UCSMdess: Ubiquitous Computing Service Model based on D-S Evidence Theory and Extended SPKI/SDSI

Daoqing Sun, Yishu Luo, and Qiying Cao

***Abstract*—**Ubiquitous computing systems typically have lots of security problems in the area of service supply. The service sorts and levels, the security delegation of services, the simple privacy protection of principal and the trust computing etc are all these unsolved problems. In this paper, UCSMdess, a new novel ubiquitous computing service model based on D-S Evidence Theory and extended SPKI/SDSI is presented. D-S Evidence Theory is used in UCSMdess to compute the trust value from the ubiquitous computing environment to the principal or between the different ubiquitous computing environments. SPKI-based authorization is expanded by adding the trust certificate in UCSMdess to solve above problems in the ubiquitous computing environments. The service model with the algorithm of certificate reduction is then given in the paper.

***Keywords*—**Evidence Theory, Security, Service Model, 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 identification and trust value of the principal with the related service classification and security level, the security communication among the service supply process, the security delegation authorization of the service etc. How to solve them will be a key security problem in ubiquitous computing environment.

D-S Evidence Theory [2, 3] is a suitable method in solving the computing problems of uncertainty information in ubiquitous computing environments. It is used in our model to compute the trust value from the ubiquitous computing environment to the principal or between the different ubiquitous computing environments [4].

In order to provide security service, SPKI-based authorization technology is used in our model [5]. It can solve lots of the problems such as the identification validation of the principal who wants to enjoy the ubiquitous computing system's services while binding these services to the principal in the ubiquitous computing environment. It can also solve lots of unsolved security problems. These problems are the disconnected connection network, the classification requirements of services, the difference between different environments, security delegation, service sorts, service levels, group authorization, delegating authorization and simple privacy protection.

After adding a trust certificate, the SPKI technology is expanded. It can help us integrate the trust computing into the service supply process.

Therefore, a new ubiquitous computing service model that is based on D-S Evidence Theory and extended SPKI/SDSI, named UCSMdess, is presented to solve them in this paper. It is our main innovation.

The paper is organized as follows. Introduction of D-S Evidence Theory and SPKI/SDSI are given in Section II and Section III respectively. The extended SPKI/SDSI is presented in Section IV. Then, the service authorization is described in Section V. Afterwards, the service model and algorithm of certificate reduction are presented in Section VI before a conclusion of the paper is given in section VII.

II. D-S EVIDENCE THEORY

According to D-S Evidence Theory, we can deduce the following two theorems [4]:

A. *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 $[Bel_{XZ}(\{T\}), Pl_{XZ}(\{T\})]$.

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

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

Manuscript received May 29, 2007; Revised version received September 16, 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).

Yishu Luo is with the School of Computer Sciences and Technology, Donghua University, Shanghai 201620 China (e-mail: lys@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).

B. Trust Clustering 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

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

Where,

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

III. 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. [6], has been proposed as a standard in the RFCs 2692 [7] and 2693 [8]. It provides two main features: a set of tools (for describing and delegating authorizations and an infrastructure) and two kinds of certificates (the name certificate and the authorization certificate). SPKI-based authorization is an ideal method for decentralized ubiquitous computing environments [5].

IV. Extended SPKI/SDSI

The SPKI/SDSI authorization certificate can be regarded as the authorization from one principal to another principal, meanwhile the trust can also be regarded as the authorization from one principal to another principal, so the SPKI/SDSI is expanded by adding a SPKI/SDSI trust certificate. The SPKI/SDSI trust certificate is shown in Table I.

TABLE I
SPKI/SDSI Trust Certificate

Issuer	Public key of the certificate issuer, whose signature should follow the certificate.
Subject	Public key or name composed of a public key followed by one or more identifiers.
Trust interval	Specification of the trust value that will be granted by the issuer to the subject that has a (Bel, Pl) format.
Classification	Specification of the trust classification that has a (bigClass, smallClass) format.
Original bit	Binary field that indicates whether the trust interval is original or not.
Validity	Period during which the certificate is considered valid that has a (not-before, not-after) format.

The classification item includes two parts, big class name and small class name. When both of them are not null, the trust interval of this certificate comes from the original evidences.

When the small class is null, this trust interval is a total trust in the big class and comes from the trust computing (not from original evidences).

When both of the small class and the big class are null, this trust interval is a total trust to the whole principal and comes from the trust computing (not from original evidences).

The certificate coming from the trust computing needs to be updated periodically.

The classification of trust, the example of service resource and the example of service environment are shown from Table II to Table IV respectively.

TABLE II
Classification of Trust

Classification	Original	Trust interval
(bigClass,smallClass)	Yes	Original evidence
(bigClass,NULL)	No	Trust computing *
(NULL,NULL)	No	Trust computing *

* Need to be updated periodically.

TABLE III
Example of Service Resource

Sort	Sub-Sort	Name
Identity	Bio-identity	fingerprints
	Financial	credit card numbers
	Legal	government ID numbers
	Social	ethnicity
	Relationships	parent of
	Real Property Associations	home address
	Digital ID	username
Behavior	Financial	monthly variance against baseline
		matched with experience
	Social	drug use
		violations of law
Taste	Life	buying patterns
		eating favor
	Work	research favor
Context	Historical	mobile phone records
	Real-Time	current location

TABLE IV
Example of Service Environment

General Sort	Special Sort
Home	Belonging Finding
	Household Objects Controlling
	Personalized Access
Work	Mobility Workers Support
	Efficient Tools Providing
	Domain-specific Functionalities
Health	Prevention
	Cure (short-term)
	Care (long-term)
	Optimizing of the Alarm Chain
Shopping	Personal Shopping Management
	Intelligent Combined
Mobility	Safety Need Service
	Fast Payment
	Help in Emergencies
Leisure & Entertainment	Context Awareness Game
	Self-customization Entertainment
	Cross-media Access and Retrieval

V. Service Authorization

A. *Definitions*

Firstly, definitions are given as below:

1) S_A: subject, which can enter new ubiquitous computing environment freely and need services from the new environment.

2) I_B: issuer, ubiquitous computing environment, which can provide services to the ingoing entity of S_A.

3) I_C: issuer, the subject's environment, which includes all detailed information of S_A and provides the service authorization to its subjects.

4) I_{Dj}: issuer, the third parties, the ubiquitous computing environments, which can provide recommendation TI_{4j} to I_B about I_C.

5) CS: certificate server, for providing the certificate conservation, the search of certificate chain and online validation etc.

6) CS_B: certificate server, used for supporting service supply of I_B.

7) CS_C: certificate server, used for supporting certificate authorization of I_C.

8) CD_A: mini certificate database of the mobile user or the ingoing entity.

9) E1: evidences of sort no. 1, which have the same big class and the same small class of evidences.

10) E2: evidences of sort no. 2, which have the same big class and the different small class of evidences.

11) E3: evidences of sort no. 3, which have the different big class of evidences, and are ignored in our systems.

12) TI: trust interval.

13) TI_1: $[Bel_1(\{T\}), Pl_1(\{T\})]$, trust interval from I_B to S_A through *E1*.

14) TI_{1i}: $[Bel_{1i}(\{T\}), Pl_{1i}(\{T\})]$, trust interval from I_B to S_A through *E2*, $i = 1, 2, \ldots, n$.

15) TI2: $[Bel_2(\{T\}), Pl_2(\{T\})]$, trust interval from I_C to S_A through *E1*.

16) TI2i: $[Bel_{2i}(\{T\}), Pl_{2i}(\{T\})]$, trust interval from I_C to S_A through *E2*, $i = 1, 2, \ldots, n$.

17) TI3: $[Bel_3(\{T\}), Pl_3(\{T\})]$, total trust interval from I_B to I_C.

18) TI4j: $[Bel_{4j}(\{T\}), Pl_{4j}(\{T\})]$, total trust interval from I_{Dj} to I_C, $j = 1, 2, \ldots, m$.

19) TI5j: $[Bel_{5j}(\{T\}), Pl_{5j}(\{T\})]$, total trust interval from I_B to I_{Dj}, $j = 1, 2, \ldots, m$.

B. *Service Authorization Process*

Every principal (here, the principal indicates I_C, mobile user, ingoing entity or various authorization agents) owns at least one asymmetric key pair whose public key identifies the principal globally. The I_C 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 the entity too. Every new service authorization certificate should be sent to the CS_C and saved in the CS_C. The principal can also save its service authorization certificate into its mini CD_A. This is an optional operation to the principal according to its own need.

VI. Service Model and Algorithm of Certificate Reduction

A. *Service Model*

The service model with the trust certificate reduction is shown in Fig. 1. The details are discussed as follows.

B. *Task 1*

The principal searches for its CD_A to find 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 as its service request to the I_B. If the suitable service certificate chain does not exist, the principal will send its signed service request to the I_B directly.

C. *Task 2*

After having received service request, if the certificate chain exists, the I_B will provide the services to the principal according to the judgment of the IB when the request and trust certificate pass through the I_B*'s* validation. The I_B will refuse the request when any of them does not pass through the I_B*'s* validation.

If the certificate chain does not exist, the I_B will try to find it from the CS_B or the I_B at first. And then I_B will provide the services to the principal according to the judgments of the IB when the new service certificate chain and the trust chain exist and pass through the I_B*'s* validation. Otherwise, the I_B will refuse it.

D. *Algorithm 1*

$$\begin{cases} \mathrm{I_B, S_A, TI_{1i}(Bel_{1i}, Pl_{1i}), C_{E2}(b, s_i),} \\ \qquad \mathrm{O_{1i}(1|0), V_{1i}(V_{b1i}, V_{a1i})} \end{cases}$$

$$\Rightarrow \begin{cases} \mathrm{I_B, S_A, TI_1(Bel_1, Pl_1), C_{E2}(b, NULL),} \\ \qquad \mathrm{O_1(0), V_1(V_{b1}, V_{a1})} \end{cases} \qquad (4)$$

Where, using "(3)", we can calculate the trust interval.

Take "n = 2" as an example, that is $m_1 = m_{11} \oplus m_{12}$. Then we can conclude

$$\begin{aligned} K^{-1} &= m_{11}(\{T\}) \cdot m_{12}(\{T\}) + m_{11}(\{T\}) \\ &\cdot m_{12}(\{T, D\}) + m_{11}(\{D\}) \cdot m_{12}(\{D\}) \\ &+ m_{11}(\{D\}) \cdot m_{12}(\{T, D\}) + m_{11}(\{T, D\}) \\ &\cdot m_{12}(\{T\}) + m_{11}(\{T, D\}) \cdot m_{12}(\{D\}) \\ &+ m_{11}(\{T, D\}) \cdot m_{12}(\{T, D\}) \end{aligned}$$

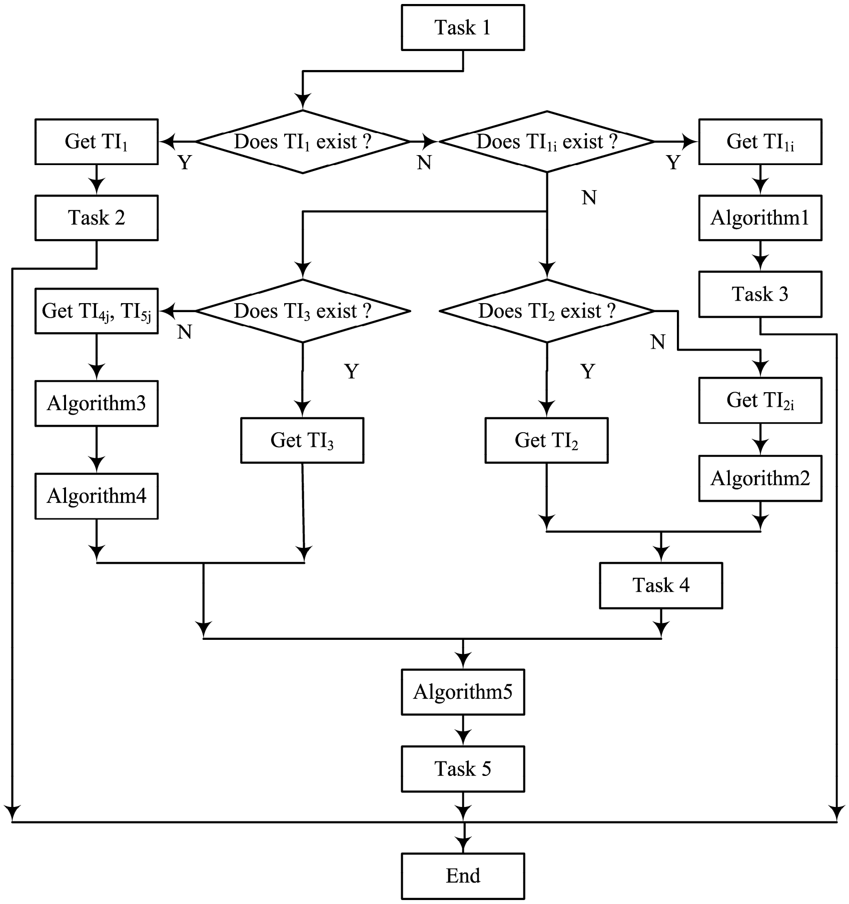

Fig. 1. Service Model with Trust Certificate Reduction

$$\Rightarrow K^{-1} = 1 - Bel_{11} - Bel_{12} + Bel_{11}\cdot Pl_{12} + Pl_{11}\cdot Bel_{12} \qquad (5)$$

$$\begin{cases} Bel_1(\{T\}) = K\cdot(m_{11}(\{T\})\cdot m_{12}(\{T\}) + m_{11}(\{T\}) \\ \qquad \cdot m_{12}(\{T,D\}) + m_{11}(\{T,D\})\cdot m_{12}(\{T\})) \end{cases}$$

$$\Rightarrow \begin{cases} Bel_1(\{T\}) = K\cdot(Bel_{11}(\{T\})\cdot Pl_{12}(\{T\}) \\ \qquad + Pl_{11}(\{T\})\cdot Bel_{12}(\{T\}) \\ \qquad - Bel_{11}(\{T\})\cdot Bel_{12}(\{T\})) \end{cases} \qquad (6)$$

$$\begin{cases} Pl_1\{\{T\}\} = K\cdot(m_{11}(\{T\})\cdot m_{12}(\{T\}) + m_{11}(\{T\}) \\ \qquad \cdot m_{12}(\{T,D\}) + m_{11}(\{T,D\})\cdot m_{12}(\{T\}) \\ \qquad + m_{11}(\{T,D\})\cdot m_{12}(\{T,D\})) \end{cases}$$

$$\Rightarrow Pl_1(\{T\}) = K\cdot Pl_{11}(\{T\})\cdot Pl_{12}(\{T\}) \qquad (7)$$

And,

$$\begin{cases} V_{b1} = \min(V_{b1}(i)), and \quad V_{a1} = \max(V_{a1}(i)) \\ \qquad where, \quad i = 1,2,\ldots,n \end{cases} \qquad (8)$$

E. Task 3

After having received service request, if the certificate chain exists, the I_B will provide the services to the principal according to the judgment of the I_B when the request and trust certificate (come from the algorithm 1) pass through the I_B's validation. The I_B will refuse the request when any of them does not pass through the I_B's validation.

If the certificate chain does not exist, the I_B will try to find it from the CS_B or the I_B at first. And then I_B will provide the services to the principal according to the judgments of the I_B when the new service certificate chain and the trust chain exist and pass through the I_B's validation. Otherwise, the I_B will refuse it.

F. Algorithm 2

$$\begin{cases} \mathrm{I_C, S_A, TI_{2i}(Bel_{2i}, Pl_{2i}), C_{E2}(b, s_i),} \\ \qquad \mathrm{O_{2i}(1|0), V_{2i}(V_{b2i}, V_{a2i})} \end{cases}$$

$$\Rightarrow \begin{cases} \mathrm{I_C, S_A, TI_2(Bel_2, Pl_2), C_{E2}(b, NULL),} \\ \qquad \mathrm{O_2(0), V_2(V_{b2}, V_{a2})} \end{cases} \qquad (9)$$

Where, using "(3)", we can calculate the trust interval.

Take "n = 2" as an example, that is $m_2 = m_{21} \oplus m_{22}$. from the conclusion of algorithm 1, we know

$$K^{-1} = 1 - Bel_{21} - Bel_{22} + Bel_{21} \cdot Pl_{22} + Pl_{21} \cdot Bel_{22} \quad (10)$$

$$\begin{cases} Bel_2(\{T\}) = K \cdot (Bel_{21}(\{T\}) \cdot Pl_{22}(\{T\}) \\ \qquad + Pl_{21}(\{T\}) \cdot Bel_{22}(\{T\}) \\ \qquad - Bel_{21}(\{T\}) \cdot Bel_{22}(\{T\})) \end{cases} \quad (11)$$

$$Pl_2(\{T\}) = K \cdot Pl_{21}(\{T\}) \cdot Pl_{22}(\{T\}) \quad (12)$$

And,

$$\begin{cases} V_{b2} = \min(V_{b2}(i)), \quad and \quad V_{a2} = \max(V_{a2}(i)) \\ \qquad where, \quad i = 1,2,\ldots,n \end{cases} \quad (13)$$

G. *Task 4*

The I_B will try to find and validate the service certificate chain from the I_C to the principal with the help of the I_C and CS_C or from the I_B. If the service certificate chain does not pass through the validation or it cannot be created, the I_B will refuse this service request and cancel this service process.

H. *Algorithm 3*

$$\left\{ \begin{array}{l} \begin{cases} \mathrm{I_{Dj}, I_C, TI_{4j}(Bel_{4j}, Pl_{4j}), C_E(NULL, NULL),} \\ \qquad \mathrm{O_{4j}(0), V_{4j}(V_{b4j}, V_{a4j})} \end{cases} \\ and \\ \begin{cases} \mathrm{I_B, I_{Dj}, TI_{5j}(Bel_{5j}, Pl_{5j}), C_E(NULL, NULL),} \\ \qquad \mathrm{O_{5j}(0), V_{5j}(V_{b5j}, V_{a5j})} \end{cases} \end{array} \right.$$

$$\Rightarrow \begin{cases} \mathrm{I_B, I_C, TI_{3j}(Bel_{3j}, Pl_{3j}), C_E(NULL, NULL),} \\ \qquad \mathrm{O_{3j}(0), V_{3j}(V_{b3j}, V_{a3j})} \end{cases} \quad (14)$$

Where,

$$\begin{cases} Bel_{3j}(\{T\}) = Bel_{4j}(\{T\}) \cdot Bel_{5j}(\{T\}) \\ \qquad where, j = 1,2,\ldots,m \end{cases} \quad (15)$$

$$\begin{cases} Pl_{3j}(\{T\}) = Pl_{4j}(\{T\}) + Pl_{5j}(\{T\}) \\ \qquad - Pl_{4j}(\{T\}) \cdot Pl_{5j}(\{T\}) \\ \qquad where, j = 1,2,\ldots,m \end{cases} \quad (16)$$

$$\begin{cases} V_{b3}(j) = \max(V_{b4}(j), V_{b5}(j)), \\ \qquad and \\ V_{a3}(j) = \min(V_{a4}(j), V_{a5}(j)) \\ \qquad where, \quad j = 1,2,\ldots,m \end{cases} \quad (17)$$

I. *Algorithm 4*

$$\begin{cases} \mathrm{I_B, I_C, TI_{3j}(Bel_{3j}, Pl_{3j}), C_E(NULL, NULL),} \\ \qquad \mathrm{O_{3j}(0), V_{3j}(V_{b3j}, V_{a3j})} \end{cases}$$

$$\Rightarrow \begin{cases} \mathrm{I_B, I_C, TI_3(Bel_3, Pl_3), C_E(NULL, NULL),} \\ \qquad \mathrm{O_3(0), V_3(V_{b3}, V_{a3})} \end{cases} \quad (18)$$

Where, using "(3)", we can calculate the trust interval.

Take "m = 2" as an example, that is $m_3 = m_{31} \oplus m_{32}$. from the conclusion of algorithm 1, we know

$$K^{-1} = 1 - Bel_{31} - Bel_{32} + Bel_{31} \cdot Pl_{32} + Pl_{31} \cdot Bel_{32} \quad (19)$$

$$\begin{cases} Bel_3(\{T\}) = K \cdot (Bel_{31}(\{T\}) \cdot Pl_{32}(\{T\}) \\ \qquad + Pl_{31}(\{T\}) \cdot Bel_{32}(\{T\}) \\ \qquad - Bel_{31}(\{T\}) \cdot Bel_{32}(\{T\})) \end{cases} \quad (20)$$

$$Pl_3(\{T\}) = K \cdot Pl_{31}(\{T\}) \cdot Pl_{32}(\{T\}) \quad (21)$$

And,

$$\begin{cases} V_{b3} = \min(V_{b3}(j)), \quad and \quad V_{a3} = \max(V_{a3}(j)) \\ \qquad where, \quad j = 1,2,\ldots,m \end{cases} \quad (22)$$

J. *Algorithm 5*

$$\left\{ \begin{array}{l} \begin{cases} \mathrm{I_C, S_A, TI_2(Bel_2, Pl_2), C_{E1}(b, s),} \\ \qquad \mathrm{O_2(1|0), V_2(V_{b2}, V_{a2})} \end{cases} \\ and \\ \begin{cases} \mathrm{I_B, I_C, TI_3(Bel_3, Pl_3), C_E(NULL, NULL),} \\ \qquad \mathrm{O_3(0), V_3(V_{b3}, V_{a3})} \end{cases} \end{array} \right.$$

$$\Rightarrow \begin{cases} \mathrm{I_B, S_A, TI_1(Bel_1, Pl_1), C_{E1}(b, s),} \\ \qquad \mathrm{O_1(0), V_1(V_{b1}, V_{a1})} \end{cases} \quad (23)$$

Where,

$$Bel_1(\{T\}) = Bel_2(\{T\}) \cdot Bel_3(\{T\}) \quad (24)$$

$$\begin{cases} Pl_1(\{T\}) = Pl_2(\{T\}) + Pl_3(\{T\}) \\ \qquad - Pl_2(\{T\}) \cdot Pl_3(\{T\}) \end{cases} \quad (25)$$

$$V_{b1} = \max(V_{b2}, V_{b3}), \quad and \quad V_{a1} = \min(V_{a2}, V_{a3}) \quad (26)$$

K. *Task 5*

If the service certificate chain from the I_B to the principal exists, the I_B will provide the services to the principal according to the judgment of the I_B when the request and trust certificate (come from the algorithm 5) pass through the I_B*'s* validation. The I_B will refuse the request when any of them does not pass through the I_B 's validation.

If the service certificate chain from the I_B to the principal does not exist, the I_B will try to find the service certificate from the I_B to the I_C, and combine it with the certificate from I_C to the principal that comes from the Task 4. And then I_B will provide the services to the principal according to the judgments of the I_B when the new service certificate chain and the trust certificate (come from the algorithm 5) exist and pass through the I_B*'s* validation. Otherwise, the I_B will refuse it.

VII. Conclusion and Future Work

This paper has presented a novel ubiquitous computing service model based on D-S Evidence Theory and extended SPKI/SDSI, called UCSMdess. We can benefit from UCSMdess, which provides a secure and feasible mechanism for solving trust service supply problems to the mobile user or the ingoing entity in ubiquitous computing environments.

Next, we will combine this work with identification, trust and security architecture. We hope it will bring us some benefits in pushing the ubiquitous computing into our life in the near future.

References

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

[2] A. P. Dempster, Upper and Lower Probability Induced by a Multivalued Mapping, *Annals Mathematical Statistics,* Vol. 38, No. 2, 1967, pp. 325-339.

[3] G. A Shafer, *Mathematical Theory of Evidence*, Princeton: Princeton University Press, 1976.

[4] 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.

[5] 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.

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

[7] C. Ellison, SPKI Requirements, *RFC 2692,* 1999.

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

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