Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System

Amang Sudarsono[1]*, Toru Nakanishi[2], and Nobuo Funabiki[2]

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

Abstract. An anonymous credential system allows the user to convince a verifier of the possession of a certificate issued by the issuing authority anonymously. One of the applications is the privacy-enhancing electronic ID (eID). A previously proposed anonymous credential system achieves constant complexity in the number of finite-set attributes of the user. However, the system is based on the RSA. In this paper, we show how to achieve the constant complexity in a pairing-based anonymous credential system excluding the RSA. The key idea of the construction is the use of a pairing-based accumulator. The accumulator outputs a constant-size value from a large set of input values. Using zero-knowledge proofs of pairing-based certificates and accumulators, we can prove AND and OR relations with constant complexity in the number of finite-set attributes.

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 services provided by the organization. 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 can be desired. For example, a service provider can refuse the access from 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 [13], [12], [10] are one of the solutions.

Anonymous credential systems allow an issuer to issue a certificate to a user. Each certificate is a proof of membership, qualification, or privilege, and contains user's attributes. The user can anonymously convince a verifier for the possession of the certificate, where the selected attributes can be disclosed without revealing any other information about the user's privacy. The user can prove complex

* This work was done while this author was with Okayama university.

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 [5].

In [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 [10], Camenisch and Groß extended the scheme to solve the drawback. They classify 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 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.

In this paper, for a pairing-based anonymous credential system using BBS+ signatures [7], we show how to prove AND and OR relations with constant complexity. The key idea of the construction is the use 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 200 KBytes to 2 MBytes for the number of attribute values $1,000$ to $10,000$. In the current mobile environments, the data size is sufficiently practical, since the public key is not changed after it is distributed.

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, we consider the set of attribute values except for the attribute value targeted by NOT relation. Then, proving that an attribute value in the set is in his certificate means that the target attribute value is not in the certificate. For example, for proving that the user is not student, we can prove that she has some of other profession attribute values.

2 Preliminaries

2.1 Bilinear Groups

Our scheme utilizes the following bilinear groups:

1. $\mathcal{G}$ and $\mathcal{T}$ are multiplicative cyclic groups of prime order p,
2. g is a randomly chosen generator of $\mathcal{G}$,
3. e is an efficiently computable bilinear map: $\mathcal{G} \times \mathcal{G} \to \mathcal{T}$, i.e., (1) for all $u, u', v, v' \in \mathcal{G}$, $e(uu', v) = e(u, v)e(u', v)$ and $e(u, vv') = e(u, v)e(u, v')$, and thus for all $u, v \in \mathcal{G}$ and $a, b \in Z$, $e(u^a, v^b) = e(u, v)^{ab}$, and (2) $e(g, g) \neq 1$.

2.2 Assumptions

The security of our scheme is based on the q-SDH assumption [7, 8], the q-HSDH (Hidden SDH) assumption [9], and q-TDH (Triple DH) assumption [4] 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*

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

is negligible, where $u \in_R \mathcal{G}$ and $a \in_R Z_p$.

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

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

is negligible, where $u, v \in_R \mathcal{G}$, $a \in_R Z_p$, and $b, b_i \in Z_p$.

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

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

is negligible, where $u \in_R \mathcal{G}$, $a, b \in_R Z_p$, and $c_i, c \in Z_p$.

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

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

is negligible, where $u \in_R \mathcal{G}$ and $a \in_R Z_p$.

2.3 Extended Accumulator with Efficient Updates

In [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 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 $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g \in_R \mathcal{G}$. Select $\gamma \in_R Z_p$ and compute and publish $p, \mathcal{G}, \mathcal{T}, e, g, g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$ and $z = e(g, g)^{\gamma^{n+1}}$ 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 $W = \prod_{\tilde{\imath} \in U} \prod_{j \in V}^{j \neq \tilde{\imath}} g_{n+1-j+\tilde{\imath}}$.

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

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

Theorem 1. *Under the n-DHE assumption, any adversary cannot output* (U, V, W) *where* $U \subseteq \{1, \ldots, n\}, V \subseteq \{1, \ldots, n\}$ *on input* $p, \mathcal{G}, \mathcal{T}, e, g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}$ *and* z *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,

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

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

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

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

Thus, we have

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

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

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

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

We obtain

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

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

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

2.4 Modified BBS+ Signatures

We utilize an extension from BB signature scheme [6], called BBS+ signatures. The extension is informally introduced in [7] and the concrete construction is shown in [15, 1]. 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 $\mathcal{G}$. Thus, we modify the BBS+ signature to be able to sign on g_i's, as follows.

mBBS+Setup: Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g, g_0, h_1, \ldots, h_L \in_R \mathcal{G}$. 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 $Y = h^X$. The secret key is X and the public key is $(p, \mathcal{G}, \mathcal{T}, e, g, g_0, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_L, Y)$.

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

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

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

$$e(A, Yg^w) = e(\prod_{1 \le j \le 2n}^{j \ne n+1} g_j^{m_j} \prod_{1 \le j \le L} h_j^{M_j} g_0^r g, g).$$

The modified BBS+ signature is unforgeable against adaptively chosen message attack under the q-SDH assumption. It is shown in a similar way to [2], as follows.

BB signatures. Since the security is proved using the security of the underlying BB signatures [6], we briefly show the scheme.

BBSetup: Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g \in_R \mathcal{G}$.
BBKeyGen: Select $X \in_R Z_p$ and compute $Y = g^X$. The secret key is X and the public key is $(p, \mathcal{G}, \mathcal{T}, e, g, Y)$.
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 $e(B, Yg^m) = e(g, g)$.

BB signatures are existentially unforgeable against *weak* chosen message attack under the q-SDH assumption [6]. 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 [2].

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 $(p, \mathcal{G}, \mathcal{T}, e, g, Y)$. 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 $h_1 = g_0^{\mu_1}, \ldots, h_L = g_0^{\mu_L}$. $\mathcal{B}$ sets the public key of mBBS+ signatures $(p, \mathcal{G}, \mathcal{T}, e, g, g_0, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_L, Y)$, 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.

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

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

To the $*$ query, $\mathcal{B}$ sets $r^* = a^* - t_i$, computes $A^* = g^{k^*} = (g g_0^{a^*})^{1/(X+w^*)}$ 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

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

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

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

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 ${A'}^{X+w'} = {A_i}^{X+w_i}$, $g g_0^{a'} = g g_0^{a_i}$ holds and we obtain $a' = a_i$. Thus, letting $\Delta r = r' - r_i$, $\Delta m_j = m'_j - m_{j,i}$, and $\Delta M_j = M'_j - M_{j,i}$,

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

Some Δm_j is not 0 or some ΔM_j is not 0. If $\Delta M_j \neq 0$, the above equation means that we can compute μ_j in case that $\mu_j = \log_{g_0} h_j$ is unknown. This contradict the DL assumption and then the q-SDH assumption.
If $\Delta m_j \neq 0$, we can compute $\gamma^j \mod p$ 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

$$A' = (g g_0^{a'})^{1/(X+w^*)} = g^{(a^* + a'(X+w^*)k^* - a')/(a^*(X+w^*))},$$

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 $\mathcal{G}$. 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 [4].

F**BBSetup:** Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order p and a bilinear map e. Select $g, \tilde{g} \in_R \mathcal{G}$.

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

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

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

Define bijection F as $F(M) = (g^M, \tilde{g}^M)$ for message M. The F-security means that no adversary cannot 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 [4].

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 $C \in \mathcal{G}$ to the bases $g_1, g_2, \ldots, g_t \in \mathcal{G}$, 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 group $\mathcal{T}$. The PK can be extended to proving multiple representations with equal parts.

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 [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 [12] for example. In the underlying system, the certificate is a BBS+ signature [7], where each attribute type is expressed as an exponent on a base assigned to the attribute type, such as $g_j^{M_j}$, and all parts of $g_j^{M_j}$ have to be signed. Namely, the certificate is (A, w, r) s.t.

$$A = (\prod_{1 \le j \le L'} h_j^{M_j} {h_{L'+1}}^x g_0^r g)^{1/(X+w)},$$

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 [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 an extended BBS+ signature,

$$A = (acc_V \cdot \prod_{1 \le j \le L} h_j^{M_j} {h_{L+1}}^x g_0^r g)^{1/(X+w)},$$

where the original representation $h_j^{M_j}$ is still used for the string type.

However, in the PK of the extended 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 $A = (R \cdot \prod_{1 \le j \le L} h_j^{M_j} {h_{L+1}}^x g_0^r g)^{1/(X+w)}$, for some $R \in \mathcal{G}$. 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:

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

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 [12] to ensure the g_j in the membership certificate. They use a weakly secure BB signature [6], based on interactive HSDH assumption [3] or HSDHE assumption [12]. We consider that it is a rather strong assumption. This is why we use the F-secure BB signature [4] 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 [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

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 $\mathcal{G}$, $\mathcal{T}$ with the same order p with length ℓ and the bilinear map e.
2. Select $g, g_0, \tilde{g}, \hat{g}, h_1, \ldots, h_{L+1} \in_R \mathcal{G}$. Select $X, \tilde{X}, \hat{X}, \tilde{X'}, \hat{X'}, \gamma \in_R Z_p^*$, compute $Y = g^X, \tilde{Y} = g^{\tilde{X}}$, $\hat{Y} = g^{\hat{X}}$, $\tilde{Y'} = g^{\tilde{X'}}$ and $\hat{Y'} = g^{\hat{X'}}$. Compute $g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$, and $z = (g, g)^{\gamma^{n+1}}$. Select hash function $H : \{0,1\}^* \to Z_p$.
3. For every $g_j = g^{\gamma^j}$ with $1 \le j \le n$, select $\mu_j \in_R Z_p - \{\frac{\tilde{X'} - \gamma^j}{\hat{X'}}\}$ and compute the F-secure BB signature on g_j as follows:

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

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

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 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 $A' = {h_{L+1}}^x g_0^{r'}$. 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. $A' = {h_{L+1}}^x g_0^{r'}$.
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:
$$A = (acc(\prod_{1 \le j \le L} h_j^{M_j}) A' g_0^{r''} g)^{1/(X+w)} = (acc(\prod_{1 \le j \le L} h_j^{M_j}) h_{L+1}^x g_0^{r'+r''} g)^{1/(X+w)}.$$
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:
$$S = g^{1/(\tilde{X} + \sum_{a \in \mathrm{FA}} \gamma^{n+1-a} + \mu \hat{X})}, \quad T = \hat{Y}^{\mu}, \quad U = \tilde{g}^{\mu}, \quad F = \tilde{g}^{\sum_{a \in \mathrm{FA}} \gamma^{n+1-a}}.$$
Return (A, S, T, U, F, w, r'') to **User**.
3. [**User**] Compute $r = r' + r''$, verify:
$$e(A, Yg^w) \stackrel{?}{=} e(acc(\prod_{1 \le j \le L} h_j^{M_j}) h_{L+1}^x g_0^r g, g)$$
$$\wedge\ e(S, \tilde{Y} \cdot acc \cdot T) \stackrel{?}{=} e(g, g) \wedge\ e(\tilde{g}, T) \stackrel{?}{=} e(U, \hat{Y}) \wedge e(\tilde{g}, acc) \stackrel{?}{=} e(F, g).$$
Output $cert = (A, S, T, U, F, x, w, r)$.

Attribute Proofs. This is an interactive 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 prover 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 prover computes the witness that $a_1, \ldots, a_k$ are included in the accumulator of FA as: $W = \prod_{1 \le j \le k} (\prod_{a \in \mathrm{FA}}^{a \ne a_j} g_{n+1-a+a_j})$. Set $D = \prod_{1 \le j \le k} g_{a_j}$.

2. The prover selects $\rho_A, \rho_S, \rho_T, \rho_U, \rho_F, \rho_a, \rho_W \in_R Z_p^*$, and compute commitments $C_A = A\hat{g}^{\rho_A}$, $C_S = S\hat{g}^{\rho_S}$, $C_T = T\hat{g}^{\rho_T}$, $C_U = U\hat{g}^{\rho_U}$, $C_F = F\hat{g}^{\rho_F}$, $C_a = acc \cdot \hat{g}^{\rho_a}$, and $C_W = W\hat{g}^{\rho_W}$.
3. The prover 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 prover computes auxiliary commitments $C_w = g^w\hat{g}^{\rho_w}$ and $C_{\rho_S} = g^{\rho_S}\hat{g}^{\rho'}$. Then, the prover sets $\rho_\alpha = \rho_w\rho_A$, $\rho_\zeta = \rho'\rho_a$, and $\rho_\xi = \rho'\rho_T$.
4. The prover 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 prover 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.

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

$$e(C_A, Y)e(C_a(\prod_{1 \le j \le L, M_j \in \text{TSA}} h_j^{M_j})g, g)^{-1} = (\prod_{1 \le j \le L, M_j \notin \text{TSA}} e(h_j, g)^{M_j})$$
$$\cdot e(h_{L+1}, g)^x e(g_0, g)^r e(\hat{g}, Y)^{\rho_A} e(\hat{g}, g)^\alpha e(C_A, g)^{-w} e(\hat{g}, g)^{-\rho_a}, \tag{2}$$

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

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

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

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

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

Proving OR Relation. For TFA $= \{a_1, \ldots, a_k\}$, the prover 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 prover and the verifier prepare another accumulator $acc' = \prod_{a_j \in \text{TFA}} g_{n+1-a_j}$. This protocol is obtained by modifying that of the AND relation, as follows.

1. Similarly, the prover computes $W = \prod_{a \in \text{FA}}^{a \neq \tilde{a}} g_{n+1-a+\tilde{a}}$ for acc. Furthermore, the prover computes the new witness $W' = \prod_{a_j \in \text{TFA}}^{a_j \neq \tilde{a}} g_{n+1-a_j+\tilde{a}}$ for acc'.
2. In addition to step 2 in AND relation, the prover selects $\rho_g, \rho_{W'}, \rho_{\tilde{S}}, \rho_{\tilde{T}}, \rho_{\tilde{U}}, \rho_{\tilde{F}} \in_R \mathbb{Z}_p^*$, and compute the new commitment $C_g = g_{\tilde{a}}\hat{g}^{\rho_g}$, $C_{W'} = W'\hat{g}^{\rho_{W'}}$, $C_{\tilde{S}} = \tilde{S}_{\tilde{a}}\hat{g}^{\rho_{\tilde{S}}}$, $C_{\tilde{T}} = \tilde{T}_{\tilde{a}}\hat{g}^{\rho_{\tilde{T}}}$, $C_{\tilde{U}} = \tilde{U}_{\tilde{a}}\hat{g}^{\rho_{\tilde{U}}}$, and $C_{\tilde{F}} = \tilde{F}_{\tilde{a}}\hat{g}^{\rho_{\tilde{F}}}$.
3. In addition to step 3 in AND relation, the prover selects $\tilde{\rho}, \tilde{\rho'} \in_R Z_p^*$, sets $\delta = \rho_g\rho_a$, $\tilde{\zeta} = \rho_{\tilde{S}}\rho_g$ and $\tilde{\xi} = \rho_{\tilde{S}}\rho_{\tilde{T}}$. The prover computes auxiliary commitments $C_{\rho_g} = g^{\rho_g}\hat{g}^{\tilde{\rho}}$ and $C_{\rho_{\tilde{S}}} = g^{\rho_{\tilde{S}}}\hat{g}^{\tilde{\rho'}}$. Then, the prover 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 prover sends the commitments $(C_A, C_S, C_T, C_U, C_F, C_g, C_a, C_W, C_{W'}, C_{\tilde{S}}, C_{\tilde{T}}, C_{\tilde{U}}, C_{\tilde{F}}, C_w, C_{\rho_S}, C_{\rho_g}, C_{\rho_{\tilde{S}}})$ to the verifier.

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

$$C_{\rho_g} = g^{\rho_g}\hat{g}^{\tilde{\rho}}, 1 = {C_{\rho_g}}^{\rho_a} g^{-\delta}\hat{g}^{-\rho_\delta}, \tag{8}$$
$$e(C_g, C_a)e(g, C_W)^{-1}z^{-1} = e(C_g, \hat{g})^{\rho_a} e(\hat{g}, C_a)^{\rho_g} e(\hat{g}, \hat{g})^{-\delta} e(g, \hat{g})^{-\rho_W}, \tag{9}$$

and the following equations are added:

$$C_{\rho_{\tilde{S}}} = g^{\rho_{\tilde{S}}}\hat{g}^{\tilde{\rho'}}, 1 = C_{\rho_{\tilde{S}}}^{\rho_g} g^{-\tilde{\zeta}}\hat{g}^{-\rho_{\tilde{\zeta}}}, 1 = C_{\rho_{\tilde{S}}}^{\rho_{\tilde{T}}} g^{-\tilde{\xi}}\hat{g}^{-\rho_{\tilde{\xi}}}, \tag{10}$$
$$e(C_{\tilde{S}}, \tilde{Y'}C_g C_{\tilde{T}})e(g, g)^{-1} = e(\hat{g}, \tilde{Y'}C_g C_{\tilde{T}})^{\rho_{\tilde{S}}} e(C_{\tilde{S}}, \hat{g})^{\rho_g+\rho_{\tilde{T}}} e(\hat{g}, \hat{g})^{-\tilde{\zeta}-\tilde{\xi}}, \tag{11}$$
$$e(\tilde{g}, C_{\tilde{T}})e(C_{\tilde{U}}, \hat{Y'})^{-1} = e(\tilde{g}, \hat{g})^{\rho_{\tilde{T}}} e(\hat{g}, \hat{Y'})^{-\rho_{\tilde{U}}}, \tag{12}$$
$$e(\tilde{g}, C_g)e(C_{\tilde{F}}, g)^{-1} = e(\tilde{g}, \hat{g})^{\rho_g} e(\hat{g}, g)^{-\rho_{\tilde{F}}}, \tag{13}$$
$$e(C_g, acc')e(g, C_{W'})^{-1}z^{-1} = e(\hat{g}, acc')^{\rho_g} e(g, \hat{g})^{-\rho_{W'}}. \tag{14}$$

4 Security

Here, we show the proposed protocols are the PKs 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 the equations. Using the equations (1), we obtain $1 = (g^w\hat{g}^{\rho_w})^{\rho_A} g^{-\alpha}\hat{g}^{-\rho_\alpha}$, and thus $1 = g^{w\rho_A-\alpha}\hat{g}^{\rho_w\rho_A-\rho_\alpha}$. Since the discrete log of $\hat{g}$ to base g is unknown under the DL assumption (due to q-SDH assumption), this means $\alpha = w\rho_A$. By substituting this to equation (2), we have

$$e(C_A, Y)e(C_a(\prod_{1\le j\le L, M_j\in \mathrm{TSA}} h_j^{M_j})g, g)^{-1} = (\prod_{1\le j\le L, M_j\notin \mathrm{TSA}} e(h_j, g)^{M_j})e(h_{L+1}, g)^x$$
$$\cdot e(g_0, g)^r e(\hat{g}, Y)^{\rho_A} e(\hat{g}, g)^{w\rho_A} e(C_A, g)^{-w} e(\hat{g}, g)^{-\rho_a}$$
$$e(C_A, Y)e(\hat{g}^{-\rho_A}, Y)e(\hat{g}^{-\rho_A}, g^w)e(C_A, g^w) = e(C_a(\prod_{1\le j\le L} h_j^{M_j})g, g)e({h_{L+1}}^x, g)$$
$$\cdot e(g_0^r, g)e(\hat{g}^{-\rho_a}, g)$$
$$e(C_A\hat{g}^{-\rho_A}, Yg^w) = e(C_a\hat{g}^{-\rho_a}(\prod_{1\le j\le L} h_j^{M_j}){h_{L+1}}^x g_0^r g, g)$$

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

$$e(A, Yg^w) = e(acc(\prod_{1\le j\le L} h_j^{M_j}){h_{L+1}}^x g_0^r g, g).$$

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

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

Thus, for the extracted $acc = C_a \hat{g}^{-\rho_a}$, we can extract $S = C_S \hat{g}^{-\rho_S}$ and $T = C_T \hat{g}^{-\rho_T}$ s.t. $e(S, \tilde{Y} \cdot acc \cdot T) = e(g,g)$. Similarly, using equations (5), (6), we obtain $U = C_F \hat{g}^{-\rho_F}$ and $F = C_F \hat{g}^{-\rho_F}$ s.t. $e(\tilde{g}, T) = e(U, \hat{Y})$ and $e(\tilde{g}, acc) = e(F, g)$. 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 equation (7), we can similarly extract $W = C_W \hat{g}^{-\rho_W}$ s.t. $e(D, acc) e(g, W)^{-1} = z^k$ for $D = \prod_{1 \le j \le k} g_{a_j}$. 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 equations (10) – (13), we can extract the F-secure BB signature $(\tilde{S}, \tilde{T}, \tilde{U})$ on $R = C_g \hat{g}^{-\rho_g}$ and $\tilde{F}$ s.t. $e(\tilde{S}, \tilde{Y'} R \tilde{T}) = e(g,g)$, $e(\tilde{g}, \tilde{T}) = e(\tilde{U}, \hat{Y'})$ and $e(\tilde{g}, R) = e(\tilde{F}, g)$. Since F-secure BB signatures w.r.t. the public key $\tilde{Y'}, \hat{Y'}$ is issued on only g_j's, it means $R \in \{g_1, \ldots, g_n\}$ (otherwise, the signature is forgeable), and we can set $R = g_{\tilde{a}}$.

Using equations (8), we can obtain $\delta = \rho_a \rho_g$. By substituting this into equation (9), we can extract $W = C_W \hat{g}^{-\rho_W}$ s.t. $e(g_{\tilde{a}}, acc) e(g, W)^{-1} = z$ for the extracted $g_{\tilde{a}}$. This means that attribute $\tilde{a}$ is accumulated into acc. Using equation (14), we can extract $W' = C_{W'} \hat{g}^{-\rho_{W'}}$ s.t. $e(g_{\tilde{a}}, acc') e(g, W')^{-1} = z$ for $g_{\tilde{a}}$. 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 Efficiency

We compare the efficiency between our system and the conventional pairing-based system using the BBS+ signatures. Similarly to the conventional RSA-based systems described in [10], we can construct the conventional PKs for AND and OR relations, which are described in Appendix A.

We introduce the following parameters.

Relation	Conventional system	Our system
AND	$O(L + \tilde{L})$	$O(L)$
OR	$O(L + \tilde{L} + k)$	$O(L)$

Table 1. Asymptotic computational complexity of proof.

Relation	Conventional system	Our system
AND	$(L + \tilde{L} + 5)E(\mathcal{T}) + 8E(\mathcal{G})$	$(L + 15)E(\mathcal{T}) + 24E(\mathcal{G})$
OR	$(L + \tilde{L} + 5)E(\mathcal{T}) + (5k + 8)E(\mathcal{G})$	$(L + 26)E(\mathcal{T}) + 47E(\mathcal{G})$

Table 2. Concrete number of exponentiations in proof generation ($E(\mathcal{T})$: exponentiations on $\mathcal{T}$, $E(\mathcal{G})$: exponentiations on $\mathcal{G}$).

- 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.

In the following comparisons, we consider the computational complexity based on the number of exponentiations and pairings. Namely, we ignore the number of multiplications, since the cost is much smaller than the others' costs.

Table 1 shows the comparison of asymptotic computational complexity for the proof generation and verification. In the both cases of AND and OR relations, we can see that the complexity in finite-set attributes becomes constant. This is because our scheme uses the accumulator verification with constant complexity. The demerit of our system is the length of public key. Our system needs $O(n+L)$ size, while the conventional system needs $O(\tilde{L}+L)$, where n is much larger than $\tilde{L}$.

Next, compare the concrete computational costs. We suppose that mobile devices such as smartphones manage the proof generation, and thus we concentrate in the computation complexity of the proof generation. Table 2 shows the comparison of the concrete costs. Using the pre-computation of pairings, we can omit any pairing computation with adding some slight exponentiations. In this table, we shows the number of the exponentiations needed for the proof generation after the omission. Note that the exponentiation cost on $\mathcal{T}$ is larger than that on $\mathcal{G}$. The results of this table mean that our system has constant but extra costs. Using an example of eID as in [10], we demonstrate that our scheme is effective in spite of the extra costs. Table 3 shows the example of attributes in eID. Generally, the number of string attribute types, L, is much less than the number of finite-set attribute types, $\tilde{L}$. 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, from Table2, in the general case that $\tilde{L}$ amounts to about 30–40 and $L \leq 5$, proving AND relation in our system has more efficiency.

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, ...
	...	

Table 3. Example of string and finite-set attributes.

In case of OR relation, since the efficiency of the conventional system is influenced by parameter k, our system is more efficient. In [10], an example of OR relation is shown:

$$minority \in \{blind, deaf, ...\} \vee social\_benefit \in \{unemployed, social\_benefit\}$$

$$profession \in \{student, teacher, civil\_servant\} \vee type = kids\_card$$

This example considers that countries grant subsidies for access to cultural institutions to particular groups such children, students, handicapped persons, etc. In this case, $k = 10$ in addition to $L \geq 5$ and $\tilde{L} = 40$, and then our system is more efficient than the conventional one.

Finally, we discuss the concrete values of the public key size. We assume that an element of $\mathcal{G}$ is represented by 256 bits to obtain 256-bit ECC security. We set $L + \tilde{L} = 50$ and $n = 1,000$ to $n = 10,000$. In the conventional system, the public key size is less than 2KBytes. In our system, it becomes about 200KBytes to 2MBytes. In the current mobile environments, the data size is sufficiently practical, since the public key is not changed after it is distributed.

6 Conclusion

In this paper, for a pairing-based anonymous credential system, we have showed how to prove AND and OR relations on his attributes with constant complexity in the number of finite-set attributes. 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 evaluation based on the implementation, and the application to authentications in the mobile environments.

Acknowledgments

This work was supported by Grant-in-Aid for Scientific Research (21300004) from Japan Society for the Promotion of Science (JSPS). We would like to thank the anonymous reviewers.

References

1. M.H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," Security in Communication Networks: 5th International Conference, SCN 2006, LNCS 4116, pp.111–125, Springer–Verlag, 2006.
2. M.H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA." Cryptology ePrint Archive: Report 2008/136, 2008. This is the extended version of [1].
3. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya, "Non-interactive anonymous credentials." Cryptology ePrint Archive: Report 2007/384, 2007.
4. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya, "P-signatures and non-interactive anonymous credentials," Proc. 5th Theory of Cryptography Conference (TCC 2008), LNCS 4948, pp.356–374, Springer–Verlag, 2008.
5. P. Bichsel, J. Camenisch, T. Groß, and V. Shoup, "Anonymous credentials on a standard java card," Proc. ACM Conference on Computer and Communications Security 2009 (ACM-CCS'09), pp.600–610, 2009.
6. D. Boneh and X. Boyen, "Short signatures without random oracles," Advances in Cryptology — EUROCRYPT 2004, LNCS 3027, pp.56–73, Springer–Verlag, 2004.
7. D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," Advances in Cryptology — CRYPTO 2004, LNCS 3152, pp.41–55, Springer–Verlag, 2004.
8. D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," Proc. 11th ACM Conference on Computer and Communications Security (ACM-CCS '04), pp.168–177, 2004.
9. X. Boyen and B. Waters, "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. J. Camenisch and T. Groß, "Efficient attributes for anonymous credentials," Proc. ACM Conference on Computer and Communications Security 2008 (ACM-CCS'08), pp.345–356, 2008.
11. J. Camenisch, A. Kiayias, and M. Yung, "On the portability of generalized schnorr proofs," Advances in Cryptology - EUROCRYPT 2009, LNCS 5479, pp.425–442, Springer–Verlag, 2009.
12. J. Camenisch, M. Kohlweiss, and C. Soriente, "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. J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," Advances in Cryptology — CRYPTO 2002, LNCS 2442, pp.61–76, Springer–Verlag, 2002.
14. R. Cramer, Damgård, and B. Schoenmakers, "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. J. Furukawa and H. Imai, "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.

A Proving AND and OR Relations in Conventional System

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

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

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

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.

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

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 $C_j = g^{M_j}\hat{g}^{\rho_j}$ 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.

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

Additionally, prove

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

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