory to the Marketplace,” 1997.

[8] L. Barkhuus and A. Dey “Is Context-Awareness Taking Control Away from the User? Three Levels of Interactivity Examined,” In Proceedings of Ubicomp 2003: 159-166.

[9] J. Mäntyjärvi, U.Tuomela, I. Känsälä and J. Häkkilä “Context Studio – Tool for Personalizing Context-Aware Application,” in Mobile Terminals Proc: OZCHI, 2003: pp. 64-73.

[10] K. Cheverst, K. Mitchell and N. Davies, “Investigating Context-Aware Information Push vs. Information Pull to Tourists,” In Proc. of MobileHCI’01, 2001.

[11] N. Baker, M. Zafar, B. Moltschanov, and M. Knappmeyer, “Context-aware systems and implications for future Internet,” In Future Internet Conference and Technical Workshops, 2003.

[12] J. M. Matthew and Mustaque Ahamad, “Generalized role based access control,” In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS). USA, April 2001.

[13] J. Indulska and K. Henricksen. Context-Awareness. Engineering Handbook on Smart Technology for Aging, Disability and Independence, A. Helal, M. Mokhtari and B. Abdulrazak, Editors, John Wiley & Sons. ISBN 0471711551, Computer Engineering Series, 2007.

[14] S. BEJI, and N. El Kadhi, "Security Ontology Proposal for Mobile Applications," MDM 2009, Proceedings of Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, pp. 580-587, 2009.

[15] J. Hong, and J. Landay. An infrastructure approach to context-aware, computing. Human-Computer Interaction, Vol. 16, No. 2, 3 & 4, pp. 287-303, 2001.

[16] Y. Lei, D. Chen, and Z. Jiang, Generating digital signatures on mobile devices. 18th International Conference on Advanced Information Networking and Applications, AINA 2004. Volume 2, 29-31, pp. 532-535, 2004.

[17] T. Limmer, F. Dressler, and R.Gonzalez, "Evaluation of real-time aspects of multiparty security on low-power mobile devices", Springer Berlin Heidelberg, pp. 71-80, 2006.

[18] K. Mayes, and K. Markantonankis, "Smart cards, tokens, security and applications", Springer, pp. 51-73, 2008.

[19] M. J. Yuan, “Entreprise J2ME, DEVELOPING MOBILE JAVA APPLICATIONS“, Ed. Upper Saddle River: Prentice Hall PTR, pp. 20-25, 2006

[20] D. Costa et al, “Architectural Patterns for context-Aware Services Platform,” In Proceedings of the Second International Workshop on Ubiquitous Computing, IWUC, 2005.

[21] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)", RFC 5055, December 2007.

[22] Y. W. Law, J. Doumen and P. Hartel, "Survey and Benchmark of Block Ciphers for Wireless Sensor Networks", ACM Transactions on Sensor Networks (TOSN), pp. 65-93, USA, 2006.

[23] A. Ramachandran, Z. Zhou, and D. Huang, "Computing Cryptographic Algorithms in Portable and Embedded Devices", IEEE International Conference on Portable Information Devices, 2007. PORTABLE07,pp. 1-7 Orlando, 2007.

[24] D. Shah, and S. Zhong, Benchmarking Security Computations on Wireless Devices, Technical report, University at Buffalo, Buffalo, New York, USA, 2006.

[25] WAP Forum, WMLScript Crypto Library, WMLScript Crypto Library Specification, 1999.

[26] J. Bouda, J. Krhovjak, V. Matyas, and P. Svenda, "Towards True Random Number Generation in Mobile Environments", Proceedings of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age, OSLO, pp: 179 – 189, August 2009.

[27] G. Zhang, and M. Parashar, ”Context-Aware Dynamic Access Control for Pervasive Applications”, In Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2004), Western MultiConference (WMC), San Diego, CA, USA, January 2004.

[28] E. Bertino, P. A. Bonatti, and E. Ferrari . TRBAC: A Temporal Role-Based Access Control Model. ACM Transactions on Information and System Security, 4(3), pp. 191-223, 2001

[29] T. Priebe, E. B. Fernandez , J. I. Mehlau, and G. Pernul : A Pattern System for Access Control. Proc. 18th Annual IFIP WG 11.3 Working Conference on Data and Application Security, Sitges, Spain, July 2004.

[30] M. j. Jipping, Smartphone Operating System concepts with Symbian OS, Wiley, 2007.

[31] F. McPherson, Windows Mobile, How to Do Everything with, Mc-Graw-Hill, 2006.

[32] A Dey, Providing architectural support for building context-awareapplications. Ph.D. dissertation, Georgia Institute of Technology, 2000.