SCIS 2007 The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

An Efficient Anonymous Credential System

Norio Akagi * Yoshifumi Manabe † Tatsuaki Okamoto ‡

Abstract— An anonymous credential is one of the most important notions to counter some of the privacy problems with identity certificates. This paper propose an efficient anonymous credential system that is provably secure in the standard model(i.e., without random oracle model). Our system consists of two parts: a signature scheme and proving knowledge of the signature. A user gets a proof of identity from the signer by using the signature scheme from bilinear maps, and proves knowledge of the signature to a verifier by using a three-move interactive identification scheme. We first present the schemes we used. We then present the anonymous credential system, which is more efficient than existing systems.

Keywords: Anonymous Credentials, Digital Signatures, Identification, Bilinear Maps

1 Introduction

1.1 Background

The concept of anonymous credential systems was introduced by Chaum[4], and after that, many researchers proposed anonymous credential systems in order to counter some of the privacy difficulties related to identity certificates, and to idealize the implementation of physical credentials, like entry certification, driver's licenses, and so on.

The basic properties of anonymous credential systems are as follows: It should be impossible for a user to forge a credential for the user, even if users and other organizations team up and launch an adaptive attack on the organization. It should also be impossible for an organization to find out anything about the user, apart from the fact that the user has ownership of some set of credentials, even if it cooperates with other organizations. In particular, two pseudonyms belonging to the same user are unlinkable. Finally, the system is expected to be efficient. To know more about the history and motivation behind anonymous credentials, Chapter 3 of Lysyanskaya's Ph.D thesis[7] is a very-well written exposition.

Existing anonymous credential schemes are based upon the Strong RSA assumption, or the LRSW[8] assumption. For example, Camenisch and Lysyanskaya proposed an anonymous credential scheme[3] that is secure under the LRSW assumption for groups with bilinear maps. But in the case of LRSW assumption, it seems to be identical to the signature scheme proposed in [3].

* Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshidahonmachi, Sakyo-ku, Kyoto-shi, Japan (akagi@ai.soc.i.kyoto-u.ac.jp)

† NTT Cyber Space Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Japan / Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshidahonmachi, Sakyo-ku, Kyoto-shi, Japan (manabe.yoshifumi@lab.ntt.co.jp)

‡ NTT Information Sharing Platform Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Japan / Department of Social Informatics, Graduate School of Informatics, Kyoto University, Yoshidahonmachi, Sakyo-ku, Kyoto-shi, Japan(okamoto.tatsuaki@lab.ntt.co.jp)

1.2 Our results

In this paper, we construct an anonymous credential system that is based on the blind signature scheme proposed in [10], and on the three-move identification scheme proposed in [9] . The signature scheme is secure under q-SDH assumptions, and is used in various other schemes. Then the identification scheme, which is used to prove knowledge of a signature, is also secure under non-interactive assumptions. Our anonymous credential system is more efficient than that based on the LRSW assumption.

2 Preliminaries

2.1 Bilinear Groups

This paper follows the notation regarding bilinear groups given in [1, 2]. Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as follows:

1. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic groups of prime order p, where possibly $\mathbb{G}_1 = \mathbb{G}_2$,
2. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$,
3. ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2)$
4. e is a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$, i.e.,
   - (Bilinear): for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, for all $a, b \in \mathbb{Z}$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$
   - (Non-degenerate): $e(g_1, g_2) \neq 1$ (i.e., $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$),
   - (Efficient): e, ψ and the group in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ can be computed efficiently.

2.2 Anonymous Credential System

2.2.1 Definition of Anonymous Credential System

In this section, we define anonymous credential systems. An anonymous credential system consists of parties which

are users, authorities, and verifiers. Any anonymous credential system can perform the following operations.

Key Generation: Authority $\mathcal{A}$, given security parameter 1^k, outputs a pair of public-key and secret-key, (pk, sk). The public keys are then published by $\mathcal{A}$.

Credential Issuing Protocol: User $\mathcal{U}$ has some kind of data m that $\mathcal{U}$ wants to obtain a certificate for. Such properties include "belongs to Kyoto University", "is over the age of 20." or rights such as "can access the secure room". How $\mathcal{S}$ detects whether m is valid or not with regard to $\mathcal{U}$ is outside of this protocol.

Now $\mathcal{U}$ executes the credential issuing protocol for m with $\mathcal{A}$ by using user's input m and authority's secret-key. At the end of the protocol, $\mathcal{U}$ obtains the credential σ, corresponding to m.

Credential Verification Protocol: After $\mathcal{U}$ obtains the credential of m, $\mathcal{U}$ executes the credential verification protocol of m with verifier $\mathcal{V}$. At the end of the protocol, $\mathcal{V}$ outputs `accept` if the verification equations holds, and otherwise outputs `reject`.

2.2.2 Security of Anonymous Credential System

In this section, we refer to the definition of the security of anonymous credential systems. The security of anonymous credential systems are defined as follows.

Unforgeable: User $\mathcal{U}$ cannot forge a valid credential σ on any value, if σ is not issued by $\mathcal{A}$. We show a formal definition below. There exists an adversary `Adv`, which has no information about the Authority's secret-key. `Adv` can execute credential issuing protocl with $\mathcal{A}$ polynomial number of times, and get credentials of adaptively chosen messages. Then `Adv` and $\mathcal{V}$ execute the credential verification protocol of m, which has not been chosen by `Adv` yet. If the probability that $\mathcal{V}$ outputs `accept` at the end of the protocol is negligible, the anonymous credential system is unforgeable.

Anonymity and Unlinkability: Anonymous credential systems should have the property of anonymity and unlinkability. We merge these two properties into one definition of security. The definition is as follows.

There is adversary `Adv` that plays the roles of authority and verifier. Let us introduce the following game among `Adv` and two honest users $\mathcal{U}_0$ and $\mathcal{U}_1$.

1. `Adv` outputs public-key, and a message m.
2. `Adv` engages in the credential issuing protocol of m with two users, $\mathcal{U}_0$ and $\mathcal{U}_1$. The two users employ the same data to obtain credentials.
3. (a) `Adv` engages in the credential verification protocol with $\mathcal{U}_0$ and $\mathcal{U}_1$. `Adv` can execute this protocol polynomial number of times.
   (b) $d \in \{0,1\}$ is chosen randomly. $\mathcal{U}_d$ and `Adv` execute the credential verification protocol. `Adv` also can execute this polynomial number of times. Then, Adv can execute 3(a) again.
   (c) $\mathcal{A}$ outputs $d' \in \{0,1\}$, which is supposed to be the adversary's guess of value d.

If the probability that $d' = d$ is $1/2 + \epsilon$, then the adversary's advantage is defined to be ϵ. The anonymous credential system is said to be anonymous and unlikable if the advantage of any polynomial-time adversary is negligible.

2.3 Definition of Secure Signature Schemes

In this section we recall the standard notion of security, existential unforgeability against chosen message attacks[1] as well as a slightly stronger notion of security for a signature scheme: strong existential unforgeability against chosen message attacks[6]. To define existential unforgeability, we introduce the folloing game among adversary $\mathcal{A}$ and honest signer $\mathcal{S}$.

1. Key setup:
   Run key generation algorithm $\mathcal{G}(1^n)$ to obtain a pair of public-key and secret-key, (pk, sk). pk is given to adversary $\mathcal{A}$, and (pk, sk) is given to signer $\mathcal{S}$.
2. Queries to signing oracle:
   $\mathcal{A}$ adaptively requests $\mathcal{S}$ (or signing oracle) to sign on at most qs messages of his choice $m_1, ..., m_{qs}$, $\mathcal{S}$ responds to m_i with a signature $\sigma_i = \mathcal{S}(sk, m_i)$.
3. Output:
   Eventually, $\mathcal{A}$ outputs pair (m, σ). $\mathcal{A}$ wins the game if
   (a) m is not any of m_i $(i = 1, ..., qs)$
   (b) $\mathcal{V}$(pk, m, σ)=`accept`.
   We define $\mathtt{Adv}_{Sig}^{unforge}$ to be the probability that $\mathcal{A}$ wins the above game, taken over the coin tosses made by $\mathcal{A}$, $\mathcal{G}$, and $\mathcal{S}$.

Definition 1. (Existential Unforgeability) Adversary $\mathcal{A}$ (t, qs, ϵ)-forges a signature scheme if $\mathcal{A}$ runs in time at most t. $\mathcal{A}$ makes at most qs queries to $\mathcal{S}$, and $\mathtt{Adv}_{Sig}^{unforge}$ is at least ϵ. A signature scheme is (t, qs, ϵ)-existentially-unforgeable under adaptive chosen message attacks if no adversary $\mathcal{A}$ (t, qs, ϵ)-forges the scheme.

Remark: (Strong Existential unforgeability) If the condition in Step 3a in the above game is changed to " (m, σ) is not any of (m_i, σ_i)" (instead of "m is not any of m_i") $(i = 1, ..., qs)$, we obtain a stronger notion of unforgeability. If a scheme satisfies the above definition of unforgeability under this stronger notion, we say that it is (t, qs, ϵ)-strongly-existentially unforgeable under adaptive chosen message attacks.

2.4 Definition of Secure Identification Schemes

2.4.1 Identification

Identification schemes are defined in [9] as follows.

Definition 2. An identification scheme consists of two stages:

1. Initialization: In this stage, each user (e.g., $\mathcal{A}$) generates a secret key (e.g., $SK_{\mathcal{A}}$) and a public key (e.g., $PK_{\mathcal{A}}$) by using probabilistic polynomial-time generation algorithm G on input of the key size. A link between each user and its public key is commonly share

established. Note that in some schemes a part of the public key can be commonly shared among all users as a system parameter.

2. Operation: In this stage, any user (e.g., $\mathcal{A}$) can demonstrate its identity to a verifier by performing some identification protocol related to its public key (e.g., $PK_{\mathcal{A}}$), where the input for the verifier is the public key (e.g., $PK_{\mathcal{A}}$). At the conclusion of this stage, the verifier either outputs "accept" or "reject".

2.4.2 Security of Identification schemes

A security of identification scheme is defined in [9].

Definition 3. A prover $\mathcal{A}$ (resp. verifier $\mathcal{B}$) is a "good" prover denoted by $\bar{\mathcal{A}}$ (resp. "good" verifier denoted by $\bar{\mathcal{B}}$), if it does not deviate from the protocols dictated by the scheme. Let $\tilde{\mathcal{A}}$ be a fraudulent prover who does not complete the initialization stage of Definition as $PK_{\mathcal{A}}$ and may deviate from the protocols (so another person/machine can simulate $\tilde{\mathcal{A}}$). $\tilde{\mathcal{B}}$ is a not-good $\mathcal{B}$. $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are assumed to be polynomial time bounded machines, that may be nonuniform.

An identification scheme $(\mathcal{A}, \mathcal{B})$ is secure if

1. $\left(\bar{\mathcal{A}}, \bar{\mathcal{B}}\right)$ succeeds with overwhelming probability.

2. There is no coalition of $\tilde{\mathcal{A}}$, $\tilde{\mathcal{B}}$ with the property that, after a polynomial number of executions of $\left(\bar{\mathcal{A}}, \tilde{\mathcal{B}}\right)$ and relaying a transcript of the communication to $\bar{\mathcal{A}}$, it is possible to execute $\left(\tilde{\mathcal{A}}, \bar{\mathcal{B}}\right)$ with nonnegligible probability of success. The probability is taken over the distribution of the public key and the secret key as well as the coin tosses of $\bar{\mathcal{A}}, \tilde{\mathcal{B}}, \tilde{\mathcal{A}}$, and $\bar{\mathcal{B}}$, up to the time of the attempted impersonation.

3 Assumptions

3.1 Strong Diffie-Hellman (SDH) Assumption

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups (introduced in Section 2.1). The problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given the $(q+2)$-tuple $\left(g_1, g_2, g_2^x, ..., g_2^{x^q}\right)$ as input, output pair $\left(g_1^{\frac{1}{x+c}}, c\right)$ where $c \in \mathbb{Z}_p^*$. Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{SDH}(q)$, in solving q-SDH in $(\mathbb{G}_1, \mathbb{G}_2)$ if

$$\mathtt{Adv}_{SDH}(q) \leftarrow Pr\left[\mathcal{A}\left(g_1, g_2, g_2^x, ..., g_2^{x^q}\right) = \left(g_1^{\frac{1}{x+c}}, c\right)\right],$$

where the probability is taken over the random choices of $g_2 \in \mathbb{G}_2$, $x, y \in \mathbb{Z}_p^*$, and the coin tosses of $\mathcal{A}$.

Definition 4. Adversary $\mathcal{A}$ (t, ϵ)-breaks the q-SDH problem if $\mathcal{A}$ runs in time at most t and $\mathtt{Adv}_{SDH}(q)$ is at least ϵ. The (q, t, ϵ)-SDH assumption holds if no adversary $\mathcal{A}$ (t, ϵ)-breaks the q-SDH problem.

3.2 The Signature Scheme

We now present the signature scheme used in our anonymous credential system. The scheme below was presented by Okamoto[10].

Okamoto Signature Scheme

Key Generation:
Randomly select generators $g_2, u_2, v_2 \in \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, and $v_1 \leftarrow \psi(v_2)$. Randomly select $x \in \mathbb{Z}_p^*$ and compute $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. The public and secret keys are:

Public key: g_1, g_2, w_2, u_2, v_2
Secret key: x

Signature Generation:
Let $m \in \mathbb{Z}_p^*$ be the message to be signed. Signer $\mathcal{S}$ randomly selects r and s from $\mathbb{Z}_p^*$, and computes

$$\sigma \leftarrow \left(g_1^m u_1 v_1^s\right)^{1/(x+r)}.$$

Here $1/(x+r) \bmod p$ (and $m/(x+r) \bmod p$ and $s/(x+r) \bmod p$) are computed. In the unlikely event that $x + r \equiv 0 \bmod p$, we try again with a different random r. (σ, r, s) is the signature of m.

Signature verification:
Given public-key $(g_1, g_2, w_2, u_2, v_2)$, message m, and signature (σ, r, s), check that m, r, $s \in \mathbb{Z}_p^*$, $\sigma \in \mathbb{G}_1$, $\sigma \neq 1$, and

$$e(\sigma, w_2 g_2^r) = e\left(g_1, g_2^m u_2 v_2^s\right)$$

If they hold, the verification result is `valid`, otherwise `invalid`.

3.3 A Three-Move Identification Scheme

In our anonymous credential system, a three-move identification scheme is essential so as to prove knowledge of credentials. Okamoto proposed a three-move identification scheme[9] that is almost as efficient as the Schnorr identification scheme[5], and proved that it is as secure as the discrete logarithm problem. We use this three-move identification scheme to prove a knowledge of signature. We later describe how this identification scheme is used in our system.

4 Proposed Anonymous Credential System

In this section, we show our anonymous credential system.

4.1 Key Generation

First, authority $\mathcal{A}$ generates public-key $(g_1, g_2, w_2, u_2, v_2)$, and secret-key x in the same way as the signature scheme in Section 3.2.

4.2 Credential Issuing Protocol

First, user $\mathcal{U}$ sends data m as a message, for which $\mathcal{U}$ wants to obtain a certificate, to authority $\mathcal{A}$.

$$\mathcal{U} \xrightarrow{m} \mathcal{A}$$

When message m is received from $\mathcal{U}$, $\mathcal{A}$ signs to m by using the signature scheme described in Section 3.2. $\mathcal{A}$ then

sends triple signature (σ, r, s), to $\mathcal{U}$, where $\sigma = \left(g_1^m u_1 v_1^s\right)^{1/(x+r)}$.

$$\mathcal{U} \xleftarrow{(\sigma,r,s)} \mathcal{A}$$

$\mathcal{U}$ gets the credential corresponding to m as a signature.

4.3 Credential Verification Protocol

After getting its credential, $\mathcal{U}$ proves knowledge of the credential to verifier $\mathcal{V}$, instead of sending credential directly to $\mathcal{V}$.

Proof of Knowledge of a Credential on a Message m

Step 1: Prover $\mathcal{U}$ randomly selects t, θ from $\mathbb{Z}_p^*$, and computes

$$\sigma' \leftarrow \sigma^{t/\theta} = \left(g_1^m u_1 v_1^s\right)^{t/\theta(x+r)},$$
$$\alpha \leftarrow (w_2 g_2^r)^\theta,$$
$$\beta \leftarrow \left(g_2^m u_2 v_2^s\right)^t.$$

and sends (σ', α, β) to the verifier $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{(\sigma',\alpha,\beta)} \mathcal{A}$$

Step 2: Verifier $\mathcal{V}$ checks the equation below

$$e(\sigma', \alpha) = e(g_1, \beta)$$

Step 3: $\mathcal{U}$ proves knowledge for the following statement:

$$PK\{(\theta, r\theta) : \alpha = w_2^\theta g_2^{r\theta}\}$$

This proof of knowledge consists of the following two proofs of knowledge.

(1)$\mathcal{U}$ randomly selects t_1, t_2, t_3 from $\mathbb{Z}_p^*$, and computes

$$\gamma \leftarrow \alpha^{t_1} g_2^{t_2} u_2^{t_3},$$

and sends γ to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{\gamma} \mathcal{V}$$

$\mathcal{U}$ then proves knowledge to $\mathcal{V}$ for the following statement:

$$PK\{(t_1, t_2, t_3) : \gamma = \alpha^{t_1} g_2^{t_2} u_2^{t_3}\}.$$

(2)$\mathcal{U}$ computes

$$\delta \leftarrow \theta t_1,$$

and sends δ to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{\delta} \mathcal{V}$$

$\mathcal{U}$ then proves knowledge to $\mathcal{V}$ for the following statement:

$$PK\{(\omega, t_3) : \gamma / w_2^\delta = g_2^\omega u_2^{t_3} \ (\because \omega = r\theta t_1 + t_2)\}.$$

We detail these two protocols later.

Step 4: Prover $\mathcal{U}$ sends m to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{m} \mathcal{V}$$

$\mathcal{U}$ and $\mathcal{V}$ then executes a proof of knowledge protocol for the following statement:

$$PK\{(t, st) : \beta = \left(g_2^m\right)^t u_2^t v_2^{st}\}$$

We describe details of this protocol later.

Now we show how the protocols in (1), (2) of **Step 3** and in **Step 4** work. The three-move identification schemes, proposed in [9], are used in these protocols.

Three Proofs of Knowledge with Identification Schemes

① $PK\{(t_1, t_2, t_3) : \gamma = \alpha^{t_1} g_2^{t_2} u_2^{t_3}\}$

Common input: Public key and (α, γ)
Prover's input: (t_1, t_2, t_3)

Protocol:
Step1: $\mathcal{U}$ picks random numbers r_1, r_2, $r_3 \in \mathbb{Z}_p^*$, computes $A = \alpha^{r_1} g_2^{r_2} u_2^{r_3}$, and sends A to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{A} \mathcal{V}$$

Step2: $\mathcal{V}$ sends a random number $b \in \mathbb{Z}_p^*$ to $\mathcal{U}$.

$$\mathcal{U} \xleftarrow{b} \mathcal{V}$$

Step3: $\mathcal{U}$ sends (c_1, c_2, c_3) to $\mathcal{V}$ such that

$$c_1 = r_1 + bt_1 \bmod p,$$
$$c_2 = r_2 + bt_2 \bmod p,$$
$$c_3 = r_3 + bt_3 \bmod p$$

$$\mathcal{U} \xrightarrow{c_1,c_2,c_3} \mathcal{V}$$

Step4: $\mathcal{V}$ checks that

$$\alpha^{c_1} g_2^{c_2} u_2^{c_3} = A\gamma^b$$

If it holds, $\mathcal{V}$ outputs `accept`, otherwise `reject`.

② $PK\{(\omega, t_3) : \gamma / w_2^\delta = g_2^\omega u_2^{t_3}\}$

Common input: Public key and (γ, δ)
Prover's input: (ω, t_3)

Protocol:
Step1: $\mathcal{U}$ picks random numbers r_1, $r_2 \in \mathbb{Z}_p^*$, computes $A = g_2^{r_1} u_2^{r_2}$, and sends A to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{A} \mathcal{V}$$

Step2: $\mathcal{V}$ sends a random number $b \in \mathbb{Z}_p^*$ to $\mathcal{U}$.

$$\mathcal{U} \xleftarrow{b} \mathcal{V}$$

Step3: $\mathcal{U}$ sends (c_1, c_2) to $\mathcal{V}$ such that

$$c_1 = r_1 + b\omega \bmod p,$$
$$c_2 = r_2 + bt_3 \bmod p,$$

$$\mathcal{U} \xrightarrow{c_1, c_2} \mathcal{V}$$

Step4: $\mathcal{V}$ checks that

$$g_2^{c_1} u_2^{c_2} = A\left(\frac{\gamma}{{w_2}^{\delta}}\right)^b$$

If it holds, $\mathcal{V}$ outputs `accept`, otherwise `reject`.

③ $PK\{(t, st) : \beta = \left(g_2^m\right)^t u_2^t v_2^{st}\}$

Common input: Public key and (β, m)
Prover's input: (t, st)

Protocol:
Step1: $\mathcal{U}$ picks random numbers r_1, $r_2 \in \mathbb{Z}_p^*$, computes $A = \left(g_2^m\right)^{r_1} u_2^{r_1} v_2^{r_2}$, and sends A to $\mathcal{V}$.

$$\mathcal{U} \xrightarrow{A} \mathcal{V}$$

Step2: $\mathcal{V}$ sends a random number $b \in \mathbb{Z}_p^*$ to $\mathcal{U}$.

$$\mathcal{U} \xleftarrow{b} \mathcal{V}$$

Step3: $\mathcal{U}$ sends (c_1, c_2) to $\mathcal{V}$ such that

$$c_1 = r_1 + bt \bmod p,$$
$$c_2 = r_2 + bst \bmod p,$$

$$\mathcal{U} \xrightarrow{c_1, c_2} \mathcal{V}$$

Step4: $\mathcal{V}$ checks that

$$\left(g_2^m\right)^{c_1} u_2^{c_1} v_2^{c_2} = A\beta^b$$

If it holds, $\mathcal{V}$ outputs `accept`, otherwise `reject`.

4.4 Security

In this section, we refer to the security of our proposed anonymous credential system.

4.4.1 Unforgeable

First, the security of signature scheme in our credential issuing protocol is described in [10].

Property. If the $(qs + 1, t', \epsilon')$-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the signature scheme is (t, qs, ϵ') -strongly-exsistentially-unforgeable against adaptive chosen message attacks, provided that

$$\epsilon \geq 3qs\epsilon', \text{ and } t \leq t' - \Theta\left(q_s^2 T\right)$$

where T is the maximum time for a single exponentiation in $\mathbb{G}_1$, and $\mathbb{G}_2$.

This theorem allows us to use the signature scheme in Section 3.2 under the q-SDH assumption.

Next, we refer to the unforgeability of the credential verification protocol of our system.

Theorem 1. If the signature scheme in our system is secure under the q-SDH assumption, $\mathcal{U}$ cannot forge credentials for which $\mathcal{V}$ outputs `accept` at the end of the credential verification protocol.

Sketch of the proof. Credential verification protocol in our system consists of three protocols of proving knowledge, as described in Section 4.3. When these protocols are not secure, $\mathcal{U}$ can forge (σ', α, β) that satisfies the verifier's equation. If $\mathcal{U}$ can forge such (σ', α, β) without knowing the original signature (σ, r, s), we can construct an extracter $\mathcal{E}$. $\mathcal{E}$ can use the forger $\mathcal{U}$ as a black-box.

Let us focus on protocol $PK\{(t_1, t_2, t_3) : \gamma = \alpha^{t_1} g_2^{t_2} u_2^{t_3}\}$. First, the protocol is executed once normally. $\mathcal{V}$ sends b and gets

$$c_1 = r_1 + bt_1, c_2 = r_2 + bt_2, c_3 = r_3 + bt_3.$$

After the first execution of the protocol, $\mathcal{E}$ resets $\mathcal{U}$, and the protocol is reexecuted. $\mathcal{V}$ then sends b' and gets

$$c_1' = r_1 + b't_1, c_2' = r_2 + b't_2, c_3' = r_3 + b't_3.$$

$\mathcal{E}$ sees these executions of the protocol, and can calculate

$$t_1 = \frac{c_1' - c_1}{b_1' - b_1}, t_2 = \frac{c_2' - c_2}{b_2' - b_2}, t_3 = \frac{c_3' - c_3}{b_3' - b_3}.$$

Now $\mathcal{E}$ succeeds in extracting (t_1, t_2, t_3).

From $PK\{(\omega, t_3) : \gamma/w_2^{\delta} = g_2^{\omega} u_2^{t_3}\}$ and $PK\{(t, st) : \beta = \left(g_2^m\right)^t u_2^t v_2^{st}\}$, $\mathcal{E}$ can also extract (ω, t_3) and (t, st) in the same way, and can calculate (σ, r, s) from these extracted data. This contradicts the security of the signature scheme in our system.

Thus $\mathcal{U}$ cannot forge a credential (σ', α, β) that satisfies the verifier's check, and our proposed system is unforgeable.

4.4.2 Anonymity and Unlinkability

We now refer to the anonymity and unlinkability of our system. The game described in Section 2.2.2 is used to assess our system. `Adv` outputs public-key $(g_1, g_2, w_2, u_2, v_2)$. $\mathcal{U}_0$ and $\mathcal{U}_1$ send the same data, m, to `Adv`, and `Adv` sends credentials of m (σ_0, r_0, s_0) to $\mathcal{U}_0$, and (σ_1, r_1, s_1) to $\mathcal{U}_1$. $\mathcal{U}_0, \mathcal{U}_1$ and `Adv` then execute the protocols of proving knowledge(Section 4.3) a polynomial number of times. Finally, a random $d \in (0, 1)$ is chosen, and protocols of proving knowledge (Section 4.3) are executed among $\mathcal{U}_d$ and `Adv` a polynomial number of times. At the end of the protocols, `Adv` make a guess about number d.

Theorem. Our system is anonymous and unlinkable.

Sketch of the proof. If the system is anonymous and unlinkable, the protocols of proving knowledge are witness-

indistinguishable; that is, in the game described in Section 2.2.2, the view of Step.3(a) and that of Step.3(b) are information-theoretically independent. Namely, the system is anonymous and unlinkable if Adv, upon receiving $\left(\sigma'_d, \alpha_d, \beta_d\right)$ from $\mathcal{U}_d$ in Step 3(b), cannot decide whether this triple data set is generated from (σ_0, r_0, s_0) or (σ_1, r_1, s_1).

We cannot distinguish the distributions of t_0 from that of t_1, because both of them are randomly chosen from $\mathbb{Z}_p^*$. The distributions of θ_0 and θ_1 are also indistinguishable. Thus Adv cannot distinguish between the distribution of $\left(\sigma'_d, \alpha_d, \beta_d\right)$ generated from (σ_0, r_0, s_0) and the distribution of $\left(\sigma'_d, \alpha_d, \beta_d\right)$ generated from (σ_1, r_1, s_1). That means our system has the properties of anonymity and unlinkablity.

5 Performance Analysis

We turn now to the efficiency of our anonymous credential system. In our system, the size of public keys is 5 elements, and the size of secret keys is 1. The size of a credential is 3 and the size of proving knowledge of credential is 3. The scheme based on LRSW assumptions[3] has 4 public keys, 3 secret keys, 5 credentials, and 5 elements in proving knowledge. We consider here, the number of operations in our scheme. The signature scheme uses 3 exponentials. Next, 2 pairings and 3 exponentials are used to verify the signature, and 4 exponentials, 2 pairings, and 3 proofs of knowledge of values are needed to prove knowledge of a signature. The scheme based on LRSW assumptions[3] has 5 exponentials to generate a signature, 10 pairings and 2 exponentials to verify the signature, and 5 exponentials, 10 pairings, and 4 proofs of knowledge of values to prove knowledge of a signature. Our system is superior to the scheme based on LRSW assumptions, considering efficiency.[3].

6 Conclusion

This paper introduces an anonymous credential system that is more efficient than existing ones and is secure under the q-SDH assumptions, which are non-interactive and widely used in various schemes.

References

[1] Dan Boneh and X. Boyen. Short signatures without random oracles, 2004. Proceedings of EUROCRYPT'04, LNCS, Vol.3027, pp.382–400, Springer 2004.

[2] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. 2001. Proceedings of ASIACRYPT'01, LNCS, Vol.2248, pp.514–532.

[3] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. Crypto 2004, LNCS, Vol.3152, pp.56-72, 2004.

[4] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, Vol. 28, No. 10, pp. 1030–1044, 1985.

[5] C.P.Schnorr. Efficient signature generation by smart cards. In *Journal of cryptology*, Vol. 4, pp. 161–174, 1991.

[6] Shafi Goldwasser, Silvio Micali, and Ron L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, Vol. 17, No. 2, pp. 281–308, 1988.

[7] A. Lysyanskaya. Signature schemes and applications to cryptographic protocol design. Ph.D thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, September 2002.

[8] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography, LNCS, Vol.1758, pp.184–199, 2000.

[9] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Proceedings of CRYPTO'92, LNCS, Vol.740, pp.31–53*, 1993.

[10] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles., 2006. TCC'06, pp.80–99, full version in Cryptology ePrint Archive, http://eprint.org/2006/102.