UCAIPM: Ubiquitous Computing Agile Information Protection Mechanism

Daoqing Sun, and Qiying Cao

***Abstract*—**In order to improve the security of ubiquitous computing system, UCAIPM, a new novel ubiquitous computing agile information protection mechanism is presented. The protected information includes the identity, location, activity of the entities and their neighbors, the service resource like the currently available devices, the schedule like the developing history of the entities and their neighbors. The SPKI-based authorization technology is used in the mechanism to solve the identification validation, security communication and the security level division. The D-S Evidence Theory is used to compute the trust value for deciding the trust level from the service provider to the ingoing user.

***Keywords*—**Agile Information Protection, Evidence Theory, Security, SPKI/SDSI, Ubiquitous Computing

I. Introduction

According to the viewpoint of Mr. Weiser [1], when an entity, such as a mobile user, enters a ubiquitous computing environment, it should enjoy the autonomous and transparent services provided by the environment. During these services providing process, lots of agile information may take part in this process. This agile information may include the identity, location, activity of the entities that provide the services and their neighbors. The service resource like the currently available devices, the schedule like the developing history of the entities and their neighbors are also agile information. This information may be important privacy information. So, how to protect them will be a key security problem in ubiquitous computing environment.

In order to provide security service, SPKI-based authorization technology is used in our mechanism. It can solve lots of the problems such as the identification validation of the visiting user who wants to enjoy the ubiquitous computing system's service, security communication etc. By means of the help of SPKI server, we can distinguish the visiting user's security level. The SPKI service certificate is also the main medium during the whole communication process.

D-S Evidence Theory is a suitable method to solve the computing problems of uncertainty information in ubiquitous computing environments. It is used in our mechanism to compute the trust value from the entity to the visiting user, whereas the former will provide service to the latter. And, with the help of the risk evaluator, the trust level from the entity to the visiting user can be decided.

Therefore, a new novel ubiquitous computing agile information protection mechanism, named UCAIPM, is presented in this paper. It is based on SPKI/SDSI and D-S Evidence Theory. 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 ubiquitous computing agile information protection mechanism, UCAIPM, is presented in Section IV. Afterwards, some implement examples of UCAIPM are shown in section V before a conclusion of the paper is given in section VI.

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. [2], has been proposed as a standard in the RFCs 2692 [3] and 2693 [4]. It provides two main features: a set of tools for describing and delegating authorizations and an infrastructure. These can enable the ubiquitous computing environment, the mobile user and the ingoing entity 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 [5], every principal (i.e., ubiquitous computing environments, mobile entities or visiting users) 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 to public keys. These two features are used in UCAIPM to bind the authorizations of ubiquitous computing identification or service resources to a principal key. Furthermore, authorizations can be delegated totally or partially to other mobile entities or visiting users. Through a series of delegations it is 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.

Manuscript received January 21, 2007; Revised received November 9, 2007.

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

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 authorizations 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 UCAIPM to predigest authorization process.

The service authorizations 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 small-scale decentralized ubiquitous computing environments.

III. Related Work

While we research ubiquitous computing agile information protection mechanism, acknowledgments are made to the related work shown as follows:

Dempster and Shafer were the fathers of the classical D-S Evidence Theory [8, 9], which was suitable method to solve the related uncertain information computing in the ubiquitous computing environments.

V. Cahill et al. [10] presented a method that can solve the secure collaboration problems by combining the evidence, trust computing and risk evaluator in uncertain environments.

In ubiquitous computing environments, D-S Evidence Theory was used to solve the trust computing problems successfully in our previous research [11].

In order to solve the problem of authorization in the tag item of SPKI/SDSI, M. Dam [12] 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.

P. G. Argyroudis et al. [13] presented extended SPKI certificate examples in their Aether system.

In ubiquitous computing environments, SPKI/SDSI technology was used to solve the service supply problems successfully in our previous research [14].

However, how to protect the ubiquitous computing agile information is still an open issue. We expect UCAIPM can do some foundational work in solving the related ubiquitous computing security problems.

IV. Mechanism of Agile Information Protection

The mechanism of agile information protection is shown in Fig. 1.

A. Service Request

The service request comes from the *VU* (*Visiting User, the ingoing user who may come from outside and need to enjoy the service provided by the ubiquitous computing environment*). The *VU* checks whether the suitable SPKI service certificate chain exists or not. If the chain exists, the *VU* will sign it by using its private key and then sends its signed chain to the *AIPP* (*Agile Information Protection Proxy, the core part of our mechanism, used for accepting the user's service request, service searching, service assigning, service-providing proxy or service connection between the provider and VU by means of pseudonym technology or between the mix zone and the VU etc*). Otherwise, the *VU* can send its signed service request directly to the *AIPP*.

B. Certificate Request

1) Creation a new connection between the AIPP and the VU

When the interception thread of *AIPP* receives a service request from a *VU*, the *AIPP* will create a new connection between them.

2) Allocation memory space

The memory space will be allocated for the connection by the *AIPP*.

3) Receiving the request

The service request will be accepted by the *AIPP*. Meanwhile, thread ID and state will be assigned. And its state will also be updated by the *AIPP* when the following step is done.

4) Creation a new interception thread for waiting new service request

A new interception thread then will be created by the *AIPP* for waiting new service request.

5) Sending the service request to SSD

And then, the service request will be send to the *SSD* (*SPKI Server & Database, for providing the discovery or/and validation of SPKI certificate*) by the *AIPP*.

C. Certificate Validation

1) Response with requested service certificate chain

After having received service request from the *AIPP*, if the service certificate chain exists, the *SSD* (*Service/Entity Discovery, for service or entity and its contexts discovery from the ubiquitous computing environment*) will provide a response including a "High" user security level and related entities and service request to the *AIPP* according to the judgment of the *SSD* when the request certificate chain passes through the *SSD*'s validation.

2) Response with identification certificate

If the service certificate chain does not pass through the *SSD's* validation, or the service certificate chain does not exists but its direct signed request pass through the *SSD's* validation to its ID, the *SSD* will provide a response including a "Middle" user security level and related service request to the *AIPP*.

3) Response without requested service certificate chain and identification certificate

Otherwise, the *SSD* will provide a response including a "Low" user security level and related service request to the *AIPP*.

D. Resource Request

The *AIPP* sends the requested services to the *SED*.

E. Context Request and Context Response

The *SED* search the context (ID, Location, Activity etc.) of

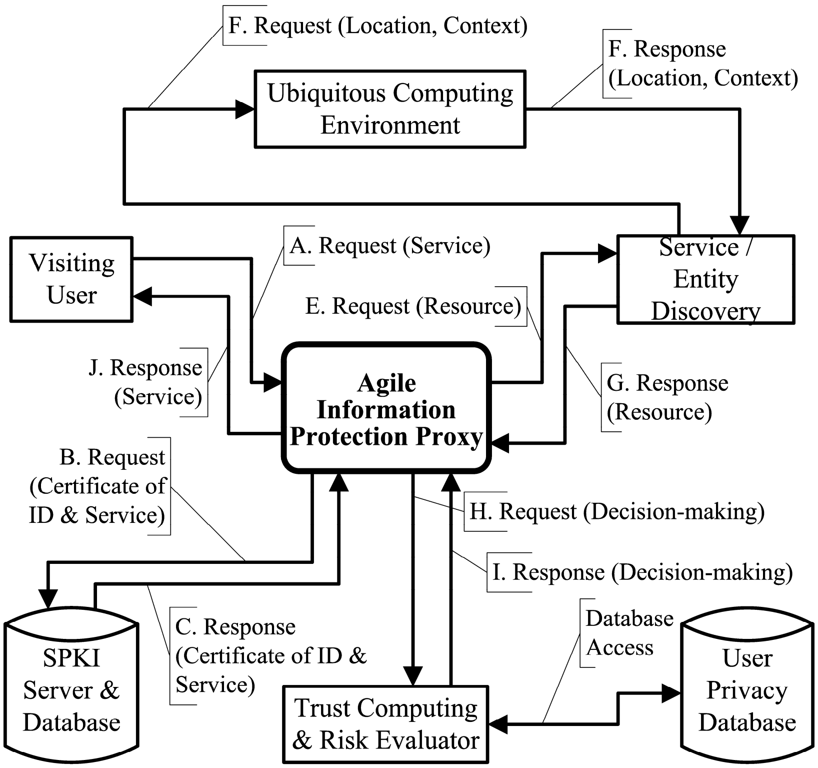

F. Request (Location, Context)
F. Response (Location, Context)
Ubiquitous Computing Environment
Visiting User
A. Request (Service)
E. Request (Resource)
Service / Entity Discovery
J. Response (Service)
Agile Information Protection Proxy
G. Response (Resource)
B. Request (Certificate of ID & Service)
H. Request (Decision-making)
I. Response (Decision-making)
C. Response (Certificate of ID & Service)
Database Access
SPKI Server & Database
Trust Computing & Risk Evaluator
User Privacy Database

Fig. 1. Mechanism of Agile Information Protection

the available entities that can provide the requested services using the entity/service discovery proxy from the ubiquitous computing environment.

F. Resource Response

The *SED* sends back the searched contexts to the *AIPP* after the previous steps have been done.

G. Decision-making Request

The *TCRE* (*Trust Computing & Risk Evaluator, which provides the trust computing and risk evaluator to the service*) computes the trust value from the entity to the *VU*. And then, by means of the help of the risk policies, the *TCRE* gives out the general trust level (Low, Middle or High). During this process, the read and save operations of the evidences, the interaction communication, the evaluator result etc are all saved into or get from the *UPD* (*User Privacy Database, used for saving evidences, trust values, interactive communication results, risk policies etc*).

H. Decision-making Response

The decision-making result will be send back to the *AIPP*.

I. Service Response

1) Total Privacy Level Computing

Firstly, the *AIPP* will compute the privacy level to decide what kinds of contexts will be protected and how to protect them. The Level computing are shown in Table I.

TABLE I
Level Computing

User Security Level	Trust Level	Protection Level	Action of AIPP
Low	Low	High	Mix Zone
Low	Middle		
Middle	Low		
Low	High	Middle	Anonymity Proxy
High	Low		
Middle	Middle		
Middle	High	Low	Pseudonym
High	Middle		
High	High		

2) Pseudonym

When the privacy level is "Low", the pseudonym technology will be used to protect the privacy of the entity that provides service to the *VU*.

The *AIPP* applies for a temporary pseudonym for the entity from the *SSD*. Then the entity can provide service for the *VU* directly by using this pseudonym. This method can be employed in the lower security case by using pseudonym to hide entity's real ID.

3) Anonymity Proxy

When the privacy level is "Middle", the anonymity proxy technology will be used to protect the privacy of the entity that provides service for the *VU*.

The *AIPP* acts as the proxy, receives the services from the entity, and then provides them for the *VU*. In this case, the entity's sensitive information, such as its ID, its physics location etc is hidden to the *VU*.

Here, the proxy can receive and explain the connection of the *VU* and build the new connection to the entity. It is the intermediate of the *VU* and the entity. That is to say, the proxy must satisfy the following conditions:

Be able to receive and explain the request from the *VU*.

Be able to create the new connection to the entity that provides service.

Be able to receive the response of entity.

Be able to send or explain the response of entity and send it back to the *VU*.

The requirement, related condition, scene etc. are translated to the proxy. The proxy gets the needed information and service resources, and then translates them to the *VU*.

4) Mix Zone

When the privacy level is "High", the mix zone technology will be used to protect the privacy of the entity that provides service to the *VU*.

The *AIPP* creates an area used for mixing the communication nodes. All entities that will provide services to the *VUs* at that time are defined as input nodes, and all *VUs* are defined as output nodes. The mix zone algorithm is random selected from the algorithm set. And then, the mix zone operation will be performed. The services will be provided between the entities and the *VUs*.

Thus, the output packets and the input packets have no relation. The attackers cannot to connect them correctly when they do not know the mixed regulation.

In the ubiquitous computing environment, some mixed areas are pre-defined. All users who enter these areas are pseudonymous; every area has a pseudonymous user set. The attackers cannot distinguish the different users. When some users enter or leave this area, the attacker can know the action of entering or leaving only.

The mix zone is developed from the pseudonymous technology. It overcomes the possible tracking to the pseudonymous user. It is suitable for some more secure application occasions.

V. Implement Examples

A. Trust Computing

According to D-S Evidence Theory, we can deduce the following two theorems [11]:

1) Trust Transfer Theorem

Under ubiquitous computing, if an environment X has not the direct trust to a principal Z , an environment Y 's recommendation is required. If the trust interval $[Bel_{XY}(\{T\}), Pl_{XY}(\{T\})]$ of X to Y is existed and the trust interval Y to Z is $[Bel_{YZ}(\{T\}), Pl_{YZ}(\{T\})]$, we know the transfer trust interval $[Bel_{XZ}(\{T\}), Pl_{XZ}(\{T\})]$.

$$Bel_{XZ}(\{T\}) = Bel_{XY}(\{T\}) \cdot Bel_{YZ}(\{T\}) \qquad (1)$$

$$\begin{aligned} Pl_{XZ}(\{T\}) = Pl_{XY}(\{T\}) + Pl_{YZ}(\{T\}) \\ - Pl_{XY}(\{T\}) \cdot Pl_{YZ}(\{T\}) \end{aligned} \qquad (2)$$

2) Trust Clustering Transfer Theorem

There are no direct trust interval between environment X and environment Y but some trust intervals $[Bel_i(\{T\}), Pl_i(\{T\})], 1 \le i \le n,$ which do not cross each other. We can compute the clustering trust interval $[Bel_{XY}(\{T\}), Pl_{XY}(\{T\})]$.

Let $m_1, m_2, \ldots, m_n$ be basic trust probability assignment function which belong to 2^U (m_i is the symbol of m_{XY_i}), their correctitude sum is $m_{XY} = m_1 \oplus m_2 \oplus \cdots \oplus m_n$, and we define

$$\begin{cases} m_{XY}(\Phi) = 0, \ \forall \ A \subseteq 2^U, A_i \subseteq 2^U, A = \Phi \\ m_{XY}(A) = K \sum\limits_{\cap A_i = A} \prod\limits_{i=1}^{n} m_i(A_i), \\ \qquad \forall \ A \subseteq 2^U, A_i \subseteq 2^U, A \neq \Phi \end{cases} \qquad (3)$$

Where,

$$K^{-1} = \sum_{\cap A_i = \Phi} \prod_{i=1}^{n} m_i(A_i)$$

When n = 2, for example, that is $m \ = m_1 \oplus m_2$. Then we can conclude

$$\begin{cases} K^{-1} = m_1(\{T\}) \cdot m_2(\{T\}) + m_1(\{T\}) \cdot m_2(\{T, D\}) \\ \quad + m_1(\{D\}) \cdot m_2(\{D\}) + m_1(\{D\}) \cdot m_2(\{T, D\}) \\ \quad + m_1(\{T, D\}) \cdot m_2(\{T\}) + m_1(\{T, D\}) \\ \quad \cdot m_2(\{D\}) + m_1(\{T, D\}) \cdot m_2(\{T, D\}) \end{cases}$$

$$\Rightarrow K^{-1} = 1 - Bel_1 - Bel_2 + Bel_1 \cdot Pl_2 + Pl_1 \cdot Bel_2 \qquad (4)$$

$$\begin{cases} Bel \ (\{T\}) = K \cdot (m_1(\{T\}) \cdot m_2(\{T\}) + m_1(\{T\}) \\ \qquad \cdot m_2(\{T, D\}) + m_1(\{T, D\}) \cdot m_2(\{T\})) \end{cases}$$

$$\Rightarrow \begin{cases} Bel \ (\{T\}) = K \cdot (Bel_1(\{T\}) \cdot Pl_2(\{T\}) \\ \qquad + Pl_1(\{T\}) \cdot Bel_2(\{T\}) \\ \qquad - Bel_1(\{T\}) \cdot Bel_2(\{T\})) \end{cases} \qquad (5)$$

$$\begin{cases} Pl\ \{\{T\}\} = K \cdot (m_1(\{T\}) \cdot m_2(\{T\}) + m_1(\{T\}) \\ \qquad \cdot m_2(\{T, D\}) + m_1(\{T, D\}) \cdot m_2(\{T\}) \\ \qquad + m_1(\{T, D\}) \cdot m_2(\{T, D\})) \end{cases}$$

$$\Rightarrow \qquad Pl\ (\{T\}) = K \cdot Pl_1(\{T\}) \cdot Pl_2(\{T\}) \qquad (6)$$

B. *Mix Zone*

Algorithm description: The shuffle regulation needs to be changed in time. The current regulation comes from the random algorithm. The input nodes are numbered by their ingoing order, and the output nodes are numbered by the shuffle regulation.

The sample code of shuffle regulation is shown Fig. 2.

VI. Conclusion And Future Work

This paper has presented a novel agile information protection mechanism based on D-S Evidence Theory and SPKI/SDSI. It can fit the developing security need in the ubiquitous computing environment.

Next, we will combine this work with security service and security architecture. We hope it will bring the ubiquitous computing into our lives more safely in the near future.

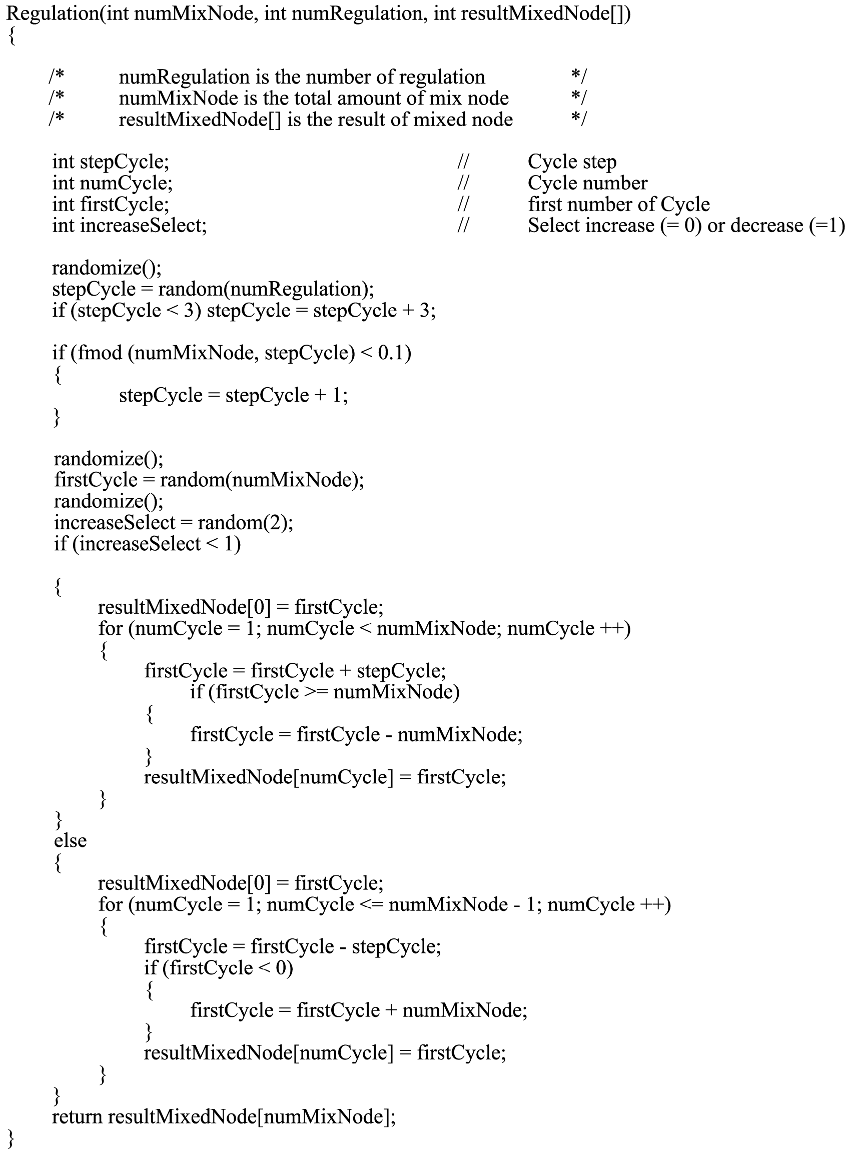
References

[1] M. Weiser, The computer of the 21st century, *Scientific American*, Vol. 265, 1991, pp. 66-75.

[2] R Rivest ,et al. (1996) SDSI :a simple distributed security infrastructure, [Online] Available: http://theory.lcs.mit.edu/~rivest/publications.html.

[3] C. Ellison, SPKI requirements, *RFC 2692,* 1999.

[4] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, SPKI certificate theory, *RFC 2693,* 1999.



```
Regulation(int numMixNode, int numRegulation, int resultMixedNode[])
{

    /*      numRegulation is the number of regulation          */
    /*      numMixNode is the total amount of mix node         */
    /*      resultMixedNode[] is the result of mixed node      */

    int stepCycle;                          //      Cycle step
    int numCycle;                           //      Cycle number
    int firstCycle;                         //      first number of Cycle
    int increaseSelect;                     //      Select increase (= 0) or decrease (=1)

    randomize();
    stepCycle = random(numRegulation);
    if (stepCycle < 3) stepCycle = stepCycle + 3;

    if (fmod (numMixNode, stepCycle) < 0.1)
    {
            stepCycle = stepCycle + 1;
    }

    randomize();
    firstCycle = random(numMixNode);
    randomize();
    increaseSelect = random(2);
    if (increaseSelect < 1)

    {
        resultMixedNode[0] = firstCycle;
        for (numCycle = 1; numCycle < numMixNode; numCycle ++)
        {
            firstCycle = firstCycle + stepCycle;
                if (firstCycle >= numMixNode)
            {
                firstCycle = firstCycle - numMixNode;
            }
            resultMixedNode[numCycle] = firstCycle;
        }
    }
    else
    {
        resultMixedNode[0] = firstCycle;
        for (numCycle = 1; numCycle <= numMixNode - 1; numCycle ++)
        {
            firstCycle = firstCycle - stepCycle;
            if (firstCycle < 0)
            {
                firstCycle = firstCycle + numMixNode;
            }
            resultMixedNode[numCycle] = firstCycle;
        }
    }
    return resultMixedNode[numMixNode];
}
```

Fig. 2. Example of shuffle regulation program implement

[5] ITU-T, Recommendation X.509 (1997 E): Information technology – open systems interconnection – the directory, *ISO/IEC*, 1997, pp. 9594-9598.

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

[7] K. Herrmann, M.A. Jaeger, PAYFLUX - secure electronic payment in mobile ad hoc networks, *Information and Communications Security. In Proc. 2004 6th International Conference, ICICS 2004. Proceedings (Lecture Notes in Computer Science* vol. 3269*)*, 2004, pp. 66-78.

[8] A. P. Dempster, Upper and lower probability induced by a multivalued mapping, *Annals Mathematical Statistics*, Vol. 2, 1967, pp. 325-339.

[9] G. A Shafer, *Mathematical theory of evidence*, Princeton University Press, Princeton, 1976.

[10] V. Cahill, et al. Using trust for secure collaboration in uncertain environments, *IEEE Pervasive Computing*, Vol. 3, 2003, pp. 52-61.

[11] D. Sun, H. Cai, Q. Cao, F. Pu, and R. Huang, Ubiquitous Computing Trust Mechanism Based On D-S Evidence Theory, *Dynamics of Continuous, Discrete and Impulsive System, Series B*, Vol. 13E, No. 3, 2006, pp. 1240-1245.

[12] M. Dam, Regular SPKI, *Lecture Notes in Computer Science*, Vol. 3364, 2005, pp. 134-152.

[13] P. G. Argyroudis, D. O'Mahony, Securing communications in the smart home, *Embedded and Ubiquitous Computing, Lecture Notes in Computer Science*, Vol. 3207, 2004, pp. 891-902.

[14] 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.

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