UCCSSM: Ubiquitous Computing Context-aware Service Supply Mechanism

Daoqing Sun, and Qiying Cao

Abstract**—**Ubiquitous computing systems typically have lots of security problems in the area of context-aware service supply by means of classical PKI methods. The service types and service levels, the difference between different environments, the security group authorization and delegation authorization of the services, the identity hiding, the collection and application of the current and the history location and activities of the principal and its neighbor etc are all these unsolved problems. In this paper, UCCSSM, a new novel SPKI-based ubiquitous computing context-aware service supply mechanism is presented to solve these problems. SPKI-based service authorization and service supply are used in UCCSSM to solve above problems while binding these services to the principal using service authorization certificate. By classifying and analyzing the service environment and service resources, the three context-aware levels are given to suit the different needs of context-aware services. The service authorization and service sensing processes, the services supply processes etc of UCCSSM are described in the paper. The performance analysis shows that UCCSSM is a suitable security solution in context-aware ubiquitous computing environments.

Keywords**—**Context-aware, Security, Service Supply Mechanism, SPKI/SDSI, Ubiquitous Computing.

I. Introduction

According to the viewpoints of Werser [1], the father of ubiquitous computing, the suitable services will be automatically provided when a principal (mobile user or ingoing entity) enters a new ubiquitous computing environment. But, in this process, there are lots of unsolved security problems, such as the service types and service levels, the difference between different environments, the security group authorization and delegation authorization of the services, the identity hiding etc. How to solve them will be a key security problem in ubiquitous computing environment.

SPKI-based service authorization and service supply are ideally suitable method for decentralized ubiquitous computing environment in solving the service supply to the principal (the mobile user or the ingoing entity) security while binding these services to the principals. It can solve lots of these unsolved security problems by means of the well-known identification technology like PKI with the support by X.509 [2].

In order to provide more and more suitable services to a principal, the contexts of this principal must be considered. These contexts include the current and the history location and activities of the principal and its neighbor etc. A solving method is to combine the service supply with the principal's context. In this paper, by classifying and analyzing the service environment and service resource, the three context-aware levels are given to suit the different need of services.

Therefore, a new novel SPKI-based ubiquitous computing context-aware services mechanism, named UCCSSM, is presented to solve them in this paper. It is our main innovation.

The paper is organized as follows. Introduction of SPKI/SDSI is given in Section II. A description of related work is provided in Section III. Then, a description of context -aware level is provided in Section IV. Afterwards, the ubiquitous computing service authorization and service supply mechanism, UCCSSM, are presented in Section V and section VI respectively. At last, the performance analyses of UCCSSM are shown in section VII before a conclusion of the paper is given in section VIII.

II. SPKI/SDSI

Simple Public Key Infrastructure (SPKI), which is based on the Simple Distributed Security Infrastructure (SDSI) presented in 1996 by R. Rivest et al. [3], has been proposed as a standard in the RFCs 2692 [4] and 2693 [5]. It provides two main features: a set of tools for describing and delegating authorizations and an infrastructure. These can enable the ubiquitous computing system to create a local namespace, which can be integrated into any global namespace at the same time.

In contrast to other PKIs like X.509, every principal in SPKI owns at least one asymmetric key pair whose public key identifies the principal globally. Classical PKIs provide name certificates, which bind names to public keys for authentication. SPKI, however, also provides authorization certificates that bind authorizations (e.g. the right to use ubiquitous computing service resources) to public keys. These two features are used in UCCSSM to bind the authorizations of ubiquitous computing service resources to a principal's key. Furthermore, authorizations can be delegated totally or partially to other principals. Through a series of delegations it is

Manuscript received January 21, 2007: Revised version received March 1, 2007.

Daoqing Sun is with the College of Information Sciences and Technology, Donghua University, Shanghai 201620 China (e-mail: sundq@mail.dhu.edu.cn). He is also with the College of Mathematics and Computer Science, Anhui Normal University, Wuhu 241000 China (e-mail: sundq@mail.ahnu.edu.cn).

Qiying Cao is with the College of Information Sciences and Technology, Donghua University, Shanghai 201620 China (corresponding author to provide phone: +86 21 62378632; e-mail: caoqiying@dhu.edu.cn).

possible to build certificate chains.

Authorizations in SPKI are formulated by using so-called tags, which are defined with S-expressions [6]. A set of rules defines how to intersect tags when delegating an authorization. An authorization certificate contains a flag that indicates whether the principal's authorization is allowed to delegate to others. When delegating an authorization, the principal is allowed to constraint the rights given or to delegate them fully. However, it is not allowed to expand them [7].

The service authorization can also be award to a group when a local name certificate is used and the subject item of authorization certificate is a public name. This can be applied in UCCSSM to predigest authorization process.

The service authorization can also be award to a principal that wants to protect its privacy when a pseudonym is used to name the principal's certificate.

Thus, SPKI-based authorization is an ideal method for decentralized ubiquitous computing environments.

III. Related Work

While we research ubiquitous computing service model, UCCSSM, acknowledgments are made to the related work shown as follows:

In the PAYFLUX [7], K. Herrmann et al. presented decentralized creation and delegation of authorizations based on SPKI/SDSI in the electronic payment systems.

In order to solve the problem of authorization in the tag item of SPKI/SDSI, M. Dam [8] gave a sound and complete inference system for a fragment of the regular language of SPKI/SDSI, which was decidable in polynomial time. He also put forward how to use the extended syntax to represent constrained delegation in SPKI/SDSI.

C. Doulkeridis et al. [9] presented architecture for context-aware service discovery. He described the increase of the quality of service discovery when context-aware is taken into account and the extra cost/burden imposed by context management.

U. Hengartner et al. [10] presented a distributed, certificate-based access-control architecture for context-aware services that avoids privacy violations by means of access-rights-graphs-based algorithm.

J. Hoebeke et al. [11] compared the performance of proactive and reactive discovery protocol and gave some guidelines for developing new resource and service discovery protocols or extending existing ones.

In our previous research [12], we have presented a simple SPKI-based ubiquitous computing service model in solving the security problems occurring in ubiquitous computing environments.

IV. Context-aware Level

A. K. Dey [13] presented a general definition of a context-aware system: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task." According to this definition, we can describe the context-aware level as follows.

A. Basic Context

The basic context of a principal includes four primary context types for characterizing the particular situation of a principal. They are identity, location, activity and time.

Where

1) *Identity:* the identity of the relevant principal.
2) *Location:* the geographical position of the relevant principal.
3) *Activity:* the activity or activities being performed.
4) *Time:* the time period at which the principals perform the activity.

B. Extended Context

The extended context of a principal includes its current neighbor principals for characterizing the particular surrounding situation of this principal. Each neighbor's context can also be described with its current identity, location, activity and time too.

C. Developmental Context

The developmental context is the developed history of the basic context of the principal and its neighbors.

D. Context-aware Level

The context-aware level is decided by analyzing the service environment and the service resource. Three context-aware levels are given to suit the different needs of service.

1) *Low*: in this case, the service provided must be associated with the basic context of the principal.
2) *Middle*: the service provided must be associated with the basic context of the principal and the extended contexts of its neighbors.
3) *High*: the service provided must be associated with the basic context of the principal and its neighbors. The context history of the principal and its neighbors are also necessary in this case.

V. Services Authorization

A. Definitions

Next, the basic definitions are given. They will be used in UCCSSM later:

1) *SSS*: Services Supply Server, for service certificate authorization and/or service supply in ubiquitous computing systems. The validation of certificate and the decision-making of service control are also provided by the SSS.
2) *CS*: Certificate Server, for providing the certificate conservation, the search of certificate chain and online validation etc.
3) *SCL*: Services Control Lists, for providing the sort lists and level lists of the available services that can be updated dynamically.
4) *CD*: Certificate Database.

5) *SD*: Sensing Database, used for saving the resource sensing data that includes name, location, state, access permission and service environment of the resources.
6) SSS_A: *SSS* used for certificate authorization, which grants the service certificate to its principal directly or indirectly. Such *SSS* is the original ubiquitous computing environment of the principal.
7) SSS_S: *SSS* used for service supply, which provides the services to the mobile users or the ingoing entities. Such *SSS* is the new ubiquitous computing environment. When the principal enters this new environment, it needs services from the environment. SSS_A and SSS_S can be the same one when the principal belongs to SSS_S and gets its service authorization certificate from the SSS_S.
8) CS_A: Certificate Server, used for supporting certificate authorization of SSS_A.
9) CS_S: Certificate Server, used for supporting service supply of SSS_S.
10) SCL_A: Service Control List of $SSS_{A.}$
11) SCL_S: Service Control List of $SSS_{S.}$
12) CD_A: Certificate Database of CS_A.
13) CD_S: Certificate Database of CS_S.
14) CD_P: Mini Certificate Database of the mobile user or the ingoing entity.
15) *LP*: Location Proxy, used for discovering the current location of the given principal.
16) *NDP*: Neighbor Discovery Proxy, used for discovering the identification and the location of the given principal's neighbors. The neighbor's information includes its identification, state, the relation to the services principal etc.
17) *HP*: History Proxy, used for recording the history of the principal. During the service supply process, the context of the principal needs to be classified and saved in the database of *HP*. The analysis of the historical contexts can also be performed by the *HP*.
18) HP_D: Database of *HP*.

B. *Service Authorization Process*

Every principal (here, the principal indicates SSS_A, CS_A, mobile user, ingoing entity or various authorization agents) owns at least one asymmetric key pair whose public key identifies the principal globally. The SSS_A awards services authorization certificates to the prime agents to indicate the sorts of services and their valid lifetime. The prime agents can award its authorization certificates totally or partially to the successive agents one by one until the terminal principal to indicate the sorts of services and their valid lifetime too. The prime agent and the successive agent can be the mobile user or entity. Every new service authorization certificates should be sent to the CS_A and saved in the CS_A. The principal can also save its service authorization certificates into its mini CD_P. This is an optional operation to the principal according to its own need.

C. *Service Sensing*

The SSS_S performs the service sensing automatically and periodically. The location, state, access permission and service environment of the resources are sensed by the SSS_S and saved in the *SD*.

VI. Service Supply Mechanism

A. *Service Supply Process*

The service supply process is shown in Fig. 1, where:

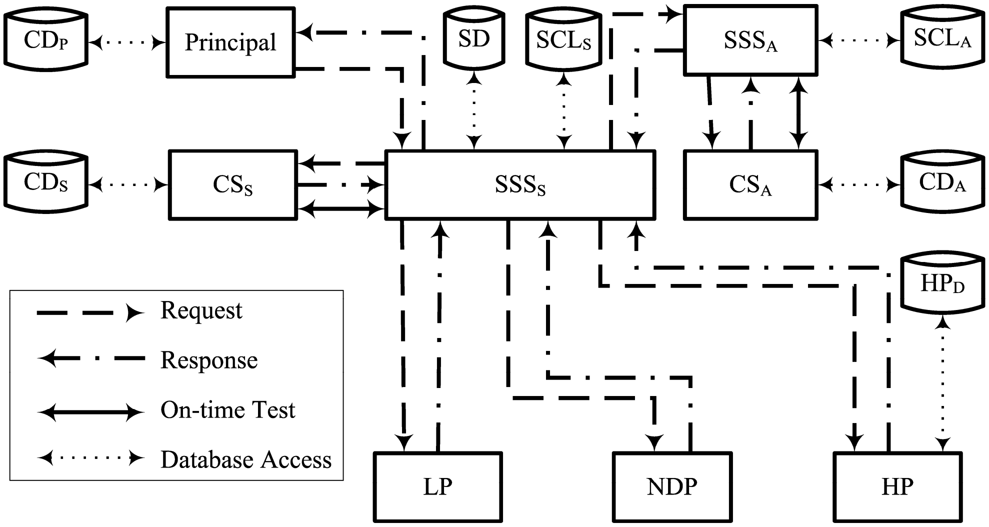

Fig. 1. Ubiquitous Computing Context-Sensitive Service Mechanism

1) *Service request*: the principal (the mobile user or the ingoing entity who wants to enjoy the service from the ubiquitous computing system) searches for its CD_P to look for the suitable service certificate chain. If the chain exists, the principal signs it by using its private key and then sends the signed service certificate chain to the SSS_S. If the suitable service certificate chain does not exist, the principal will send its signed service request to the $\mathrm{SSS_S}$

directly.

2) *Validation of service request*: after having received service request from the request principal, if the certificate chain exists, the validation of this SPKI certificate will be performed. If the certificate chain does not exist or does not pass through the validation of the *SSS_S*, the *SSS_S* will try to find the new suitable service certificate with the help of the *CD_S* or by means of searching the principal's authorization domain, the *SSS_A*, for identification when the domain finds the principal does not belong to it.

3) *Context-aware Level:* After the service request has been accepted, the context-aware level is decided by analyzing the service environment and service resource. Three context-aware levels are given to suit the different needs of service. They are Low, Middle or High. Some examples are given as follows to show how to analyze the context-aware level.

When the sort of service environment is "Home", the service supply will be the housemaster or his/her authorized guest. In this case, the system will just locate this man with his/her current activity and related available resources. The context-aware level is "Low" certainly.

When the sort of service environment is "Work", the system will decide the enjoyed services according to the principal's duty and his/her current activity. When the work is cooperative, the current work results and current state of his/her colleague ("neighbor") will be used in the meantime. So, the context-aware level is "Middle" now.

When the sort of service environment is "Health", the current illness of the patient will be examined. The history of this patient's illness is also needed to the doctor. If this illness is genetics, the histories of patient family numbers are needed to the doctor as well. The context-aware level is "High" then.

4) *Response with "Low" awareness level service request*: The *SSS_S* tries to find the current location and activities of the principal that will enjoy the service by means of the help of *LP*. And then the *SSS_S* searches the available resources from the *SD* in the area of the principal's location assisted with the activities. At last, the *SSS_S* will provide the services to the principal.

5) *Response with "Middle" awareness level service request*: The *SSS_S* try to find the current location and activities of the principal that will enjoy the service by means of the help of *LP*. And then the *SSS_S* try to find the identification, the current location and activities of the principal's neighbor by means of the help of *NDP*. Next, the *SSS_S* searches the available resources from the *SD* in the area of the principal's location assisted with the activities. The related and necessary resources of its neighbor are also obtained by the *SSS_S*. At last, the *SSS_S* will provide the services to the principal.

6) *Response with "High" awareness level service request*: The *SSS_S* try to find the current location and activities of the principal that will enjoy the service by means of the help of *LP*. And then the *SSS_S* try to find the identification, the current location and activities of the principal's neighbor by means of the help of *NDP*. Next, the *SSS_S* searches the available resources from the *SD* in the area of the principal's location assisted with the activities. The related and necessary resources of its neighbor are also obtained by the *SSS_S*. The history information of the principal and its neighbors are obtained to the *SSS_S* by means of the help of the *HP* from the *HP_D*. At last, the services will be provided from the *SSS_S* to the principal.

B. Some More Discussions

1) *The first interaction between the SSS_S and the mobile user or the ingoing entity*: in order to communicate correctly, the first interaction will do some special work. When the mobile user or the entity knows the public key of *SSS_S*, it can use the public key of the *SSS_S* to encrypt its SDSI name or the public key of its *SSS_A* to show its identification or its domain's identification to *SSS_S*. When the mobile user or the entity does not know the public key of the *SSS_S*, it can also show its identification to *SSS_S* directly without any encryption. Then, the mobile user or the entity will wait the *SSS_S* to establish the communication link between them.

2) *The public key of SSS_A*: the *SSS_S* must check whether it owns the public key of *SSS_A* or not. If the public key does not exist, the *SSS_S* must obtain it from the remote domain of the *SSS_A*.

3) *The validation algorithm of authorization certificate chain*: the validation algorithm of authorization certificate chain includes the examination of the Certificate Revocation List (CRL) or the Timed Revalidations (TR) or Onetime Revalidations (OR). We can select one of them according to the actual scenario.

4) *Reconstruction of the new service authorization certificate chain*: if the certificate chain does not exist, the *SSS_S* will use a depth-first-search [14] to determine whether there is a path from the *SSS_S* or the *SSS_A* to the principal according to the principal's service request.

5) *Reduction of certificate:* when one new service authorization certificate chain is reconstructed, the reduction of the certificate chain will be done. J. Biskup et al. [15] gave some algorithms of name certificate reducing closure and extended certificate chain discovery. They are suitable in solving the correlative problems in UCCSSM.

6) *Authorization policy and S-expression*: the *SSS_A* decides the service authorization sorts and definitions by itself. The mobile user brings the service authorization certificates into *SSS_S*, a new ubiquitous computing environment. But the new environment, the *SSS_S*, decides that how to explain the service authorization certificates and what sorts of services will be supplied according to its own judgments. The detail of service authorizations is described with the S-expression [6] in the Tag item of certificate.

VII. Performance Analyses

In this section, we will indicate how to use UCCSSM to solve those problems presented in Section 1.

A. Service Types and Service Levels

We define different service types and service levels, and then give them different authorization to different principals through the service authorization certifications. So, the problems of service types and service levels have been solved in UCCSSM. But in the classical PKI systems, it is impossible to handle them.

B. The Difference between Different Environments

As discussed in the previous section, the SSS_A decides the service authorization types and definitions according to its ubiquitous computing environment during authorizing. But the new environment, the SSS_S, decides that how to explain the service authorization and what types of services will be supplied according to its own judgments when supplying service.

C. Security Delegation and Group Authorization

The principal can delegate its authorizations totally or partially to the other principals using the principal's asymmetric key pair.

The service authorization can also be award to a group when a local name certificate is used and the subject item of authorization certificate is a public name.

So, the delegation authorization and group authorization process based on SPKI/SDSI through the certificate chain is secure and feasible in UCCSSM.

D. Identity Hiding

As already discussed in Section 2, the SSS_A can use pseudonym to name its principal's certificate. Therefore, the principal's actual figure can be hidden in the new ubiquitous environment of SSS_S when the services are secure provided.

E. The Collection and Application of the Current and the History Location and Activities of the Principal and its Neighbor

As discussed in the section 4 to section 6:

Firstly, the SSS_S try to decide the coming service's context-aware level according to the principal's service request certificate.

Secondly, the *LP* will try to find the principal's location, and the *NDP* will try to find the information of the principal's neighbor when needed.

Among the service supply processes, the *HP* will try to classify and save the correlative information.

Thus, the SSS_S can provide suitable context-aware services to the needed principal.

Therefore, we believe that UCCSSM is a suitable security solution in context-aware ubiquitous computing environments.

VIII. Conclusions

This paper has presented a novel ubiquitous computing context-aware service supply mechanism based on SPKI/SDSI, called UCCSSM. We can benefit from UCCSSM that it provides a secure and feasible mechanism for solving service supply problems of the mobile user or the entity with context in ubiquitous computing environments.

References

[1] M. Weiser, "The Computer of the 21st Century," *Scientific American,* vol. 265, no. 3, 1991, pp. 66-75.

[2] ITU-T, "Recommendation X.509 (1997 E): Information Technology – Open Systems Interconnection – The Directory(1997)," *ISO/IEC* pp. 9594-9598, 1997.

[3] R Rivest ,et al., "SDSI :A Simple Distributed Security Infrastructure," *http://theory.lcs.mit.edu/~rivest/publications.html,* 1996.

[4] C. Ellison, "SPKI Requirements," *RFC 2692.* September 1999.

[5] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI Certificate Theory," *RFC 2693.* September 1999.

[6] R.L Rivest, "SEXP: S-expressions," *http://theory.lcs.mit.edu/~rivest/sexp.html,* 2002

[7] K. Herrmann, and M.A. Jaeger, "PAYFLUX - Secure Electronic Payment in Mobile Ad Hoc Networks," *Information and Communications Security. 6th International Conference, ICICS 2004. Proceedings (Lecture Notes in Computer Science vol. 3269)* pp. 66-78 2004

[8] M. Dam, "Regular SPKI," *LECT NOTES COMPUT SC 3364:* pp. 134-152, 2005.

[9] C. Doulkeridis, N. Loutas and M. Vazirgiannis, "A System Architecture for Context-Aware Service Discovery," *Electronic Notes in Theoretical Computer Science,* Volume 146, Issue 1, 24 January 2006, Pages 101-116

[10] U. Hengartner and P. Steenkiste, "Avoiding privacy violations caused by context-sensitive services," *Pervasive and Mobile Computing,* Volume 2, Issue 4, November 2006, Pages 427-452.

[11] J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester, "Analysis of decentralized resource and service discovery mechanisms in wireless multi-hop networks," *Computer Communications,* Volume 29, Issues 13-14, 21 August 2006, Pages 2710-2720

[12] D. Sun, J. Pan, Q. Cao, T. Li, and F. Yang, "Ubiquitous Computing Service Model Based On SPKI/SDSI," *Dynamics of Continuous, Discrete and Impulsive System, Series B,* Vol. 13E, No. 5, 2006, pp. 2218-2223.

[13] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing* 5:20-24.

[14] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms,* MIT Press/McGraw-Hill, 1990.

[15] J. Biskup, and S. Wortmann, "Towards a CredentialBased Implementation of Compound Access Control Policies," *Proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT 2002) Proceedings on the Ninth ACM Symposium on Access Control Models and Technologies, SACMAT.* vol. 9, 2004.

Daoqing Sun is a PhD candidate in the College of Information Sciences and Technology at Donghua University, Shanghai, China. He received his Bachelor degree from the Petroleum University, Dongying, China in 1988, and obtained his Master degree from the Beijing Institute of Technology,Beijing, China in 1991. He is an assistant professor and master's advisor of computer science and technology at Anhui Normal University, Wuhu, China. His research interests include ubiquitous computing and computer network security. Contact him at 33-3-702, Changjiangchang Modern Area, 2 South Zhongshan road, 241000 Wuhu, China. Email: sundq@mail.ahnu.edu.cn