Electronic Preprint for Journal of Information Processing Vol.20 No.3

Recommended Paper

A Pairing-Based Anonymous Credential System with Efficient Attribute Proofs*

Amang Sudarsono[1,a)] Toru Nakanishi[2] Nobuo Funabiki[2]

Received: October 4, 2011, Accepted: March 2, 2012

Abstract: To enhance user privacy, anonymous credential systems allow the user to convince a verifier of the possession of a certificate issued by the issuing authority anonymously. The typical application is the privacy-enhancing electronic ID (eID). Although a previously proposed system achieves the constant complexity in the number of finite-set attributes of the user, it requires the use of RSA. In this paper, we propose a pairing-based anonymous credential system excluding RSA that achieves the constant complexity. The key idea of our proposal is the adoption of a pairing-based accumulator that outputs a constant-size value from a large set of input values. Using zero-knowledge proofs of pairing-based certificates and accumulators, any AND and OR relation can be proved with the constant complexity in the number of finite-set attributes. We implement the proposed system using the fast pairing library, compare the efficiency with the conventional systems, and show the practicality in a mobile eID application.

Keywords: anonymity, anonymous credentials, attributes, pairings, accumulators

1. Introduction

Electronic identification has been widely applied to access authorization to buildings, use of facilities, Web services, etc. Currently, electronic identity (eID) such as eID card is often used. The eID is issued by a trusted organization such as the government, company, or university, and is used for its service. Trusted ID is very attractive for secondary use in commercial services. The eID includes attributes of the user such as the name, the address, the gender, the occupation, and the date of birth. In commercial cases, the attribute-based authentication is desired. For example, a service provider can deny access to kids, by checking the age in the eID.

One of serious issues in the existing eID systems is user's privacy. In the systems, the eID may reveal the user's identity. The service provider can collect the use history of each user. Anonymous credential systems [10], [12], [13] are one of the solutions.

An anonymous credential system basically consists of the setup algorithm, issuing protocol and proof protocol. The participants are an issuing authority, users, and verifiers. In the setup algorithm, an issuing authority generates his/her public key ipk and secret key isk. ipk is bound to the issuing authority. In the issuing protocol, the issuing authority with isk issues a certificate $cert$ to a user, given common input ipk. Each certificate is a proof of membership, qualification, or privilege, and contains user's attributes. In the proof protocol, given the common input ipk, the user with $cert$ can prove to a verifier that the user has a certificate issued by the authority bound to ipk. This protocol is anonymous, i.e., the verifier cannot know who is the proving user. This means that only ipk is the public key and the user's public key is not used. In addition, the proving user can disclose some selected attributes without revealing any other information about the user's privacy. Furthermore, the user can prove complex relations of the attributes using AND and OR relations. AND relation is used when proving the possession of all of the multiple attributes. For example, the user can prove that he is a student, and has a valid student card, when entering the faculty building. OR relation represents the proof for possession of one of multiple attributes. For example, he can prove that he is either a staff or a teacher when using a copy machine in a laboratory. An implementation of eID on a standard java card is shown in Ref. [6].

In Ref. [13], Camenisch and Lysyanskaya firstly proposed an anonymous credential system based on RSA. Unfortunately, it suffers from a linear complexity in the number of user's attributes in proving AND and OR relations. Hence, this system is not suitable for small devices such as smart cards. In Ref. [10], Camenisch and Groß extended the scheme to solve the drawback. They classified attribute types into two categories: string attributes and finite-set attributes. The former can be represented as a string, such as name and ID number. The latter can be represented as an element from relatively small finite-set, such as gender and profession. There are much fewer string type of attributes, and thus the costs on finite-set attribute types impacts the total efficiency. In Camenisch-Groß system, by encoding a large number

1 Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya, Indonesia
2 Department of Communication Network Engineering, Okayama University, Okayama 700–8530, Japan
a) nakanisi@cne.okayama-u.ac.jp

* The preliminary versions of this paper were presented at PETS 2011 [18] and IMECS 2011 [19].

The initial version of this paper was presented at Computer Security Symposium 2010 (CSS2010) held in October 2010, which was sponsored by SIG-CSEC. This paper was recommended to be submitted to Journal of Information Processing by the program chair of CSS2010.

© 2012 Information Processing Society of Japan

of finite-set attributes into prime numbers, one value for the finite-set attributes can be embedded into the certificate. Then, the AND and OR relations are proved with the constant complexity in the number of finite-set attributes using zero-knowledge proofs of integer relations on prime numbers. However, this extended system also depends on RSA. Currently, 1,024-bit RSA modulus is obsolete, and we require 2,048 bits or more. As the modulus size increases, the data size in the authentication becomes larger, and the processing times also become larger. To shorten the data size, a pairing-based system excluding the RSA assumption is desired.

In this paper, for a pairing-based anonymous credential system using BBS+ signatures [8], we show how to prove AND and OR relations with constant complexity. It seems difficult to adopt the approach in Camenisch-Groß system. This is because their approach strictly depends on integer commitments on the strong RSA assumption to prove the integer relation over integers, as mentioned in Ref. [10]. This is why we adopt a different approach. The key idea of the construction is the adoption of a pairing-based accumulator [12]. The accumulator outputs a constant-size value from a large set of input values. We consider that the input values are assigned to attributes. Then, we utilize an extended BBS+ signatures to certify a set of attributes as the accumulator. Using zero-knowledge proofs of BBS+ signatures and accumulators, we can prove AND and OR relations with constant complexity in the number of finite-set attributes. The drawback is that the size of public key is depending on the number of attribute values. It varies from 1 MBytes to 10 MBytes for the number of attribute values 1,000 to 15,000. In the current mobile environments, the data size is sufficiently practical, since the public key is not changed after it is distributed.

To show the efficiency of our system, we implemented the system using a pairing library, and compare the processing time to a conventional pairing-based anonymous credential system with linear complexity. In addition, we compare our system to the RSA-based Camenisch-Groß system. Furthermore, we implemented a prototype system of eID application and show the processing time including the communication to confirm practicality.

Remark 1 In the RSA-based anonymous credential system with efficient complexity [10], NOT relation is also equipped. Namely, the prover can prove that a specified attribute is not in his certificate. On the other hand, our system does not have the protocol to directly prove NOT relation. However, OR relation substitutes NOT relation. In an attribute type, let $S = \{a_1, \ldots, a_k\}$ be the set of all attribute values of the attribute type. Into the user's certificate, an element from S is embedded. Here we assume that any user owns an attribute from S. In the case that the user does not own any attribute value from the attribute type, we can insert the value indicating no attribute value into S. Consider the proof that the user's attribute is not a_1. This proof can be performed by the OR proof that the user owns some attribute of $S - \{a_1\}$. For example, for proving that the user is not a student, we can prove that she has some of all other profession attribute values including the value indicating no profession.

2. Preliminaries

2.1 Bilinear Groups

Our scheme utilizes the following bilinear groups:

(1) $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p,

(2) g and h are randomly chosen generator of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively,

(3) e is an efficiently computable bilinear map: $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, namely, (1) for all $u, u' \in \mathbb{G}_1$, $v, v' \in \mathbb{G}_2$, $e(uu', v) = e(u, v)e(u', v)$ and $e(u, vv') = e(u, v)e(u, v')$, and thus for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in Z$, $e(u^a, v^b) = e(u, v)^{ab}$, and (2) $e(g, h) \neq 1$.

2.2 Assumptions

The security of our scheme is based on the q-SDH assumption [7], the q-HSDH (Hidden SDH) assumption [9], and q-TDH (Triple DH) assumption [5] for the underlying signatures, and n-DHE assumption [12] for the accumulator, where q, n are non-negative integer.

Definition 1 (q-SDH assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$\Pr[\mathcal{A}(g, h, h^a, g^a, \ldots, g^{a^q}) = (b, g^{1/(a+b)}) \wedge b \in Z_p]$$

is negligible, where $g \in_R \mathbb{G}_1$, $h \in_R \mathbb{G}_2$ and $a \in_R Z_p$.

Definition 2 (q-HSDH assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$\Pr[\mathcal{A}(g, h, \hat{h}, g^a, h^a, (h^{1/(a+b_1)}, g^{b_1}, \hat{h}^{b_1}), \ldots, (h^{1/(a+b_q)}, g^{b_q}, \hat{h}^{b_q})) = (h^{1/(a+b)}, g^b, \hat{h}^b) \wedge \forall i \in [1, q] : g^b \neq g^{b_i}]$$

is negligible, where $g \in_R \mathbb{G}_1$, $h, \hat{h} \in_R \mathbb{G}_2$, $a \in_R Z_p$, and $b, b_i \in Z_p$.

Definition 3 (q-TDH assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$\Pr[\mathcal{A}(g, h, g^a, h^a, h^b, (c_1, h^{1/(a+c_1)}), \ldots, (c_q, h^{1/(a+c_q)})) = (g^{ra}, h^{rb}, h^{rab}) \wedge \forall i \in [1, q] : c \neq c_i \wedge r \neq 0]$$

is negligible, where $g \in_R \mathbb{G}_1$, $h \in_R \mathbb{G}_2$, $a, b \in_R Z_p$, and $c_i, c \in Z_p$.

Definition 4 (n-DHE assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$\Pr[\mathcal{A}(g, h, , g^a, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}}, h^a, \ldots, h^{a^n}, h^{a^{n+2}}, \ldots, h^{a^{2n}}) = h^{a^{n+1}}]$$

is negligible, where $g \in_R \mathbb{G}_1$, $h \in_R \mathbb{G}_2$ and $a \in_R Z_p$.

2.3 Extended Accumulator with Efficient Updates

In Ref. [12], the accumulator with efficient updates is proposed. The accumulator is generated from a set of values, and we can verify that a single value is accumulated. Thus, for k values, we have to verify that each value is accumulated multiple times. This means that the complexity depends on the number of proved values, k. Here, we extend the accumulator to verify that k values are accumulated with the constant complexity.

Here, we consider that some values in $\{1, \ldots, n\}$ with size n are accumulated. Let V be a set of accumulated values that is a subset of $\{1, \ldots, n\}$. Let $U = \{i_1, \ldots, i_k\}$ be a subset of V with size k. The

© 2012 Information Processing Society of Japan

accumulator allows us to confirm that all elements of U belong to V, i.e., $U \subseteq V$, all at once.

AccSetup: This is the algorithm to output the public parameters. Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Select $g \in_R \mathbb{G}_1$ and $h \in_R \mathbb{G}_2$. Select $\gamma \in_R Z_p$ and compute and publish $p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$, $h_1 = h^{\gamma^1}, \ldots, h_n = h^{\gamma^n}, h_{n+2} = h^{\gamma^{n+2}}, \ldots, h_{2n} = h^{\gamma^{2n}}$ and $z = e(g,h)^{\gamma^{n+1}}$ as the public parameters.

AccUpdate: This is the algorithm to compute the accumulator using the public parameters. The accumulator acc_V of V is computed as $acc_V = \prod_{i \in V} g_{n+1-i}$.

AccWitUpdate: This is the algorithm to compute the witness that values are included in an accumulator, using the public parameters. Given V and the accumulator acc_V, the witness of values $i_1, \ldots, i_k$ in U is computed as $W = \prod_{\tilde{\imath} \in U} \prod_{j \in V}^{j \neq \tilde{\imath}} h_{n+1-j+\tilde{\imath}}$.

AccVerify: This is the algorithm to verify that all values in U are included in an accumulator, using the witness and the public parameters. Given acc_V, U, and W, accept if

$$\frac{e(acc_V, \prod_{\tilde{\imath} \in U} h_{\tilde{\imath}})}{e(g, W)} = z^k.$$

Theorem 1 Under the n-DHE assumption, any adversary cannot output (U, V, W) where $U \subseteq \{1, \ldots, n\}, V \subseteq \{1, \ldots, n\}$ s.t. **AccVerify** accepts U, acc_V, W and $U \setminus V \neq \emptyset$.

Proof. Assume an adversary which outputs (U, V, W) s.t. **AccVerify** accepts U, acc_V, W and $U \setminus V \neq \emptyset$. Let $U_1 = U \setminus V$ and $U_2 = U \cap V$. $U \setminus V \neq \emptyset$ (i.e., $U_1 \neq \emptyset$) implies $|U_2| \neq k$.

Since **AccVerify** accepts these,

$$\frac{e(acc_V, \prod_{\tilde{\imath} \in U} h_{\tilde{\imath}})}{e(g, W)} = z^k = e(g, h_{n+1})^k,$$

where $h_{n+1} = h^{\gamma^{n+1}}$. From $acc_V = \prod_{i \in V} g_{n+1-i}$,

$$\frac{e(\prod_{i \in V} g_{n+1-i}, \prod_{\tilde{\imath} \in U} h_{\tilde{\imath}})}{e(g, W)} = e(g, h_{n+1})^k,$$

$$e\left(g, \prod_{\tilde{\imath} \in U} \prod_{i \in V} h_{n+1-i+\tilde{\imath}}\right) = e(g, W {h_{n+1}}^k).$$

Thus, we have

$$\prod_{\tilde{\imath} \in U} \prod_{i \in V} h_{n+1-i+\tilde{\imath}} = W {h_{n+1}}^k,$$

$$\prod_{\tilde{\imath} \in U_1} \prod_{i \in V} h_{n+1-i+\tilde{\imath}} \cdot \prod_{\tilde{\imath} \in U_2} \prod_{i \in V} h_{n+1-i+\tilde{\imath}} = W {h_{n+1}}^k,$$

$$\left(\prod_{\tilde{\imath} \in U_1} \prod_{i \in V} h_{n+1-i+\tilde{\imath}}\right) \cdot {h_{n+1}}^{|U_2|} \prod_{\tilde{\imath} \in U_2} \prod_{i \in V, i \neq \tilde{\imath}} h_{n+1-i+\tilde{\imath}} = W {h_{n+1}}^k,$$

$$\prod_{\tilde{\imath} \in U_1} \prod_{i \in V} h_{n+1-i+\tilde{\imath}} \cdot \prod_{\tilde{\imath} \in U_2} \prod_{i \in V, i \neq \tilde{\imath}} h_{n+1-i+\tilde{\imath}} = W {h_{n+1}}^{k-|U_2|}.$$

We obtain

$$h_{n+1} = \left(W^{-1} \cdot \prod_{\tilde{\imath} \in U_1} \prod_{i \in V} h_{n+1-i+\tilde{\imath}} \cdot \prod_{\tilde{\imath} \in U_2} \prod_{i \in V, i \neq \tilde{\imath}} h_{n+1-i+\tilde{\imath}}\right)^{1/(k-|U_2|)},$$

where $k - |U_2| \neq 0$ due to $|U_2| \neq k$.

For any $\tilde{\imath} \in U_1$ and any $i \in V$, $h_{n+1-i+\tilde{\imath}} \neq h_{n+1}$, due to $U_1 \cap V = \phi$. Also, for any $\tilde{\imath} \in U_2$ and any $i \in V$ satisfying $i \neq \tilde{\imath}$, $h_{n+1-i+\tilde{\imath}} \neq h_{n+1}$. Therefore, we can compute h_{n+1} given $g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}$, $h_1, \ldots, h_n, h_{n+2}, \ldots, h_{2n}$, which contradicts n-DHE assumption. □

2.4 Modified BBS+ Signatures

We utilize an extension of BB signature scheme [7], called BBS+ signatures. The extension is informally introduced in Ref. [8] and the concrete construction is shown in Refs. [2], [15]. This scheme allows us to sign a set of messages. Our system requires that the accumulator is signed. In the BBS+ signature, the messages to be signed are set in exponents (elements of Z_p), whereas the accumulator is the product of g_i's from $\mathbb{G}_1$. Thus, we modify the BBS+ signature to be able to sign on g_i's, as follows.

mBBS+Setup: Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Select $g, g_0, \tilde{g}_1, \ldots, \tilde{g}_L \in_R \mathbb{G}_1$. Select $h \in_R \mathbb{G}_2$. Select $\gamma \in_R Z_p$ and compute $g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$.

mBBS+KeyGen: Select $X \in_R Z_p$ and compute $Y = g^X$ and $Z = h^X$. The secret key is X and the public key is $(p, \mathbb{G}_1, \mathbb{G}_2, e, g, g_0, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, \tilde{g}_1, \ldots, \tilde{g}_L, h, Y, Z)$.

mBBS+Sign: Given messages $m_1, \ldots, m_n, m_{n+2}, \ldots, m_{2n} \in \{0,1\}$, $M_1, \ldots, M_L \in Z_p$, select $w, r \in_R Z_p$ and compute

$$A = \left(\prod_{1 \le j \le 2n}^{j \neq n+1} g_j^{m_j} \prod_{1 \le j \le L} \tilde{g}_j^{M_j} g_0^r g\right)^{1/(X+w)}.$$

The signature is (A, w, r).

mBBS+Verify: Given messages $m_1, \ldots, m_n, m_{n+2}, \ldots, m_{2n}$, $M_1, \ldots, M_L$ and the signature (A, w, r), check

$$e(A, Z h^w) = e\left(\prod_{1 \le j \le 2n}^{j \neq n+1} g_j^{m_j} \prod_{1 \le j \le L} \tilde{g}_j^{M_j} g_0^r g, h\right).$$

The modified BBS+ signature is unforgeable against adaptively chosen message attack under the q-SDH assumption. It is shown in a similar way to Ref. [3], as follows.

BB signatures. Since the security is proved using the security of the underlying BB signatures [7], we briefly show the scheme.

BBSetup: Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Select $g \in_R \mathbb{G}_1$, $h \in_R \mathbb{G}_2$.

BBKeyGen: Select $X \in_R Z_p$ and compute $Y = g^X$ and $Z = h^X$. The secret key is X and the public key is $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, Y, Z)$.

BBSign: Given message $m \in Z_p$, compute $B = g^{1/(X+m)}$. The signature is B.

BBVerify: Given message m and the signature B, check $e(B, Z h^m) = e(g, h)$.

BB signatures are existentially unforgeable against *weak* chosen message attack under the q-SDH assumption [7]. In this attack, the adversary must choose messages queried for the signing oracle, before the public key is given.

Theorem 2 mBBS+ signature is unforgeable against adaptively chosen message attack under the q-SDH assumption.

Proof. This proof is derived from Ref. [3].

© 2012 Information Processing Society of Japan

Assume that $\mathcal{A}$ breaks the unforgeability of mBBS+ signatures, and we construct the following simulator $\mathcal{B}$ breaking BB signatures that are secure under the q-SDH assumption.

$\mathcal{B}$ chooses random messages $w_1, \ldots, w_{q-1}$ for BB signatures, and is given the corresponding BB signatures $B_i = g^{1/(X+w_i)}$ with the public key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, Y, Z)$. Then, $\mathcal{B}$ selects $w^*, k^*, a^* \in_R Z_p$, and compute $g_0 = ((Yg^{w^*})^{k^*} g^{-1})^{1/a^*} = g^{((X+w^*)k^*-1)/a^*}$. Also, $\mathcal{B}$ selects $\gamma, \mu_1, \ldots, \mu_L \in_R Z_p$, and compute $g_1 = g_0^{\gamma^1}, \ldots, g_n = g_0^{\gamma^n}, g_{n+2} = g_0^{\gamma^{n+2}}, \ldots, g_{2n} = g_0^{\gamma^{2n}}$, and $\tilde{g}_1 = g_0^{\mu_1}, \ldots, \tilde{g}_L = g_0^{\mu_L}$. $\mathcal{B}$ sets the public key of mBBS+ signatures $(p, \mathbb{G}_1, \mathbb{G}_2, e, g, g_0, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, \tilde{g}_1, \ldots, \tilde{g}_L, h, Y, Z)$, and runs $\mathcal{A}$. Out of q signing queries from $\mathcal{A}$, $\mathcal{B}$ randomly selects a query, which called $*$ query. For messages $(m_{1,i}, \ldots, m_{n,i}, m_{n+2,i}, \ldots, m_{2n,i}, M_{1,i}, \ldots, M_{L,i})$ of the i-th query, define

$$t_i = \sum_{1 \le j \le 2n}^{j \ne n+1} m_{j,i}\gamma^j + \sum_{1 \le j \le L} M_{j,i}\mu_j.$$

To the queries except $*$, $\mathcal{B}$ responds using the BB signature (B_i, w_i) as follows. $\mathcal{B}$ selects $r_i \in_R Z_p$, and compute $a_i = r_i + t_i$ and the following A_i.

$$\begin{aligned} A_i &= B_i^{(1-\frac{a_i+(w_i-w^*)a_ik^*}{a^*})} g^{\frac{a_i}{a^*}k^*} \\ &= B_i^{(1-\frac{a_i}{a^*})} g^{\frac{-(w_i-w^*)a_ik^*+a_ik^*(X+w_i)}{(X+w_i)a^*}} \\ &= B_i^{(1-\frac{a_i}{a^*})} (g^{\frac{a_i}{a^*}k^*})^{\frac{-w_i+w^*+X+w_i}{X+w_i}} \\ &= B_i g^{\frac{-a_i+a_ik^*(X+w^*)}{a^*(X+w_i)}} \\ &= B_i g_0^{\frac{a_i}{(X+w_i)}} = (gg_0^{a_i})^{\frac{1}{X+w_i}} \end{aligned}$$

$\mathcal{B}$ returns (A_i, w_i, r_i).

To the $*$ query, $\mathcal{B}$ sets $r^* = a^* - t_i$, computes $A^* = g^{k^*} = (gg_0^{a^*})^{1/(X+w^*)}$ and returns (A^*, w^*, r^*).

Finally, $\mathcal{A}$ outputs the forged signature (A', w', r') on message $(m'_1, \ldots, m'_n, m'_{n+2}, \ldots, m'_{2n}, M'_1, \ldots, M'_L)$. There are three cases. Define

$$a' = r' + \sum_{1 \le j \le 2n}^{j \ne n+1} m'_j\gamma^j + \sum_{1 \le j \le L} M'_j\mu_j.$$

- Case I [$w' \notin \{w_1, \ldots, w_q, w^*\}$]: $\mathcal{B}$ computes the following B'.

$$\begin{aligned} B' &= (A'g^{\frac{-k^*}{a^*}a'})^{\frac{a^*}{a^*-a'-k^*a'(w'-w^*)}} \\ &= ((gg^{\frac{(X+w^*)k^*a'-a'}{a^*}})^{\frac{1}{X+w'}} g^{\frac{-k^*}{a^*}a'})^{\frac{a^*}{a^*-a'-k^*a'(w'-w^*)}} \\ &= (g^{\frac{a^*+(X+w^*)k^*a'-a'-k^*a'(X+w')}{a^*(X+w')}})^{\frac{a^*}{a^*-a'-k^*a'(w'-w^*)}} \\ &= (g^{\frac{a^*-a'-k^*a'(w'-w^*)}{a^*(X+w')}})^{\frac{a^*}{a^*-a'-k^*a'(w'-w^*)}} = g^{\frac{1}{X+w'}} \end{aligned}$$

  This means that a BB signature for a new message w' is forged, which contradicts q-SDH assumption.

- Case II [($w' = w_i$ and $A' = A_i$ for some i) or ($w' = w^*$ and $A' = A^*$)]: Consider $w' = w_i$ and $A' = A_i$ (The other case is similar). From $A'^{X+w'} = A_i^{X+w_i}$, $gg_0^{a'} = gg_0^{a_i}$ holds and we obtain $a' = a_i$. Thus, letting $\Delta r = r' - r_i$, $\Delta m_j = m'_j - m_{j,i}$, and $\Delta M_j = M'_j - M_{j,i}$,

$$\Delta r + \sum_{1 \le j \le 2n}^{j \ne n+1} \Delta m_j\gamma^j + \sum_{1 \le j \le L} \Delta M_j\mu_j = 0.$$

  Some Δm_j is not 0 or some ΔM_j is not 0. If $\Delta M_j \ne 0$, the above equation means that we can compute μ_j in case that $\mu_j = \log_{g_0} \tilde{g}_j$ is unknown. This contradict the DL assumption and then the q-SDH assumption.

  If $\Delta m_j \ne 0$, we can compute $\gamma^j \bmod p$ and thus γ, given $g_0, g_0^{\gamma}, \ldots, g_0^{\gamma^n}, g_0^{\gamma^{n+2}}, \ldots, g_0^{\gamma^{2n}}$. This means that, given $g, g^{\gamma}, \ldots, g^{\gamma^{2n}}$, we can compute $(c, g^{1/(\gamma+c)})$ for any $c \in Z_p$, which contradicts the q-SDH assumption, where $q = 2n$.

- Case III [$w' \in \{w_1, \ldots, w_q, w^*\}$ and $A' \notin \{A_1, \ldots, A_q, A^*\}$]: With the probability $1/q$, $w' = w^*$. Then, from

$$A' = (gg_0^{a'})^{1/(X+w^*)} = g^{(a^*+a'(X+w^*)k^*-a')/(a^*(X+w^*))},$$

  compute the following B'.

$$\begin{aligned} B' &= (A'g^{\frac{-k^*a'}{a^*}})^{\frac{a^*}{a^*-a'}} \\ &= (g^{\frac{a^*-a'}{a^*(X+w^*)}})^{\frac{a^*}{a^*-a'}} \\ &= g^{\frac{1}{X+w^*}} \end{aligned}$$

  This means that a BB signature for a new message w^* is forged, which contradicts q-SDH assumption. □

The security proof assumes that valid g_j's are signed, instead of any element from $\mathbb{G}_1$. Thus, for proving the knowledge of this signature, we have to ensure the correctness of g_j's by other technique, the following F-secure BB signatures.

2.5 *F*-secure BB Signatures

We also adopt another variant of BB signature scheme, called F-secure signature [5].

***F*BBSetup:** Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with a prime order p and a bilinear map e. Select $g \in_R \mathbb{G}_1$ and $h, \hat{h} \in_R \mathbb{G}_2$.

***F*BBKeyGen:** Select $\tilde{X}, \hat{X} \in_R Z_p$ and compute $\tilde{Y} = g^{\tilde{X}}, \hat{Y} = g^{\hat{X}}$, $\tilde{Z} = h^{\tilde{X}}, \hat{Z} = h^{\hat{X}}$. The secret key is $(\tilde{X}, \hat{X})$ and the public key is $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, \hat{h}, \tilde{Y}, \hat{Y}, \tilde{Z}, \hat{Z})$.

The public key can be checked by $e(\tilde{Y}, h) = e(g, \tilde{Z})$ and $e(\hat{Y}, h) = e(g, \hat{Z})$.

***F*BBSign:** Given message $M \in Z_p$, select $\mu \in_R Z_p - \{\frac{\tilde{X}-M}{\hat{X}}\}$ and compute $S = h^{1/(\tilde{X}+M+\hat{X}\mu)}$, $T = \hat{Y}^{\mu}$, $U = \hat{h}^{\mu}$. The signature is (S, T, U).

***F*BBVerify:** Given the signature (S, T, U) on message M, check $e(\tilde{Y}g^MT, S) = e(g, h)$ and $e(T, \hat{h}) = e(\hat{Y}, U)$.

Define bijection F as $F(M) = (g^M, \hat{h}^M)$ for message M. The correctness of $F(M)$ can be checked by $e(g^M, \hat{h}) = e(g, \hat{h}^M)$. The F-security means that no adversary can output $(F(M), \sigma)$ where σ is the signature on message M s.t. he has never previously obtained the signature after his adaptive chosen message attacks. The security is proved under the q-HSDH and q-TDH assumptions [5].

2.6 Proving Relations on Representations

We adopt zero-knowledge proofs of knowledge (PKs) on representations, which are the generalization of the Schnorr identification protocol [11]. Concretely we utilize a PK proving the knowledge of a representation of $C \in \mathbb{G}_1$ to the bases $g_1, g_2, \ldots, g_t \in \mathbb{G}_1$, i.e., $x_1, \ldots, x_t$ s.t. $C = g_1^{x_1} \cdots g_t^{x_t}$. This can be also constructed on groups $\mathbb{G}_2$ and $\mathbb{G}_T$. The PK can be extended to proving multiple representations with equal parts.

© 2012 Information Processing Society of Japan

Since we use only prime-order groups, we can extract the proved secret knowledge given two accepting protocol views whose commitments are the same and whose challenges are different.

3. Proposed System

3.1 Construction Idea

As in Ref. [10], we categorize finite-set attributes and string attributes. In the finite-set attributes, the values are binary or from a pre-defined finite set, for example, gender, degree, nationality, etc. On the other hand, name and identification number are the string attributes.

Our proposal is based on the pairing-based anonymous credential system using the BBS+ signatures, which is described in Ref. [12] for example. In the underlying system, the certificate is a BBS+ signature [8], where each attribute type is expressed as an exponent on a base assigned to the attribute type, such as $\tilde{g}_j^{M_j}$, and all parts of $\tilde{g}_j^{M_j}$ have to be signed. Namely, the certificate is (A, w, r) s.t.

$$A = \left(\prod_{1 \le j \le L'} \tilde{g}_j^{M_j} \tilde{g}_{L'+1}^{x} g_0^r g \right)^{1/(X+w)},$$

where x is a secret identity that only the user with the certificate knows. Then, proving the knowledge of the signature needs the cost depending on the number of attribute types.

To express the finite-set attributes (For the string type, we still use the exponent), we use a pairing-based accumulator in Ref. [12]. Let all attribute values in all finite-set attribute types be numbered. The j-th attribute value is assigned to an input value g_j's in the accumulator. The multiple inputs (i.e., attribute values) are accumulated into a single value. When V is the set of indexes of the attribute values for a user, they are accumulated to $acc_V = \prod_{j \in V} g_{n+1-j}$. We consider that the accumulated value is signed by a modified BBS+ signature,

$$A = \left(acc_V \cdot \prod_{1 \le j \le L} \tilde{g}_j^{M_j} \tilde{g}_{L+1}^{x} g_0^r g \right)^{1/(X+w)},$$

where the original representation $\tilde{g}_j^{M_j}$ is still used for the string type.

However, in the PK of the modified BBS+ signature, acc_V is committed for secrecy. That is, the validity of the committed value (i.e., it is the form of acc_V) is unknown to the verifier. The PK for representations only proves the form of $A = (R \cdot \prod_{1 \le j \le L} \tilde{g}_j^{M_j} \tilde{g}_{L+1}^{x} g_0^r g)^{1/(X+w)}$, for some $R \in \mathbb{G}_1$. However, the security proof of the modified BBS+ signatures assumes that the message is the product of g_j's, i.e., $\prod_{1 \le j \le 2n}^{j \ne n+1} g_j^{m_j}$. For example, we can show the following forge by manipulating the value of acc_V:

$$acc_V = \prod_{1 \le j \le 2n}^{j \ne n+1} g_j^{m_j} \cdot \left(\prod_{1 \le j \le L} \tilde{g}_j^{-M_j} \right) \tilde{g}_{L+1}^{-x} \cdot g_0^{-r} g^{-1} Y g^w, \quad A = g.$$

It is unknown whether this forge is meaningful or not. However, we cannot prove the security of our protocols, if the validity of acc_V is unknown and the modified BBS+ signature is forgeable. Thus, we add another signature on acc_V by signing the exponent $\sum_{j \in V} \gamma^{n+1-j}$. This approach is also used in Ref. [12] to ensure the g_j in the membership certificate. They use a weakly secure BB signature [7], based on interactive HSDH assumption [4] or HSDHE assumption [12]. We consider that it is a rather strong assumption. This is why we use the F-secure BB signature [5] derived from fully secure BB signature, based on the better assumptions (HSDH assumption and TDH assumption).

AND relation. For AND relation $(a_1 \wedge \cdots \wedge a_k)$, it is needed to prove that a specified set of attributes $(a_1, \ldots, a_k)$ are all embedded into the user's certificate. Using **AccVerify** in the extended accumulator, we can prove that multiple values are accumulated to the accumulator in the certificate with constant complexity. By the similar way to Ref. [12], we can obtain the PK of **AccVerify** with constant complexity.

OR relation. For OR relation $(a_1 \vee \cdots \vee a_k)$, it is needed to prove that one (denoted as $\tilde{a}$) of a specified set of attributes $(a_1, \ldots, a_k)$ is embedded into the user's certificate. Similarly to AND relation, using **AccVerify**, a signer can prove that a value $\tilde{a}$ is accumulated to the accumulator in the certificate. Furthermore, the verifier prepares another accumulator acc' from specified attributes $a_1, \ldots, a_k$. Then, the signer proves that the same value $\tilde{a}$ is accumulated to the additional accumulator acc'.

3.2 Proposed Construction

The following construction uses the asymmetric pairings defined in Section 2.1. Our implementation shown later is based on a library of asymmetric pairings. By letting $\mathbb{G}_1 = \mathbb{G}_2$, our construction can be also implemented by symmetric pairings.

3.2.1 Setup

The inputs of this algorithm are ℓ, n, and L, where ℓ is the security parameter, n is the maximum number of finite-set attribute values, and L is the maximum number of string attribute types. The outputs are issuer's public key ipk and issuer's secret key isk.

(1) Select bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with the same order p with length ℓ and the bilinear map e.

(2) Select $g, g_0, \hat{g}, \tilde{g}_1, \ldots, \tilde{g}_{L+1} \in_R \mathbb{G}_1$ and $h, \hat{h}, \tilde{h} \in_R \mathbb{G}_2$. Select $X, \tilde{X}, \hat{X}, \tilde{X}', \hat{X}', \gamma \in_R Z_p^*$, compute $Y = g^X, Z = h^X, \tilde{Y} = g^{\tilde{X}}, \hat{Y} = g^{\hat{X}}, \tilde{Z} = h^{\tilde{X}}, \hat{Z} = h^{\hat{X}}, \tilde{Y}' = g^{\tilde{X}'}$ and $\hat{Y}' = g^{\hat{X}'}, \tilde{Z}' = h^{\tilde{X}'}$ and $\hat{Z}' = h^{\hat{X}'}$. Compute $g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$, $h_1 = h^{\gamma^1}, \ldots, h_n = h^{\gamma^n}, h_{n+2} = h^{\gamma^{n+2}}, \ldots, h_{2n} = h^{\gamma^{2n}}$, and $z = e(g, h)^{\gamma^{n+1}}$. Select hash function $H : \{0,1\}^* \to Z_p$.

(3) For $1 \le j \le n$, select $\mu_j \in_R Z_p - \{\frac{\tilde{X}' - \gamma^j}{\hat{X}'}\}$ and compute the F-secure BB signature on γ^j $(F(\gamma^j) = (g^{\gamma^j}, h^{\gamma^j}))$ as follows:

$$\tilde{S}_j = \tilde{h}^{1/(\tilde{X}' + \gamma^j + \mu_j \hat{X}')}, \quad \tilde{T}_j = \hat{Y}'^{\mu_j}, \quad \tilde{U}_j = h^{\mu_j}.$$

(4) Output the issuer public key $ipk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, H, g, \hat{g}, h, \hat{h}, \tilde{h}, g_0, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_n, h_{n+2}, \ldots, h_{2n}, \tilde{g}_1, \ldots, \tilde{g}_{L+1}, z, (\tilde{S}_1, \tilde{T}_1, \tilde{U}_1), \ldots, (\tilde{S}_n, \tilde{T}_n, \tilde{U}_n), Y, Z, \tilde{Y}, \hat{Y}, \tilde{Z}, \hat{Z}, \tilde{Y}', \hat{Y}', \tilde{Z}', \hat{Z}')$, and the issuer secret key $isk = (X, \tilde{X}, \hat{X}, \tilde{X}', \hat{X}', \gamma)$.

3.2.2 Issuing Certificate

This is an interactive protocol between the issuer **Issuer** and user **User**. The common inputs of this protocol consist of ipk, and (SA, FA) that are sets of string attribute values and finite-set attribute values of the user, respectively. Each string attribute

© 2012 Information Processing Society of Japan

value of the j-th attribute type in SA is represented by an element M_j from Z_p^* (If the user does not have any attribute value in the attribute type, we assign an attribute value implying not applicable). Each finite-set attribute value is represented by an index in $\{1,\ldots,n\}$. Thus, set SA consists of attribute values and set FA consists of indexes of attribute values (sets TSA and TFA shown later are similar). The input of **Issuer** is isk. The output of **User** is the certificate $cert$.

(1) [**User**] Select $x, r' \in_R Z_p^*$. Compute $A' = \tilde{g}_{L+1}^{x} g_0^{r'}$. Send A' to **Issuer**. In addition, prove the validity of A' using PK for representations, i.e., prove the knowledge of x, r' s.t. $A' = \tilde{g}_{L+1}^{x} g_0^{r'}$.

(2) [**Issuer**] Given the user's attributes (SA, FA), compute the accumulator of the finite-set attributes as $acc = \prod_{a\in \mathrm{FA}} g_{n+1-a}$. Select $w, r'' \in_R Z_p^*$. Compute the modified BBS+ signature as follows:

$$A = \left(acc\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right) A' g_0^{r''} g\right)^{1/(X+w)} = \left(acc\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right) \tilde{g}_{L+1}^{x} g_0^{r'+r''} g\right)^{1/(X+w)}.$$

In addition, select $\mu \in_R Z_p - \{\frac{\tilde{X}-\sum_{a\in \mathrm{FA}} \gamma^{n+1-a}}{\hat{X}}\}$ and compute an F-secure BB signature ensuring acc as follows:

$$S = h^{1/(\tilde{X}+\sum_{a\in \mathrm{FA}} \gamma^{n+1-a}+\mu\hat{X})}, \quad T = \hat{Y}^{\mu}, \quad U = \tilde{h}^{\mu}, \quad F = \tilde{h}^{\sum_{a\in \mathrm{FA}} \gamma^{n+1-a}}.$$

Return (A, S, T, U, F, w, r'') to **User**.

(3) [**User**] Compute $r = r' + r''$, verify:

$$e(A, Zh^{w}) \stackrel{?}{=} e\left(acc\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right) \tilde{g}_{L+1}^{x} g_0^{r} g, h\right)$$
$$\wedge\ e(\tilde{Y}\cdot acc\cdot T, S) \stackrel{?}{=} e(g,h) \wedge\ e(T,\tilde{h}) \stackrel{?}{=} e(\hat{Y}, U) \wedge e(acc, \tilde{h}) \stackrel{?}{=} e(g, F).$$

Output $cert = (A, S, T, U, F, x, w, r)$.

3.2.3 Attribute Proofs

This is a protocol between the user and the verifier. The common inputs are ipk, and (TSA, TFA) are subsets of string attributes and finite-set attributes respectively, which are referenced in proofs, and user's secret inputs are $cert$ and (SA, FA).

Proving AND Relation.

For TFA $= \{a_1,\ldots,a_k\}$ with $a_j \in \{1,\ldots,n\}$, the user shows his possession of the certificate which includes all of the attributes, i.e., $a_1 \wedge a_2 \wedge \ldots \wedge a_k$.

(1) The user computes the witness that $a_1,\ldots,a_k$ are included in the accumulator of FA as: $W = \prod_{1\le j\le k}(\prod_{a\in \mathrm{FA}}^{a\neq a_j} h_{n+1-a+a_j})$. Set $D = \prod_{1\le j\le k} h_{a_j}$.

(2) The user selects $\rho_A, \rho_S, \rho_T, \rho_U, \rho_F, \rho_a, \rho_W \in_R Z_p^*$, and compute commitments $C_A = A\hat{g}^{\rho_A}$, $C_S = S\hat{h}^{\rho_S}$, $C_T = T\hat{g}^{\rho_T}$, $C_U = U\hat{h}^{\rho_U}$, $C_F = F\hat{h}^{\rho_F}$, $C_a = acc\cdot\hat{g}^{\rho_a}$, and $C_W = W\hat{h}^{\rho_W}$.

(3) The user selects $\rho_w, \rho' \in_R Z_p^*$, sets $\alpha = w\rho_A$, $\zeta = \rho_S\rho_a$ and $\xi = \rho_S\rho_T$. The user computes auxiliary commitments $C_w = h^{w}\hat{h}^{\rho_w}$ and $C_{\rho_S} = h^{\rho_S}\hat{h}^{\rho'}$. Then, the user sets $\rho_\alpha = \rho_w\rho_A$, $\rho_\zeta = \rho'\rho_a$, and $\rho_\xi = \rho'\rho_T$.

(4) The user sends the commitments $(C_A, C_S, C_T, C_U, C_F, C_a, C_W, C_w, C_{\rho_S})$ to the verifier.

(5) By using the proof of knowledge (PK) for representations, the user proves the knowledge of $x, w, r, \rho_A, \rho_S, \rho_T, \rho_U, \rho_F, \rho_a, \rho_W, \rho_w, \rho', \alpha, \zeta, \xi, \rho_\alpha, \rho_\zeta, \rho_\xi$, and M_j for $M_j \notin$ TSA s.t.

$$C_w = h^{w}\hat{h}^{\rho_w}, 1 = C_w^{\rho_A} h^{-\alpha}\hat{h}^{-\rho_\alpha}, \tag{1}$$

$$e(C_A, Y)e\left(C_a\left(\prod_{1\le j\le L, M_j\in \mathrm{TSA}} \tilde{g}_j^{M_j}\right) g, h\right)^{-1} = \left(\prod_{1\le j\le L, M_j\notin \mathrm{TSA}} e(\tilde{g}_j, h)^{M_j}\right) e(\tilde{g}_{L+1}, h)^{x} e(g_0, h)^{r} \cdot e(\hat{g}, Y)^{\rho_A} e(\hat{g}, h)^{\alpha} e(C_A, h)^{-w} e(\hat{g}, h)^{-\rho_a}, \tag{2}$$

$$C_{\rho_S} = h^{\rho_S}\hat{h}^{\rho'}, 1 = C_{\rho_S}^{\rho_a} h^{-\zeta}\hat{h}^{-\rho_\zeta}, 1 = C_{\rho_S}^{\rho_T} h^{-\xi}\hat{h}^{-\rho_\xi}, \tag{3}$$

$$e(\tilde{Y}C_aC_T, C_S)e(g,h)^{-1} = e(\tilde{Y}C_aC_T, \hat{h})^{\rho_S} e(\hat{g}, C_S)^{\rho_a+\rho_T} e(\hat{g}, \hat{h})^{-\zeta-\xi}, \tag{4}$$

$$e(C_T, \tilde{h})e(\hat{Y}, C_U)^{-1} = e(\hat{g}, \tilde{h})^{\rho_T} e(\hat{Y}, \hat{h})^{-\rho_U}, \tag{5}$$

$$e(C_a, \tilde{h})e(g, C_F)^{-1} = e(\hat{g}, \tilde{h})^{\rho_a} e(g, \hat{h})^{-\rho_F}, \tag{6}$$

$$e(C_a, D)e(g, C_W)^{-1} z^{-k} = e(\hat{g}, D)^{\rho_a} e(g, \hat{h})^{-\rho_W}. \tag{7}$$

Proving OR Relation.

For TFA $= \{a_1,\ldots,a_k\}$, the user shows his possession of the certificate which includes one of the attributes, i.e., $a_1 \vee a_2 \vee \ldots \vee a_k$. Assume that $\tilde{a}$ is the proved attribute.

Before the protocol, the user and the verifier prepare another accumulator $acc' = \prod_{a_j\in \mathrm{TFA}} g_{n+1-a_j}$. This protocol is obtained by modifying the protocol of the AND relation, as follows.

(1) Similarly, the user computes $W = \prod_{a\in \mathrm{FA}}^{a\neq\tilde{a}} h_{n+1-a+\tilde{a}}$ for acc. Furthermore, the user computes the new witness $W' = \prod_{a_j\in \mathrm{TFA}}^{a_j\neq\tilde{a}} h_{n+1-a_j+\tilde{a}}$ for acc'.

(2) In addition to step 2 in AND relation, the user selects $\rho_g, \rho_{W'}, \rho_{\tilde{S}}, \rho_{\tilde{T}}, \rho_{\tilde{U}}, \rho_h \in_R Z_p^*$, and compute the new commitment $C_g = g_{\tilde{a}}\hat{g}^{\rho_g}$, $C_{W'} = W'\hat{h}^{\rho_{W'}}$, $C_{\tilde{S}} = \tilde{S}_{\tilde{a}}\hat{h}^{\rho_{\tilde{S}}}$, $C_{\tilde{T}} = \tilde{T}_{\tilde{a}}\hat{g}^{\rho_{\tilde{T}}}$, $C_{\tilde{U}} = \tilde{U}_{\tilde{a}}\hat{h}^{\rho_{\tilde{U}}}$, and $C_h = h_{\tilde{a}}\hat{h}^{\rho_h}$.

(3) In addition to step 3 in AND relation, the user selects $\tilde{\rho}, \tilde{\rho'} \in_R Z_p^*$, sets $\delta = \rho_h\rho_a$, $\tilde{\zeta} = \rho_{\tilde{S}}\rho_g$ and $\tilde{\xi} = \rho_{\tilde{S}}\rho_{\tilde{T}}$. The user computes auxiliary commitments $C_{\rho_h} = h^{\rho_h}\hat{h}^{\tilde{\rho}}$ and $C_{\rho_{\tilde{S}}} = h^{\rho_{\tilde{S}}}\hat{h}^{\tilde{\rho'}}$. Then, the user sets $\rho_\delta = \tilde{\rho}\rho_a$, $\rho_{\tilde{\zeta}} = \tilde{\rho'}\rho_g$, and $\rho_{\tilde{\xi}} = \tilde{\rho'}\rho_{\tilde{T}}$.

(4) The user sends the commitments $(C_A, C_S, C_T, C_U, C_F, C_g, C_a, C_W, C_{W'}, C_{\tilde{S}}, C_{\tilde{T}}, C_{\tilde{U}}, C_h, C_w, C_{\rho_S}, C_{\rho_h}, C_{\rho_{\tilde{S}}})$ to the verifier.

(5) Similarly to the AND relation, the user conducts the PK, where Eq. (7) is replaced by

$$C_{\rho_h} = g^{\rho_h}\hat{h}^{\tilde{\rho}}, 1 = C_{\rho_h}^{\rho_a} g^{-\delta}\hat{h}^{-\rho_\delta}, \tag{8}$$

$$e(C_a, C_h)e(g, C_W)^{-1} z^{-1} = e(\hat{g}, C_h)^{\rho_a} e(C_a, \hat{h})^{\rho_h} e(\hat{g}, \hat{h})^{-\delta} e(g, \hat{h})^{-\rho_W}, \tag{9}$$

and the following equations are added:

$$C_{\rho_{\tilde{S}}} = h^{\rho_{\tilde{S}}}\hat{h}^{\tilde{\rho'}}, 1 = C_{\rho_{\tilde{S}}}^{\rho_g} h^{-\tilde{\zeta}}\hat{h}^{-\rho_{\tilde{\zeta}}}, 1 = C_{\rho_{\tilde{S}}}^{\rho_{\tilde{T}}} h^{-\tilde{\xi}}\hat{h}^{-\rho_{\tilde{\xi}}}, \tag{10}$$

$$e(\tilde{Y'}C_gC_{\tilde{T}}, C_{\tilde{S}})e(g, \tilde{h})^{-1} = e(\tilde{Y'}C_gC_{\tilde{T}}, \hat{h})^{\rho_{\tilde{S}}} e(\hat{g}, C_{\tilde{S}})^{\rho_g+\rho_{\tilde{T}}} e(\hat{g}, \hat{h})^{-\tilde{\zeta}-\tilde{\xi}}, \tag{11}$$

$$e(C_{\tilde{T}}, h)e(\hat{Y'}, C_{\tilde{U}})^{-1} = e(\hat{g}, h)^{\rho_{\tilde{T}}} e(\hat{Y'}, \hat{h})^{-\rho_{\tilde{U}}}, \tag{12}$$

© 2012 Information Processing Society of Japan

$$e(C_g, h)e(g, C_h)^{-1} = e(\hat{g}, h)^{\rho_g} e(g, \hat{h})^{-\rho_h}, \quad (13)$$

$$e(acc', C_h)e(g, C_{W'})^{-1} z^{-1} = e(acc', \hat{h})^{\rho_h} e(\hat{g}, \hat{h})^{-\rho_{W'}}. \quad (14)$$

4. Security

Here, we show that the proposed protocols are the *PK*s for AND and OR relations on the finite-set attributes. The security on the string attributes can be proved in the similar way to the underlying protocols.

Theorem 3 The protocol of AND relation is a proof of knowledge of a modified BBS+ signature (A, w, r) on secret x, the string type of attributes $M_1, \ldots, M_L$, and the finite-set type of attributes indicated by accumulator acc, s.t. all attributes in TFA are accumulated to acc.

Proof. From the PK, we have an extractor of knowledge satisfying Eqs. (1)–(7). Using Eq. (1), we obtain $1 = (h^w \hat{h}^{\rho_w})^{\rho_A} h^{-\alpha} \hat{h}^{-\rho_\alpha}$, and thus $1 = h^{w\rho_A - \alpha} \hat{h}^{\rho_w \rho_A - \rho_\alpha}$. Since the discrete log of $\hat{h}$ to base h is unknown under the DL assumption (due to q-SDH assumption), this means $\alpha = w\rho_A$. By substituting this to Eq. (2), we have

$$e(C_A, Y)e\left(C_a\left(\prod_{1\le j\le L, M_j\in \mathrm{TSA}} \tilde{g}_j^{M_j}\right) g, h\right)^{-1}$$
$$= \left(\prod_{1\le j\le L, M_j\in \mathrm{TSA}} e(\tilde{g}_j, h)^{M_j}\right) e(\tilde{g}_{L+1}, h)^x e(g_0, h)^r e(\hat{g}, Y)^{\rho_A}$$
$$\cdot\, e(\hat{g}, h)^{w\rho_A} e(C_A, h)^{-w} e(\hat{g}, h)^{-\rho_a}$$
$$e(C_A, Y)e(\hat{g}^{-\rho_A}, Y)e(\hat{g}^{-\rho_A}, h^w)e(C_A, h^w)$$
$$= e\left(C_a\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right) g, h\right) e(\tilde{g}_{L+1}^x, h)e(g_0^r, h)e(\hat{g}^{-\rho_a}, h)$$
$$e(C_A\hat{g}^{-\rho_A}, Yh^w) = e(C_a\hat{g}^{-\rho_a}\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right)\tilde{g}_{L+1}^x g_0^r g, h)$$

Thus, we can extract $A = C_A\hat{g}^{-\rho_A}$ and $acc = C_a\hat{g}^{-\rho_a}$ s.t.

$$e(A, Yh^w) = e\left(acc\left(\prod_{1\le j\le L} \tilde{g}_j^{M_j}\right)\tilde{g}_{L+1}^x g_0^r g, h\right).$$

Similarly, using Eq. (3), we have $\zeta = \rho_S\rho_a$ and $\xi = \rho_S\rho_T$. By substituting them to Eq. (4), we have

$$e(\tilde{Y}C_aC_T, C_S)e(g, h)^{-1}$$
$$= e(\tilde{Y}C_aC_T, \hat{h})^{\rho_S} e(\hat{g}, C_S)^{\rho_a+\rho_T} e(\hat{g}, \hat{h})^{-\rho_S\cdot\rho_a-\rho_S\cdot\rho_T}$$
$$e(\tilde{Y}C_aC_T, C_S)e(\tilde{Y}C_aC_T, \hat{h}^{-\rho_S})e(\hat{g}^{-\rho_a-\rho_T}, C_S)e(\hat{g}^{-\rho_a-\rho_T}, \hat{h}^{-\rho_S})$$
$$= e(g, h)$$
$$e(\tilde{Y}C_a\hat{g}^{-\rho_a}C_T\hat{g}^{-\rho_T}, C_S\hat{h}^{-\rho_S}) = e(g, h)$$

Thus, for the extracted $acc = C_a\hat{g}^{-\rho_a}$, we can extract $S = C_S\hat{h}^{-\rho_S}$ and $T = C_T\hat{g}^{-\rho_T}$ s.t. $e(\tilde{Y}\cdot acc\cdot T, S) = e(g, h)$. Similarly, using Eqs. (5), (6), we obtain $U = C_U\hat{h}^{-\rho_U}$ and $F = C_F\hat{h}^{-\rho_F}$ s.t. $e(T, \tilde{h}) = e(\hat{Y}, U)$ and $e(acc, \tilde{h}) = e(g, F)$. Since F-secure BB signatures w.r.t. the public key $\tilde{Y}, \hat{Y}$ is issued on only accumulators, it means $acc = \prod_{a\in \mathrm{FA}} g_{n+1-a}$ for FA of a user (otherwise, the signature is forgeable).

On the other hand, using Eq. (7), we can similarly extract $W = C_W\hat{h}^{-\rho_W}$ s.t. $e(acc, D)e(g, W)^{-1} = z^k$ for $D = \prod_{1\le j\le k} h_{a_j}$. From the security of the extended accumulator, all values $a_1, \ldots, a_k$ are accumulated into acc. □

Theorem 4 The protocol of OR relation is a proof of knowledge of a modified BBS+ signature (A, w, r) on secret x, the string type of attributes $M_1, \ldots, M_L$, and the finite-set type of attributes indicated by accumulator acc, s.t. one of attributes in TFA is accumulated to acc.

Proof. From the PK, we have an extractor of knowledge satisfying the equations. Similarly to AND relation, we can extract a modified BBS+ signature (A, w, r) as the certificate including $acc = \prod_{a\in \mathrm{FA}} g_{n+1-a}$.

Similarly to the extraction of F-secure BB signature in the AND relation, using Eqs. (10)–(13), we can extract the F-secure BB signature $(\tilde{S}, \tilde{T}, \tilde{U})$ on $\tilde{G} = C_g\hat{g}^{-\rho_g}$ and $\tilde{F} = C_h\hat{g}^{-\rho_h}$ s.t. $e(\tilde{Y}'\tilde{G}\tilde{T}, \tilde{S}) = e(g, \tilde{h})$, $e(\tilde{T}, h) = e(\hat{Y}', \tilde{U})$ and $e(\tilde{G}, h) = e(g, \tilde{F})$. Since F-secure BB signatures w.r.t. the public key $\tilde{Y}', \hat{Y}'$ is issued on only $(g_j = g^{\gamma^j}, h_j = h^{\gamma^j})$'s, it means $\tilde{G} = g_{\tilde{a}}$ and $\tilde{F} = h_{\tilde{a}}$ (otherwise, the signature is forgeable).

Using Eq. (8), we can obtain $\delta = \rho_a\rho_h$. By substituting this into Eq. (9), we can extract $W = C_W\hat{g}^{-\rho_W}$ s.t. $e(acc, h_{\tilde{a}})e(g, W)^{-1} = z$ for the extracted $(g_{\tilde{a}}, h_{\tilde{a}})$. This means that attribute $\tilde{a}$ is accumulated into acc. Using Eq. (14), we can extract $W' = C_{W'}\hat{g}^{-\rho_{W'}}$ s.t. $e(acc', h_{\tilde{a}})e(g, W')^{-1} = z$ for $(g_{\tilde{a}}, h_{\tilde{a}})$. This means that attribute $\tilde{a}$ is also accumulated into acc', that is, attribute $\tilde{a}$ is one of attributes $a_1, \ldots, a_k$. □

5. Implementation and Experiments

5.1 Used Pairing Library

In the implementation of our system (and the compared conventional system), we utilize the fast pairing library called "Cross-twisted χ-based Ate (Xt-Xate) pairing" [1] with 254-bit group order and the embedding degree is 12, as in Ref. [20].

5.2 Comparisons to Conventional System

We compare the efficiency between our system and the conventional pairing-based system [12] using the BBS+ signatures. The anonymous credential system described in Ref. [12] does not equip proofs of AND and OR relations. However, similarly to the conventional RSA-based systems described in Ref. [10], we can construct the proof protocols for AND and OR relations, which are described in Appendix A.1. We introduce the following parameters.

- L: the total number of string attribute types
- $\tilde{L}$: the total number of finite-set attribute types (e.g., gender, profession)
- n: the total number of finite-set attribute values (e.g., male, female, student, teacher)
- k: the number of attributes referenced in a proof.

We measured the computation time of the systems using a laptop PC with the specifications shown in **Table 1**.

In this environment, the computation times for a pairing computation, an exponentiation on $\mathbb{G}_1$, an exponentiation on $\mathbb{G}_2$, and an exponentiation on $\mathbb{G}_T$ are about 13.69 ms, 1.84 ms, 3.52 ms, and 4.63 ms, respectively. We set $L = 3$, $\tilde{L}$ is varied from 5 to 100, and k is varied from 10 to 100. In the implementation, we use pre-computations for pairing calculations of fixed values such

© 2012 Information Processing Society of Japan

Table 1 Specifications of PC in experiments.

O/S	Ubuntu Linux kernel-2.6.35-30-generic
CPU	AMD E-350 Dual-Core Processor 1.6 GHz Cache size 512 KB
RAM	2 GB
Compiler	gcc-4.4 GNU C Compiler
Softwares	GMP Library 5.0.2, Openssl-0.9.8o

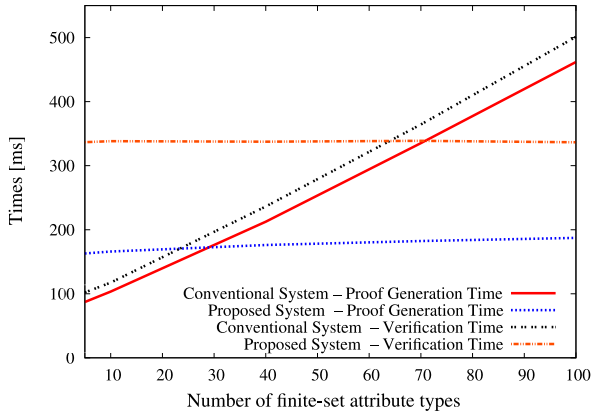

Fig. 1 AND Relation – processing times for $\tilde{L}$ (number of finite-set attribute types).

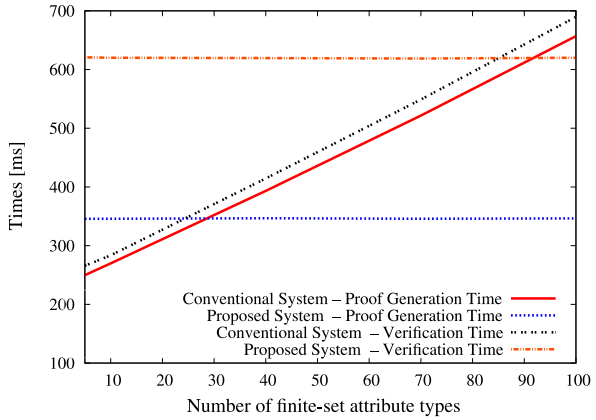

Fig. 2 OR Relation – processing times for $\tilde{L}$ (number of finite-set attribute types).

Table 2 Example of string and finite-set attributes.

String	Finite-set	Example Values
1) name	3) day of issuance	1–31
2) identity number	4) month of issuance	1–12
	5) year of issuance	2000–2011
	6) day of expiration	1–31
	7) month of expiration	1–12
	8) year of expiration	2000–2011
	9) gender	male,female
	10) day of birth	1–31
	11) month of birth	1–12
	12) year of birth	1930–2005
	13) marital status	single,marriage
	14-16) nationality	193 recognized states
	17) hometown	200 allocated cities
	18) city living	200 allocated cities
	19) residence status	citizen,immigrant,...
	20) religion	Moslem,Christian,...
	21) blood type	A,B,O,AB
	22-27) profession	student,teacher,...
	28-30) academic degree	B.S.,M.S,Ph.D.,...
	31-35) major	science,economic,...
	36-45) language	100 allocated lang.
	46-48) social benefit status	none, unemployed, ...
	49-51) eye and hair color	6 hair colors, 8 eye colors
	52-54) minority status	blind, deaf, ...
	...	

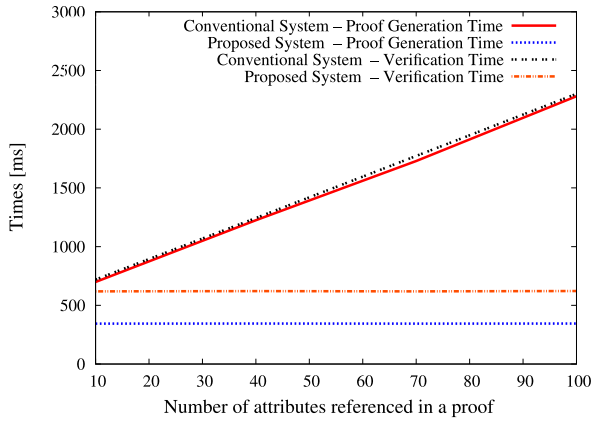

Fig. 3 OR Relation – impact of k (number of attributes referenced in a proof).

as public key and certificate, as in Ref. [17].

Figure 1 shows the comparisons for proof generation and verification times in case of AND relation. **Figure 2** shows them in case of OR relation. We vary $\tilde{L}$ from 5 to 100, and fix $k = 10$ and $n = 15,000$ (in the case of proposed system). In the proof generation, for $\tilde{L} < 30$, the time of the conventional system is better than the proposed system. In contrast, when $\tilde{L} > 30$, our proposed system becomes more efficient than the conventional one. The proof generation time of proposed system is constant at about 165 ms and 345 ms, for AND relation and OR relation, respectively. **Table 2** shows the example of attributes in eID. In the conventional system, if a user may own multiple attribute values from an attribute type, we have to prepare bases for the possible multiple values, namely $\tilde{L}$ increases by the number of possible multiple values. For example, a user can have multiple profession attributes such as student and technician in a company, and a user may own 5 or more language ability. As the results, $\tilde{L}$ becomes relatively large. Therefore, in the general case that $\tilde{L} > 30$, proving AND relation in our system has more efficiency.

Figure 3 shows the comparison for the OR relation proof in the variety of k. We vary k from 10 to 100, and set $L + \tilde{L} = 100$ and $n = 15,000$. As the result, our proposed system is not influenced by k, unlike the conventional system, which make it much more efficient.

On the other hand, the proposed system has a drawback that the size of public key varies from 1 MBytes to 10 MBytes when n is from 1,000 to 15,000, as shown in **Fig. 4**. However, in the current mobile environments, the data size is sufficiently practical, since the public key is not changed after it is distributed.

5.3 Comparisons to Camenisch-Groß System

Next, we compare our pairing-based system to RSA-based Camenisch-Groß system. For the implementation of Camenisch-Groß System, the idemix (Identity Mixer) library is available [16]. The library is implemented by Java, while our implementation uses C language. Thus, this is not a fair comparison, but for reference we show the comparisons. The specifications are the same as Table 1, where Java version is 1.6.0_26 and idemix library version is 2.3.3. The size of RSA modulus is set as 3,072 bits, to adjust the our implementation based on 254-bit

© 2012 Information Processing Society of Japan

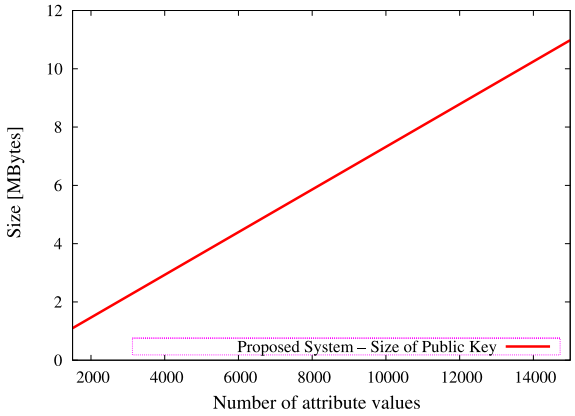

Fig. 4 Public key sizes for n (number of attribute values).

Table 3 Comparisons to Camenisch-Groß system (AND proof).

System	Proof generation time	Verification time	Proof size
Camenisch-Groß	8,901 ms	6,815 ms	2,517 Bytes
Ours	165 ms	325 ms	1,256 Bytes

Table 4 Comparisons to Camenisch-Groß system (OR proof).

System	Proof generation time	Verification time	Proof size
Camenisch-Groß	17,699 ms	14,589 ms	6,009 Bytes
Ours	345 ms	618 ms	2,184 Bytes

ECC. **Tables 3**, **4** show the comparisons for AND proof and OR proof, respectively. Note that both systems achieve the constant complexity. As for the processing times, our system is extremely better, although the implementation languages are different. As for the proof sizes, our system is also better, since data size for one element in the proof is shorter than the RSA-based system.

5.4 Application to eID

5.4.1 Summary of Implementation

To confirm the practicality of our system, we implemented a prototype system of eID application. At first, the issuer publishes its public key ipk. Then, the user registers himself along with particular attributes (SA, FA) to the issuer for certification by using the Issuing protocol via a secure channel. Based on the issued certificate, he requests a service to the Service Provider (SP). Then, the SP specifies attributes that the SP wants to know. This specification forms AND or OR relation, depending on the SP's requirement. Then, the user generates a proof for the possession of certificate w.r.t. the specified attribute(s) and shows it to the SP (verifier) anonymously by using the attribute proof protocol. If and only if the verification of user's proof is valid, the SP grants the user to access a requested service.

We implemented this prototype system using Java through the Java GUI and Java applet at the user and Java servlet at the Service Provider (SP) and the issuer, since the Java applet and Java servlet communications are often used for the web-based applications. The communication between the user and the servers (i.e., the issuer and SP) is over http connections. In our implementation, since algorithms of our anonymous credential system are implemented by C language as the middle-ware, we use Java Native Interface (JNI) as the interface between C and Java.

5.4.2 Experimental Results

The devices and software specifications used in this experiment are shown in **Table 5**. We used a laptop PC with an atom CPU for the user to simulate the mobile situation. The user PC is connected to the SP and issuer PCs via wireless LAN (IEEE802.11 b/g).

Table 5 Specification of devices in experiments.

Verifier/ Issuer	Intel Core i5 CPU 650 3.2 GHz, 2 GB RAM Ubuntu Linux 10.10 kernel-2.6.35 gcc-4.4.4, OpenSSL-0.9.8o, GMP-5.0.2 Java-1.6.0_26, apache-tomcat-7.0.20
Prover/ Joining-User	Intel Atom 1.5 GHz, 1 GB RAM Windows 7 Atheros AR9285 Wireless Adapter MinGW-5.1.6, OpenSSL-0.9.8g, GMP-4.3.1, Java-1.6.0_24

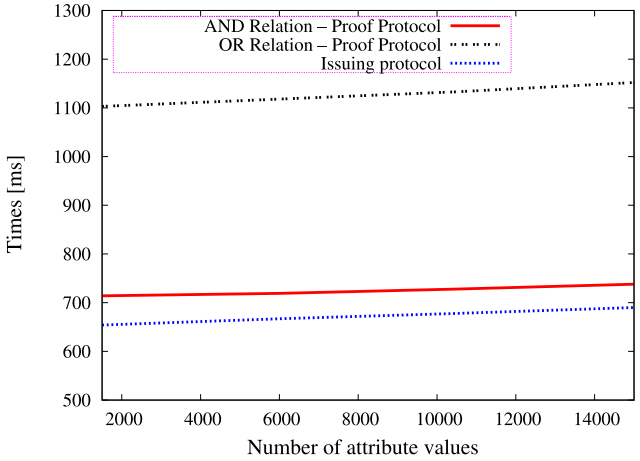

Fig. 5 Total processing times.

Figure 5 shows the total processing times of the issuing protocol and proof protocol. The total time of the proof protocol includes the proof generation time, the verification time, and the communication time during the process. This time varies from 714 ms to 738 ms for AND relation, when n is from 1,500 to 15,000. It is from 1,103 ms to 1,152 ms for the OR relation. The communication time is measured from sending the proof until receiving the response of the proof process from the verifier without verification time. The communication time is about 120 ms. The time of the issuing protocol varies from 654 ms to 690 ms when n increases from 1,500 to 15,000. In every case, we can confirm the practicality of our system for a mobile PC.

6. Conclusion

In this paper, for a pairing-based anonymous credential system, we have showed how to prove AND and OR relations on attributes with constant complexity in the number of finite-set attributes. From the experiments for the implementations, we showed the more efficiency than the conventional system and the RSA-based system, and the practicality in the mobile applications. The compensation is the increase of the public key size, although the public key is not changed after it is distributed.

Our future works include the implementations in smartphones and their applications to network services.

References

[1] Akane, M., Nogami, Y. and Morikawa, Y.: Fast Ate Pairing Computation of Embedding Degree 12 Using Subfield-twisted Eliptic Curve,

© 2012 Information Processing Society of Japan

IEICE Trans. Fundamentals, Vol.E92-A, No.2, pp.508–516 (2009).
[2] Au, M.H., Susilo, W. and Mu, Y.: Constant-Size Dynamic k-TAA, *Security in Communication Networks: 5th International Conference, SCN 2006*, LNCS 4116, pp.111–125, Springer-Verlag (2006).
[3] Au, M.H., Susilo, W. and Mu, Y.: Constant-Size Dynamic k-TAA, Cryptology ePrint Archive: Report 2008/136 (2008). This is the extended version of Ref. [2].
[4] Belenkiy, M., Chase, M., Kohlweiss, M. and Lysyanskaya, A.: Non-Interactive Anonymous Credentials, Cryptology ePrint Archive: Report 2007/384 (2007).
[5] Belenkiy, M., Chase, M., Kohlweiss, M. and Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials, *Proc. 5th Theory of Cryptography Conference* (*TCC 2008*), LNCS 4948, pp.356–374, Springer-Verlag (2008).
[6] Bichsel, P., Camenisch, J., Groß, T. and Shoup, V.: Anonymous credentials on a standard java card, *Proc. ACM Conference on Computer and Communications Security 2009* (*ACM-CCS'09*), pp.600–610 (2009).
[7] Boneh, D. and Boyen, X.: Short Signatures Without Random Oracles, *Journal of Cryptology*, Vol.21, No.2, pp.149–177 (2008). Extended abstract in *Proc. Eurocrypt 2004*, LNCS 3027, pp.223–238 (2004).
[8] Boneh, D., Boyen, X. and Shacham, H.: Short Group Signatures, *Advances in Cryptology — CRYPTO 2004*, LNCS 3152, pp.41–55, Springer-Verlag (2004).
[9] Boyen, X. and Waters, B.: Full-Domain Subgroup Hiding and Constant-Size Group Signatures, *Proc. 10th International Conference on Theory and Practice of Public-Key Cryptography* (*PKC 2007*), LNCS 4450, pp.1–15, Springer-Verlag (2007).
[10] Camenisch, J. and Groß, T.: Efficient attributes for anonymous credentials, *Proc. ACM Conference on Computer and Communications Security 2008* (*ACM-CCS'08*), pp.345–356 (2008).
[11] Camenisch, J., Kiayias, A. and Yung, M.: On the Portability of Generalized Schnorr Proofs, *Advances in Cryptology — EUROCRYPT 2009*, LNCS 5479, pp.425–442, Springer-Verlag (2009).
[12] Camenisch, J., Kohlweiss, M. and Soriente, C.: An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials, *Proc. 12th International Conference on Practice and Theory in Public Key Cryptography* (*PKC 2009*), LNCS 5443, pp.481–500, Springer-Verlag (2009).
[13] Camenisch, J. and Lysyanskaya, A.: Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials, *Advances in Cryptology — CRYPTO 2002*, LNCS 2442, pp.61–76, Springer-Verlag (2002).
[14] Cramer, R., Damgård, I. and Schoenmakers, B.: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols, *Advances in Cryptology — CRYPTO '94*, LNCS 839, pp.174–187, Springer-Verlag (1994).
[15] Furukawa, J. and Imai, H.: An Efficient Group Signature Scheme from Bilinear Maps, *Proc. 10th Australasian Conference on Information Security and Privacy* (*ACISP 2005*), LNCS 3574, pp.455–467, Springer-Verlag (2005).
[16] Identity Mixer: available from ⟨http://idemix.wordpress.com/⟩.
[17] Nakanishi, T. and Funabiki, N.: Short Verifier-Local Revocation Group Signature Scheme with Backward Unlinkability, *IEICE Trans. Fundamentals*, Vol.E90-A, No.9, pp.1793–1802 (2007).
[18] Sudarsono, A., Nakanishi, T. and Funabiki, N.: Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System, *Proc. 11th Privacy Enhancing Technologies Symposium* (*PETS 2011*), LNCS 6794, pp.246–263, Springer-Verlag (2011).
[19] Sudarsono, A., Nakanishi, T. and Funabiki, N.: An Implementation of a Pairing-Based Anonymous Credential System with Constant Complexity, *Proc. International MultiConference of Engineers and Computer Scientists 2011* (*IMECS 2011*), pp.630–635 (2011).
[20] Sudarsono, A., Nakanishi, T., Nogami, Y. and Funabiki, N.: Anonymous IEEE802.1X Authentication System Using Group Signatures, *IPSJ Journal*, Vol.51, No.3, pp.691–704 (2010).

Appendix

A.1 Proving AND and OR Relations in Conventional System

For the reference, we describe proving AND and OR relations in the conventional system.

A.1.1 Certificate

Let L' be the total number of attribute types. Then, the certificate is as follows.

$$A = \left(\left(\prod_{1\le j\le L'} \tilde{g}_j^{M_j}\right)\tilde{g}_{L'+1}^{x} g_0^{r} g\right)^{1/(X+w)}.$$

A.1.2 Proving AND relation

Let TA be the set of attributes referenced in the proof. Similarly to the proposed system, compute C_A, C_w. Then, prove the knowledge of $x, w, r, \rho_A, \rho_w, \alpha, \rho_\alpha$ and M_j for $M_j \notin$ TA s.t.

$$\begin{aligned}
&C_w = h^w \hat{h}^{\rho_w}, 1 = C_w^{\rho_A} h^{-\alpha} \hat{h}^{-\rho_\alpha},\\
&e(C_A, Y) e\left(\left(\prod_{1\le j\le L', M_j\in \mathrm{TA}} \tilde{g}_j^{M_j}\right) g, h\right)^{-1}\\
&\quad = \left(\prod_{1\le j\le L', M_j\notin \mathrm{TA}} e(\tilde{g}_j, h)^{M_j}\right) e(\tilde{g}_{L'+1}, h)^{x}\\
&\qquad \cdot e(g_0, h)^{r} e(\hat{g}, Y)^{\rho_A} e(\hat{g}, h)^{\alpha} e(C_A, h)^{-w}.
\end{aligned}$$

A.1.3 Proving OR relation

Let TA= $\{M'_{j_1}, \ldots, M'_{j_k}\}$ be the set of attributes referenced in the proof, where j_i means the j_i-th attribute types. Let STA be the set of j_i. Similarly to the proposed system, compute C_A, C_w, and additionally $C_j = h^{M_j} \hat{h}^{\rho_j}$ for $\rho_j \in_R Z_p^*$ with $j \in$ STA. Then, prove the knowledge of $x, w, r, \rho_A, \rho_w, \alpha, \rho_\alpha$, all M_j, and $\rho_{j'}$ for $j' \in$ STA s.t.

$$\begin{aligned}
&C_w = h^w \hat{h}^{\rho_w}, 1 = C_w^{\rho_A} h^{-\alpha} \hat{h}^{-\rho_\alpha},\\
&e(C_A, Y) e(g, h)^{-1}\\
&\quad = \left(\prod_{1\le j\le L', j\in \mathrm{STA}} e(\tilde{g}_j, h)^{M_j}\right)\left(\prod_{1\le j\le L', j\notin \mathrm{STA}} e(\tilde{g}_j, h)^{M_j}\right)\\
&\qquad \cdot e(\tilde{g}_{L'+1}, h)^{x} e(g_0, h)^{r} e(\hat{g}, Y)^{\rho_A} e(\hat{g}, h)^{\alpha} e(C_A, h)^{-w},\\
&C_j = h^{M_j} \hat{h}^{\rho_j} (\text{for } j \in \mathrm{STA}),
\end{aligned}$$

Additionally, prove

$$C_{j_1}/h^{M'_{j_1}} = \hat{h}^{\rho_{j_1}} \vee \cdots \vee C_{j_k}/h^{M'_{j_k}} = \hat{h}^{\rho_{j_k}}.$$

This *PK* for OR relation on representations is described in Ref. [14].

Editor's Recommendation

Scalability of an anonymous credential system is a critical factor for wide and practical use of the system. By using a pairing-based accumulator, this paper proposes an efficient scheme which achieves the constant complexity regarding the number of user attributes. In addition to this theoretical efficiency, this paper shows a real implementation of a mobile eID system based on the proposed scheme. Thus the proposal has both theoretical and practical impacts on privacy-enhancing technologies.

(Program chair of CSS2010, Kanta Matsuura)

© 2012 Information Processing Society of Japan

Amang Sudarsono received his B.E. degree in electrical engineering, telecommunication and multimedia program from Sepuluh Nopember Institute of Technology, Indonesia, in 2001. He received his Ph.D. degree in communication network engineering from Okayama University, Japan, in 2011. From 1997 to 2002, he was with the Network Engineering Division, Metro Cellular Nusantara, Ltd., Indonesia. He joined the Department of Telecommunication Technology at Electronics Engineering Polytechnic Institute of Surabaya, Indonesia, as a lecturer in 2002. His research interests include group signatures and network securities.

Toru Nakanishi received his M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000, respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor and an associate professor in 2003 and 2006, respectively. His research interests include cryptography and information security. He is a member of IEICE.

Nobuo Funabiki received his B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received his M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000-2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering at Okayama University as a professor. His research interests include network protocols, optimization algorithms, image processing, educational technology, and network security. Dr. Funabiki is a member of IEICE and IEEE.

© 2012 Information Processing Society of Japan