ID-Based Access Control and Authority Delegations

So-Young Park and Sang-Ho Lee

Dept. of Computer Science and Engineering, Ewha Womans University,
11-1 Daehyun-Dong, Seodaemoon-Gu, Seoul, Korea
`soyoung@ewhain.net`, `shlee@ewha.ac.kr`

Abstract. In ubiquitous computing environments, a user can utilize a certain service in anywhere and anytime using any devices, then authentication and authorization are crucial for security in pervasive computing. In this paper, we propose a new ID-based access control model adequate for ubiquitous computing. We eliminate the use of X.509 certificates of trusted parties for authentication and allow ID-based authority delegations for flexible access control in distributed networks. We implement our proposed model using a bilinear map such as Weil pairing, and we prove the security of the proposed model against chosen-message attack under the random oracle model.

1 Introduction

1.1 Background and Related Work

With our computing environments becoming ubiquitous, it is able to be realized to utilize any services what a user wants in anywhere and anytime using any devices. In order to be possible seamless and secure communications under distributed and heterogeneous network environments, in addition to confidentiality, integrity and availability that are usually primary goals of any security systems, authentication and authorization are crucial for security in pervasive computing. Due to the distributed nature of networks, we cannot assume an architecture with a central authority for authentication such like Kerberos[15]. Distributed trust can be a solution to this problem. However, the well-known existing security infrastructure[7][8][12][14][16][2] that deal with authentication and authorization in distributed networks are based on X.509[17] certificates in public key infrastructure (PKI). Certificate-based authentication require very high computational costs and bandwidth, moreover they cause other certificate-related management problems including revocation, storage and distribution[10]. Kagal et al.[11] proposed a distributed trust model for a secure Smart Office not with certificates but with XML signatures from a trusted authority.

We suggest a flexible access control model adequate for pervasive computing. Let us consider the following ubiquitous service to illustrate our potential application. There is a service provider that provides multimedia services, and a client Alice registered to the service provider with her identity. Alice has several

types of devices such as a personal computer, a cellular phone, a PDA or a laptop, etc., and wants to enjoy a certain service on different devices continuously regardless to her location. However, Alice does not want to register all her devices to the service provider. Then, how can the service provider authenticate the authorization of each device? We provide a lightweight ID-based access control model. Each device has its own identity such as an IP address or a phone number, etc. The device can create a token for authentication using its identity with the help of the device holder and send it to the service provider. Then, the service provider can authenticate the device by verifying the token using Alice's identity. In addition, Alice wants to enjoy the same multimedia services on TV which is in her friend's house with her friend John, or on some other foreign devices. But Alice does not want her personal or secure information to be written or memorized in the foreign devices. Or, John wants to enjoy a certain multimedia service of the service provider on his TV for a short period of time under the agreement of Alice, but he does not want to register to the service provider. In this case, Alice can delegate her access authority to John's device temporarily. Alice creates a delegation token on the identity of John's device using her own device (for example, a cellular phone) and sends it to John's device. Then, John tries to access the service with the delegation token, and the service provider can decide whether John's device is authorized or not, using the identity of the device and the delegation token.

1.2 Our Contribution

In this paper, we introduce a new ID-based access control model appropriate to ubiquitous computing. We eliminate the use of X.509 certificates, and allow ID-based authority delegations.

Decentralized-Certificateless Authentication: We do not assume a centralized authentication server such as Kerberos or the uses of certificates of a trusted authority for authentication. We assume that the entire network is decentralized and collections of several different types of sub-networks. We remove the need for certificates and some of the problems associated with them by applying ID-based cryptography[13][3][4][6] to our scheme. Using the Gap Diffie-Hellman property of the pairing, we implement a lightweight ID-based access control model without certificates.

ID-Based Authority Delegation: We provide ID-based authority delegations. A delegation is a temporary permit issued by the user and given to another subject that authorizes the subject to act on the user's behalf[5]. We allow authority delegations to be carried out between two subjects in different network domains using the identities of the subjects without any exposure of private or secure information of the subjects, and without any helps (authorization) of a trusted party. We assume a temporary delegation like one-time permission. The delegated subject can use a certain service for a short period of time, and once the delegation expires, the delegated subject is denied access to any services.

Our ID-based access control model is built from a modified bilinear map such as the weil pairing on elliptic curves[3]. We define a chosen-message security for our ID-based access model and analyze the security of the proposed scheme against the chosen-message attack under the random oracle model.

The rest of the paper is organized as follows. In Section 2, we review the notion of a bilinear map. We formalize our model and explain concrete descriptions of ID-based access control scheme in Section 3. In Section 4, we analyze the security of the proposed scheme, and then conclude the paper in Section 5.

2 Bilinear Map

We briefly review the bilinear map and several hard problems which provide the basic security of our scheme. Boneh and Franklin introduced a bilinear map called a "pairing" in their IBE scheme. Typically, the pairing used is a modified Weil or Tate pairing on a supersingular curve or abelian variety[3][1]. Let G_1 be an additive group of prime order q and G_2 be a multiplicative group of the same order. Our scheme makes use of an admissible bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ between these two groups. The map must satisfy the following properties:

1. Bilinear: We say that a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is bilinear, for all $Q, W, Z \in G_1$, if $\hat{e}(Q, W + Z) = \hat{e}(Q, W)\hat{e}(Q, Z)$ and $\hat{e}(Q + W, Z) = \hat{e}(Q, Z)\hat{e}(W, Z)$. Consequently, for any $a, b \in Z_q^*$, $\hat{e}(aQ, bW) = \hat{e}(Q, W)^{ab} = \hat{e}(abQ, W)$.
2. Non-degenerate: The map does not send all pairs in $G_1 \times G_1$ to the identity in G_2 . Observe that since G_1, G_2 are prime order groups, it is implied that if P is a generator of G_1 then $\hat{e}(P, P)$ is a generator of G_2 .
3. Computable: There is an efficient algorithm to compute $\hat{e}(Q, W)$ for all $Q, W \in G_1$.

In general, G_1 is a cyclic subgroup of the additive group of points on a supersingular elliptic curve over $E(F_p)$. G_2 is a cyclic subgroup of the multiplicative group associated with a finite extension of F_p .

The security of the pairing-based schemes relies on the hardness of the following problems.

- Discrete Logarithm Problem (DLP): Given two group elements P and $Q = nP$ in G_1 , find n .
- Computational Diffie-Hellman Problem (CDHP): For any $a, b \in Z_q^*$, given $\langle P, aP, bP \rangle$, compute abP .
- Decisional Diffie-Hellman Problem (DDHP): For any $a, b, c \in Z_q^*$, given $\langle P, aP, bP, cP \rangle$, decide whether $c \equiv ab \pmod{q}$. DDHP in G_1 is easy. To see this, observe that given $\langle P, aP, bP, cP \rangle \in G_1$ we have

$$c = ab \pmod{q} \Leftrightarrow \hat{e}(P, cP) = \hat{e}(aP, bP)$$

- Gap Diffie-Hellman Problem (GDHP): A class of problems where DDHP is easy while CDHP is hard.
- Bilinear Diffie-Hellman Problem (BDHP): For any $a, b, c \in Z_q^*$, given $\langle P, aP, bP, cP \rangle$, compute $\hat{e}(P, P)^{abc} \in G_2$.

3 ID-Based Access Control Scheme

In this section, we present a formal definition for our ID-based access control scheme, and then we explain concrete descriptions of the scheme.

3.1 System Model and Security Requirements

Our ID-based access control model basically consists of a service provider, users (or subjects) and devices to be able to utilize certain services of the service provider. We assume that each user has n -types of devices, and that all devices have their own identities such as IP addresses or phone numbers, etc., and can communicate each other through wired or wireless networks.

Definition 1. *We define that access control is a mechanism that monitors and controls the subjects access to the service provider's resources. The term of control means to decide to assign access rights to the subjects after authenticating whether the subjects are authorized or not.*

A detailed system model for ID-based access control is described below. Let k be a security parameter, and let $\mathcal{IG}$ be a BDH parameter generator.

Definition 2. *An ID-based access model is specified by six algorithms (Gen_{Sys} , Reg , $\text{GEN}_{\text{AToken}}$, $\text{GEN}_{\text{DToken}}$, $\text{Auth}_{\text{Device}}$, $\text{Auth}_{\text{FDevice}}$) such that:*

1. Gen_{Sys} : *It takes as input the security parameter k and $\mathcal{IG}$. It returns system parameters.*
2. Reg : *It takes as input a user A 's identity ID_A , $\text{Auth-}ID_A$, user information and service-related information. Then, it adds the information to Access Control List (ACL) maintained by the service provider.*
3. $\text{GEN}_{\text{AToken}}$: *It takes as input A 's secure master identity MID_A , a device identity DID and an access-request message. It generates an authentication token for the device.*
4. $\text{GEN}_{\text{DToken}}$: *It takes as input A 's secure master identity MID_A , a device identity DID and an authority-delegation message. It generates a delegation token for the device.*
5. $\text{Auth}_{\text{Device}}$: *It takes as input an access-request message, a device identity and an authentication token. It decides to assign an access right to the device or not.*
6. $\text{Auth}_{\text{FDevice}}$: *It takes as input an access-request message, a authority-delegation message, a device identity and a delegation token. It decides to assign an access right to the device or not.*

Our goal is to provide a secure access control process and a secure delegation process on "open" insecure networks. Since we assumed that the network is insecure, an adversary can intercept and modify all data transmitted on the network, and also can retransmit a valid data transmission intercepted by the adversary maliciously or fraudulently (replay attack). In addition, we assume that the adversary can do chosen-message attacks[9]. We will formalize the chosen-message

security model of our scheme against the chosen-message attacks in Section 4. Under the above attack environments, we consider the proposed scheme as satisfying at least the following security requirements for secure access control and delegations.

- **Authorization:** An access controller of a service provider can authenticate whether each subject is authorized or not.
- **Unforgeability:** The adversary cannot forge a valid token by an authorized subject without knowing secure information of the subject.
- **Accountability:** An authorized subject that carried out a delegation cannot repudiate the fact of that the subject delegated its authority to a foreign subject.
- **Revoking:** The delegator must have the ability to cancel delegations it has issued (before the valid date of delegation is expired).
- **Replay Attack Resistance:** The access controller can detect and protect the replay attack by the adversary or any subjects.

3.2 Implementation of the ID-Based Access Control Scheme

We describe the six algorithms needed to define our ID-based access control scheme. For a simple explanation, let a service provider be SP , a user be A , the user's device be DA_i , a foreign user be B and the foreign user's device be DB_i , where $i = 1, \dots, n$. First, the SP generates its system parameters as follows:

Gen_{Sys} : The SP

1. runs $\mathcal{IG}$ on input k to generate groups G_1, G_2 of some prime order q and a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$;
2. chooses arbitrary generators $P \in G_1$, and chooses cryptographic hash functions $H_1 : \{0, 1\}^* \rightarrow G_1, H_2 : \{0, 1\}^* \rightarrow Z_q^*$.

The system parameters are $params = (G_1, G_2, \hat{e}, P, H_1, H_2)$, and those are publicly available for its clients. Now, we suppose that A wants to register to the SP . The algorithm of Reg is carried out between A and SP .

Reg :

1. The SP sends the system parameters to A .
2. A makes its master secure identity $MID_A \in \{0, 1\}^*$;
3. and sets $Auth-ID_A = H_2(MID_A) \cdot P \in G_1$;
4. and sends $\langle ID_A, Auth-ID_A \rangle$ with other user information required to registration to SP .
5. SP adds the user information to its access control list.

SP maintains an access control list (ACL) for client authentication. We suppose that ACL basically consists of the attributes of client ID , $Auth-ID$, user information, contents of services and service-related information. User information can contain some private personal information that the SP wants to know

including full-name, age, mail address and phone number, etc. The contents of services can contain services allowed to A , and the service-related information can contain constraints or policies defined by SP about the services. The important thing is that SP does not contain any secret information of its clients such as passwords. SP can authenticate its clients using the pair of $(ID, Auth-ID)$ in ACL, and can provide the corresponding services to clients.

Authority Authentication. We suppose that A wants to access to SP using her own device DA_i . Then A performs GEN_{AToken} on DA_i as follows:

GEN_{AToken} :

1. A inputs ID_A and MID_A to DA_i .
2. DA_i generates $access\text{-}req = 'access\text{-}request' || ID_A || DID_{DA_i} || Time_i$, where DID_{DA_i} is the identity of DA_i , and $Time_i$ is current time information such like GMT.
3. DA_i computes $a = \frac{H_2(MID_A)}{H_2(DID_{DA_i})} \in Z_q^*$ and $AT_A = a \cdot H_1(access\text{-}req) \in G_1$.

When SP receives an access signal from a certain device DA_i , SP carries out $Auth_{Device}$ to authenticate DA_i as follows:

$Auth_{Device}$:

1. DA_i sends $\langle access\text{-}req, AT_A \rangle$ to SP .
2. SP checks $access\text{-}req$ and gains $Auth-ID_A$ from ACL corresponding to ID_A .
3. SP computes $H_1(access\text{-}req) \in G_1$ and $H_2(DID_{DA_i}) \in Z_q^*$.
4. SP checks if $\hat{e}(AT_A, H_2(DID_{DA_i}) \cdot P) = \hat{e}(H_1(access\text{-}req), Auth-ID_A)$.
5. If the equation holds, SP assigns an access right to DA_i .

Authority Delegations. An authorized subject can delegate its authority to a foreign subject that the authorized subject trusts. We suppose A wants to delegate her authority to a foreign device DB_i that B holds. We assume that A has her own device DA_i . A and B carry out GEN_{DToken} as follows:

GEN_{DToken} :

1. B inputs ID_B to DB_i .
2. DB_i generates $auth\text{-}deleg\text{-}req = 'authority\text{-}delegation\text{-}request' || ID_B || DID_{DB_i} || Time_i$, where DID_{DB_i} is the identity of DB_i and $Time_i$ is current time information.
3. B sends $auth\text{-}deleg\text{-}req$ to DA_i .
4. A inputs ID_A and MID_A to DA_i .
5. DA_i generates $auth\text{-}deleg\text{-}msg = 'authority\text{-}delegation\text{-}message' || ID_A || DID_{DB_i} || Contents\ of\ Services || Time_i$, where 'Contents of Services' are a list of services allowed to DB_i .
6. DA_i computes $d = \frac{H_2(MID_A)}{H_2(DID_{DB_i})} \in Z_q^*$ and $DT_A = d \cdot H_1(auth\text{-}deleg\text{-}msg) \in G_1$, and sends $\langle auth\text{-}deleg\text{-}msg, DT_A \rangle$ to DB_i .

When SP receives an access signal from a certain device DB_i , SP carries out $Auth_{FDevice}$ to authenticate DB_i as follows:

Auth_{FDDevice}:

1. DB_i generates $\text{deleg-access-req} = \text{'delegate-access-request'} || DID_{DB_i} || Time_i$.
2. DB_i sends $\langle \text{deleg-access-req}, \text{auth-deleg-msg}, DT_A \rangle$ to SP .
3. SP checks deleg-access-req and auth-deleg-msg , and gains Auth-ID_A from ACL corresponding to ID_A .
4. SP computes $H_1(\text{auth-deleg-msg}) \in G_1$ and $H_2(DID_{DB_i}) \in Z_q^*$.
5. SP checks if $\hat{e}(DT_A, H_2(DID_{DB_i}) \cdot P) = \hat{e}(H_1(\text{auth-deleg-msg}), \text{Auth-ID}_A)$.
6. If the equation holds, SP assigns an access right to DB_i .

4 Security Analysis

In this section, we analyze the security of our proposed scheme about the requirements mentioned in Section 3.1.

Authorization: The service provider assigns an access right to subjects with tokens created by master identities corresponding to *Auth-IDs* pre-registered to the service provider. If the tokens are valid, the service provider can authenticate that the subjects with the tokens are authorized, because only users who know the master identities corresponding to *Auth-IDs* can create the valid tokens.

Unforgeability: We show that the tokens cannot be forged by adversaries at below. We prove the security of our scheme against existential forgery under a chosen-message attack in the random oracle model. Existential forgery means that the adversary attempt to forge the valid tokens, on messages of his choice, by an authorized subject. The token generation protocols of our scheme are similar to create a kind of short signature from Weil pairing[4]. Therefore, our security analysis follows basically the security proof in Boneh et al's scheme[4] except that our system parameters are selected in two groups G_1, G_2 where G_1 is an additive group and G_2 is a multiplicative group, with the same prime order q .

At first, we formalize a chosen-message security model. In this model, the adversary $\mathcal{A}$ is given public ID and Auth-ID . His goal is to existential forgery of a token. We give the adversary power to choose all public ID s and Auth-ID s except the challenge public ID and Auth-ID . The adversary is also given access to a token generating oracle on the challenge's master ID . His advantage $\text{AdvToken}_{\mathcal{A}}$, is defined to be his probability of success in the following game.

- **Setup:** The adversary $\mathcal{A}$ is provided with a public $ID_{\mathcal{A}}$ and $\text{Auth-ID}_{\mathcal{A}}$, generated at random.
- **Queries:** Proceeding adaptively, $\mathcal{A}$ requests tokens with $\text{Auth-ID}_{\mathcal{A}}$ on messages of his choice.
- **Response:** Finally, $\mathcal{A}$ outputs a message $M_{\mathcal{A}}$, and a token $\text{Token}_{\mathcal{A}}$ on $M_{\mathcal{A}}$.

The adversary wins if the token $\text{Token}_{\mathcal{A}}$ is a valid token on message $M_{\mathcal{A}}$ under $\text{Auth-ID}_{\mathcal{A}}$, and $\text{Token}_{\mathcal{A}}$ is non-trivial.

Definition 3. An adversary $\mathcal{A}(t, q_H, q_T, \epsilon)$ -breaks a token generating scheme in the chosen-message security model if: $\mathcal{A}$ runs in time at most t ; $\mathcal{A}$ makes at most q_H queries to the hash function and at most q_T queries to the token generating oracle; $\text{AdvToken}_{\mathcal{A}}$ is at least ϵ . A token generating scheme is (t, q_H, q_T, ϵ) -secure against existential forgery in the chosen-message model if no adversary (t, q_H, q_T, ϵ) -breaks it.

Lemma 1. Let (G_1, G_2) be a (t', ϵ') -GDH group pair of order q . Then the token generation scheme on (G_1, G_2) is (t, q_H, q_T, ϵ) -secure against existential forgery under an adaptive chosen-message attack for all t and ϵ satisfying $\epsilon \geq e(q_T + 1) \cdot \epsilon'$ and $t \leq t' - c_{G_1}(q_H + 2q_T)$, where c_{G_1} is multiplication time in G_1 for each query, and e is the base of the natural logarithm.

Proof. Suppose $\mathcal{A}$ is a forger algorithm that (t, q_H, q_T, ϵ) -breaks the token generating scheme. We show how to construct a t' -algorithm $\mathcal{B}$ that solves CDH problem with probability at least ϵ' . This will contradict the fact that G_1 is a t', ϵ' -CDH group. Let P be a generator of G_1 . Algorithm $\mathcal{B}$ is given P , $A\text{-ID}$, $Q \in G_1$ and a random $z \in Z_q^*$, where $A\text{-ID} = a \cdot P$ for a random secure $a \in Z_q^*$. Its goal is to output $(a/z) \cdot Q \in G_1$. Algorithm $\mathcal{B}$ simulates the challenger and interacts with the forger $\mathcal{A}$ as follows:

Setup: Algorithm $\mathcal{B}$ starts by giving $\mathcal{A}$ the generator P and the public $\text{Auth-ID}_1 = A\text{-ID} + r \cdot P = a \cdot P + r \cdot P$ where r is a random value in Z_q^* .

H-queries: At any time algorithm $\mathcal{A}$ can query the random oracles of H . To respond to these queries algorithm $\mathcal{B}$ maintains a list of tuples $(ID_j, z_j, M_j, w_j, b_j, c_j)$ as explained below. We refer to this list as the H -list. The list is initially empty. When $\mathcal{A}$ queries the oracle H at points of ID_i and $M_i \in \{0, 1\}^*$, algorithm $\mathcal{B}$ responds as follows:

1. If the query (ID_i, M_i) already appears on the H -list in a tuple $(ID_i, z_i, M_i, w_i, b_i, c_i)$ then $\mathcal{B}$ responds with $H(ID_i) = z_i \in Z_q^*$ and $H(M_i) = w_i \in G_1$.
2. Otherwise, $\mathcal{B}$ generates a random coin $c_i \in \{0, 1\}$ so that $\Pr[c_i = 0] = 1/(q_T + 1)$. And $\mathcal{B}$ picks random values $z_i \in Z_q^*$ and $b_i \in Z_q^*$.
 - If $c_i = 0$, $\mathcal{B}$ computes $w_i = b_i \cdot P + Q \in G_1$.
 - else if $c_i = 1$, $\mathcal{B}$ computes $w_i = b_i \cdot P \in G_1$.
3. $\mathcal{B}$ adds the tuple $(ID_i, z_i, M_i, w_i, b_i, c_i)$ to the H -list and respond to $\mathcal{A}$ by setting $H(ID_i) = z_i \in Z_q^*$ and $H(M_i) = w_i \in G_1$.

T-queries: Let (ID_i, M_i) be a token query issued by $\mathcal{A}$. $\mathcal{B}$ responds to this query as follows:

1. $\mathcal{B}$ runs the above algorithm for responding to H -queries to obtain z_i and w_i . If $c_i = 0$ then $\mathcal{B}$ reports failure and terminates.
2. Otherwise, we know $c_i = 1$ and hence $w_i = b_i \cdot P$. Define $\text{Token}_i = \frac{b_i}{z_i} \cdot A\text{-ID} + \frac{rb_i}{z_i} \cdot P \in G_1$. Observe that $\frac{a}{z_i} \cdot w_i + \frac{r}{z_i} \cdot w_i = \frac{a+r}{z_i} \cdot w_i = \frac{a+r}{z_i} \cdot H(M_i)$ and therefore Token_i is a valid token on ID_i and M_i under the public $\text{Auth-ID}_1 = a \cdot P + r \cdot P = (a + r) \cdot P$. $\mathcal{B}$ gives Token_i to $\mathcal{A}$.

Output: Eventually $\mathcal{A}$ produces a message-token pair $M_f, Token_f$ such that no T -query was issued for M_f . If there is no tuple on the H -list containing M_f then $\mathcal{B}$ issues a query itself for $H(M_f)$ to ensure that such a tuple exists. We assume $Token_f$ is a valid token on M_f under the given public information; If it is not, $\mathcal{B}$ reports failure and terminates. Next, $\mathcal{B}$ finds the tuple $(ID_f, z, M_f, w_f, b_f, c_f)$ on the H -list. If $c_f = 1$ then $\mathcal{B}$ reports failure and terminates. Otherwise, $c_f = 0$ and therefore $H(M_f) = w_f = b_f \cdot P + Q$. Hence $Token_f = \frac{a+r}{z} \cdot (b_f \cdot P + Q) = \frac{a+r}{z} \cdot b_f \cdot P + \frac{a+r}{z} \cdot Q$. Then $\mathcal{B}$ outputs the required $\frac{a}{z} \cdot Q$ as

$$\frac{a}{z} \cdot Q \leftarrow Token_f - \frac{b_f}{z} \cdot A-ID - \frac{rb_f}{z} \cdot P - \frac{r}{z} \cdot Q.$$

This completes the description of algorithm $\mathcal{B}$.

It remains to show that $\mathcal{B}$ solves the given instance CDH problem with probability at least ϵ' . The probability analysis is identical to that in Boneh et al.'s scheme[4] and therefore omitted.

Accountability: An authorized subject (delegator) delegates its authority to a foreign subject by creating a delegation token using the master identity of the delegator. Since the master identity is secret information only known to the delegator, the delegator cannot repudiate the fact of delegation. Then, the service provider can know it from the delegation token that the delegation is carried out between the two subjects.

Revoking: If delegator with ID_A wants to revoke a delegation to a certain device DC during a certain service is provided to DC , then the delegator can create a revocation token for DC . The revocation token generation protocol is identical to GEN_{AToken} except that the delegator generates a revocation-request message $revoke_req = 'revocation-request' || ID_A || DID_{DC} || Time_i$, and that $RT_A = a \cdot H_1(revoke_req)$, where $a = \frac{H_2(MID_A)}{H_2(DID_{DC})}$. If the delegator sends $\langle revoke_req, RT_A \rangle$ to the SP , then SP can stop to providing the service to the corresponding device immediately after checking the token.

Replay Attacks: Whenever generating authentication tokens or delegation tokens, an authorized subject creates a new **access-req** or **auth-deleg-msg** that includes current time information. Therefore, the service provider can protect the replay attack weakly by checking the time information. The service provider can deny the token authentication for the tokens that are arrived at later than communication or computation delay time pre-defined by the SP .

5 Conclusion

In this paper, we introduced an ID-based access control model adequate for ubiquitous computing. We eliminated the use of certificates and proposed more flexible access control model by allowing subject-oriented authority delegations using the identities of subjects. Users can enjoy certain services in anywhere

and any time using any devices by generating authentication tokens or delegation tokens. Our tokens provide device authentication and user authorization simultaneously. Therefore, service providers can easily authenticate subjects just by verifying the tokens. We implemented the token generation protocols based on pairing and proved the security of our proposed model against the chosen-message attack under the random oracle model.

References

1. Barreto, P., Kim, H., Lynn, B. and Scott, M.: Efficient Algorithms for Pairing-based Cryptosystems. *Advances in Cryptology - Crypto'02. LNCS 2442* (2002) 354-368
2. Blaze, M., Feigenbaum, J. and Lacy, J.: Decentralized Trust Management. *Proc. of the IEEE Conference on Security and Privacy*, (1996)
3. Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology - Crypto'01. LNCS 2139* (2001) 213
4. Boneh, D. Lynn, B. and Shacham, H.: Short Signature from the Weil Pairing. *Advances in Cryptology - Asiacrypt'01. LNCS 2248* (2001) 514-532
5. Campbell, R., Sturman, D. and Tock, T.: Mobile Computing, Security and Delegation. *Proc. of the International Workshop on Multi-Dimensional Mobile Communications*, (1994)
6. Cha, J.C. and Cheon, J.H.: An Identity-Based Signature from Gap Diffie-Hellman Groups. *Proc. of PKC03. LNCS 2567* (2003) 18-30
7. Ellison, C. M., Frantz, B., Lampson, B., Rivest, R., Thomas, B. M. and Ylonen, T.: Simple Public Key Certificate. Internet-Draft draft-ietf-spki-certstructure- 05.txt, Internet Engineering Task Force, (1998)
8. Gasser, M. and McDermott, E.: An Architecture for Practical Delegation in a Distributed System. *Proc. of the Symposium on Security and Privacy*. (1990) 20-30
9. Goldwasser, S., Micali, S. and Rivest, R.: A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. *SIAM J. Computing*, **17**(2) (1988) 281-308
10. Gutmann, P.: PKI: Its not Dead, Just Resting. *IEEE Computer*. vol. 35, no. 8 (2002) 41-49
11. Kagal, L., Finin, T. and Joshi, A.: Moving from Security to Distributed Trust in Ubiquitous Computing Environment. *IEEE Computer* (2001)
12. Nikander, P.: An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems. Ph.D. Thesis, Helsinki University of Technology, (1999)
13. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. *Advances in Cryptology - Crypto84. LNCS 0196* (1984) 47-53
14. Sollins, K. R.: Cascaded Authentication. *Proc. of the Symposium on Security and Privacy*, (1988) 156-163.
15. Steiner, J. G., Neuman, C. and Schiller, J. I.: Kerberos: An Authentication Service for Open Network Systems. *Proc. of the Winter USENIX Conference*, (1998) 191-202
16. Zimmermann, P. R.: The Official PGP User's Guide. MIT Press, (1995)
17. Public-Key Infrastructure (X.509), <http://www.ietf.org/html.charters/pkix-charter.html>