An Efficient Anonymous Credential System

Norio Akagi[1], Yoshifumi Manabe[1,2], and Tatsuaki Okamoto[1,2]

[1] Department of Social Informatics, Graduate School of Informatics, Kyoto University
akagi@ai.soc.i.kyoto-u.ac.jp
[2] NTT Laboratories, Nippon Telegraph and Telephone Corporation
{manabe.yoshifumi, okamoto.tatsuaki}@lab.ntt.co.jp

Abstract. This paper presents an efficient anonymous credential system that includes two variants. One is a system that lacks a credential revoking protocol, but provides perfect anonymity-unlinkability and computational unforgeability under the strong Diffie-Hellman assumption. It is more efficient than existing credential systems with no revocation. The other is a system that provides revocation as well as computational anonymity-unlinkability and unforgeability under the strong Diffie-Hellman and decision linear Diffie-Hellman assumptions. This system provides two types of revocation simultaneously: one is to blacklist a user who acted wrong so that he can no longer use his credential, and the other is identifying a user who acted wrong from his usage of credential. Both systems are provably secure under the above-mentioned assumptions in the standard model.

1 Introduction

1.1 Background

The concept of anonymous credential systems was introduced by Chaum [1], and many anonymous credential systems since then have been proposed.

The basic properties of any anonymous credential system are as follows: It should be hard for a user to forge a credential. Credentials also should be anonymous and unlinkable, thus, a verifier cannot learn anything about the user when it proves its credential to the verifier. Finally, the system is expected to be efficient. The details of the history and motivation behind anonymous credentials can be found in [2].

One of the most efficient existing anonymous credential systems is the Camenisch-Lysyanskaya system [3] that is secure under the LRSW assumption for groups with bilinear maps [4]. However, this system lacks a credential revoking protocol.

There are roughly two types of revocations in anonymous credential systems. One is to reveal the user's identity if the user misbehaves, and the other enables a verifier to reject blacklisted users when they show their credentials to the verifier.

One of the most efficient existing anonymous credential systems with revocation of revealing the misbehaved user's identity is [5], which is secure under the strong RSA (SRSA) and decisional Diffie-Hellman (DDH) assumptions. The only existing anonymous credential system with revocation of blacklisting users is [6], which is secure under the strong Diffie-Hellman (SDH) and DDH assumptions in the random oracle model.

No efficient anonymous credential system with two types of revocation simultaneously has been proposed.

1.2 Our Result

This paper proposes two variants of a anonymous credential system.

One is an anonymous credential system without revocation (called a "basic anonymous credential system") that is more efficient than the most efficient existing protocol without revocation [3]. It is unforgeable under the SDH assumption, and perfectly (information theoretically) anonymous-and-unlinkable.

The other is the first efficient anonymous credential system that provides two types of revocation (blacklisting and revealing an identity) simultaneously. Our system is unforgeable under the SDH assumption, and anonymous-and-unlinkable under the decision linear Diffie-Hellman assumption (the decision linear assumption).

Both systems are provably secure under the above-mentioned assumptions in the standard model.

2 Preliminaries

2.1 Notation

We will use notation PK as follows: $PK\{(\alpha,\beta) : y = g^{\alpha}h^{\beta}\}$ denotes a "zero-knowledge proof of Knowledge of integers α and β such that $y = g^{\alpha}h^{\beta}$ where y, g, and h are elements of some group $\mathbb{G} = \langle g \rangle = \langle h \rangle$.

2.2 Bilinear Groups

This paper follows the notation regarding bilinear groups given in [**?**,**?**]. 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 $\psi(g_2) = g_1$.
4. e is a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, 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 $a, b \in \mathbb{Z}_{\mathbb{P}}^*$, $e\left(u^a, v^b\right) = e(u,v)^{ab}$
   - (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.

2.3 Anonymous Credential System

In this section, we outline the protocols and the security of anonymous credential systems. We first refer to the basic system, without the credential revoking protocol.

Definition of Basic Anonymous Credential System A basic anonymous credential system consists of three parties users, an authority, and verifiers. An anonymous credential system performs the following operations.

Key Generation: Authority `Auth`, given security parameter 1^k, outputs a pair of public-key and secret-key, (pk, sk).

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

$\mathcal{U}$ executes the credential issuing protocol for m with `Auth` by using $\mathcal{U}$'s input m and `Auth`'s secret-keys. At the end of the protocol, $\mathcal{U}$ obtains a credential `Cred`, corresponding to m.

Credential Proving Protocol: After $\mathcal{U}$ obtains the credential of m, $\mathcal{U}$ executes the credential proving protocol of m with a verifier $\mathcal{V}$, that proves $\mathcal{U}$'s possession of `Cred`. At the end of the protocol, $\mathcal{V}$ outputs `accept` if $\mathcal{U}$ really has a valid `Cred`, otherwise outputs `reject`.

Security of Basic Anonymous Credential System In this section, we refer to the definition of the security of the basic anonymous credential system. The security of the basic anonymous credential system is defined as follows.

Unforgeability: $\mathcal{U}$ cannot forge a valid credential `Cred` on any value unless `Cred` was issued by `Auth`. We show a more formal definition: Let us consider the following game. Let `Adv` be an adversary. `Adv` runs in time at most τ. It first executes the credential issuing protocol with `Auth` at most q_{Auth} times, and obtains valid credentials of adaptively chosen messages. Finally, `Adv` and $\mathcal{V}$ execute the credential proving protocol for message m, which has not been chosen by `Adv` yet, and $\mathcal{V}$ outputs `accept` or `reject`. If the probability that $\mathcal{V}$ outputs `accept` at the end of the protocol is at most ϵ for any `Adv`, the anonymous credential system is $(\tau, q_{Auth}, \epsilon)$-unforgeable.

Anonymity and Unlinkability: An anonymous credential system should provide user privacy. It should be impossible for verifier $\mathcal{V}$ and authority `Auth` to find anything about user $\mathcal{U}$, except the fact that $\mathcal{U}$ has some set of credentials, even if $\mathcal{V}$ cooperates with other verifiers or the authority (this feature is called anonymity). In particular, two credentials belonging to the same user $\mathcal{U}$ cannot be linked by $\mathcal{V}$ and `Auth` (this feature is called unlinkability). We merge these two properties into one definition of security. Anonymous credential systems should have the property of (τ, ϵ)-anonymity-and-unlinkability.

The formal definition is as follows: There is an adversary `Adv` that plays the role of a verifier and an authority. Let us introduce the following game among `Adv` and two honest users $\mathcal{U}_0$ and $\mathcal{U}_1$.

1. `Adv` outputs its public-key (except some system parameters).

2. Adv engages in the credential issuing protocol of m with two users, $\mathcal{U}_0$ and $\mathcal{U}_1$. These two users employ the same data, m, to obtain credentials.
3. (a) Adv engages in the credential proving protocol with $\mathcal{U}_0$ and $\mathcal{U}_1$. Adv can execute this protocol a polynomial number of times.
   (b) $d \in \{0, 1\}$ is chosen randomly. $\mathcal{U}_d$ and Adv execute the credential proving protocol. Adv also can execute this a protocol polynomial number of times. Next, Adv can execute 3(a) again.
   (c) Adv outputs $d' \in \{0, 1\}$, which is supposed to be the Adv'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 (τ, ϵ)-anonymous-and-unlinkable if the advantage of any adversary, whose running time is at most τ, is at most ϵ.

We next refer to an anonymous credential system that has the credential revoking functions.

Definition of Anonymous Credential System with Revocation In this paper, we provide two types of revocation functions, blacklisting and identity revealing. Blacklisting is where Auth creates a blacklist BL of unacceptable users, and $\mathcal{V}$ reads the list and can reject the listed users in the credential proving protocol. In the existing anonymous credential system with this type of revocation [6], $\mathcal{V}$ lists bad users to BL when $\mathcal{V}$ notices that they had done something wrong, by using the transcript which $\mathcal{V}$ obtained in the authentication protocol (corresponding to the credential proving protocol in this paper). In our system, the authority Auth creates BL, by listing users when Auth detects that they did something wrong. $\mathcal{V}$ can read but not write BL.

Identity revealing, where $\mathcal{V}$ can know the identity of some user whose transactions are illegal [5]. In order to achieve this property, an anonymous credential system needs another party, an opener $\mathcal{O}$. $\mathcal{O}$ can reveal the identity of $\mathcal{U}$ for a successful credential proving transaction between $\mathcal{U}$ and $\mathcal{V}$. Auth also has a database DB to record the data used in the credential issuing protocol with users. $\mathcal{O}$ can read but not write DB.

In this system, not only Auth but also $\mathcal{U}$ and $\mathcal{O}$ generate a pair of public-key and secret-key. $\mathcal{U}$ then uses $\mathcal{O}$'s published data in the credential proving protocol.

Identity Revealing Protocol: This protocol is executed between $\mathcal{V}$ and $\mathcal{O}$, and reveals the relations between Cred and the data $\mathcal{U}$ sends to $\mathcal{V}$ in the credential proving protocol, and that identifies the user.

Security of Anonymous Credential System with Revocation In addition to **Unforgeability** and **Anonymity and Unlinkability**, the anonymous credential system with revocation needs the following security properties:

Traceability: Traceability demands that user $\mathcal{U}$ is unable to produce a credential such that either the honest opener $\mathcal{O}$ declares itself unable to identify the origin of the credential, or, $\mathcal{O}$ believes it has identified the origin but is unable to produce a correct proof of its claim.

The formal definition is as follows: Let Adv be an adversary, which runs in time at most τ, corrupts users, and interacts with Auth on their behalf. Now Adv obtains

credential Cred on m from Auth, and proves the credential to $\mathcal{V}$. If the probability that O fails in the credential revoking protocol of Cred is at most ϵ for any Adv, the anonymous credential system with revocation is (τ, ϵ)-traceable.

Non-frameability: Opener O is unable to create a proof, accepted by $\mathcal{V}$, that an honest user produced a certain valid proof of the credential unless the user really did produce the proof of the credential.

The formal definition is as follows: Let Adv be an adversary, and $\mathcal{U}$ be an honest user that does not produce an accepted proof of the credential Cred to an honest verifier $\mathcal{V}$. Now Adv, who acts as a user, the authority, and the opener, whose running time is at most τ, first successfully executes the credential proving protocol to $\mathcal{V}$ in the credential proving protocol, and then tries to prove to $\mathcal{V}$ that honest $\mathcal{U}$ is the user of the credential proving protocol by the identity revealing protocol. If the probability of Adv's success is at most ϵ for any Adv, the the anonymous credential system with revocation is (τ, ϵ)-non-frameable.

3 Assumptions and Basic Signature Scheme

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 $\mathtt{Adv}_{SDH}(q) \leftarrow Pr[\mathcal{A}\left(\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, g_2^x, ..., g_2^{x^q}\right) = \left(g_1^{\frac{1}{x+c}}, c\right); g_2 \overset{\cup}{\leftarrow} \mathbb{G}_2, g_1 \overset{\cup}{\leftarrow} \mathbb{G}_1, x, y \overset{\cup}{\leftarrow} \mathbb{Z}_p^*]$.

Definition 1. *Adversary* Adv (τ, ϵ)*-breaks the q-SDH problem if* Adv *runs in time at most* τ *and* $\mathtt{Adv}_{SDH}(q)$ *is at least* ϵ*. The* (q, τ, ϵ)*-SDH assumption holds if no adversary* Adv (τ, ϵ)*-breaks the q-SDH problem.*

3.2 The Decision Linear Diffie-Hellman Assumption [7]

Let $\mathbb{G}$ be a cyclic group of prime order p. Let u, v, h be generators of $\mathbb{G}$. The problem in $\mathbb{G}$ is defined as follows: Given $u, v, h, u^a, v^b, h^c \in \mathbb{G}$ as input, output yes if $a + b = c$ and no otherwise.

Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{Linear}$ in deciding the Decision Linear problem in $\mathbb{G}$ if $\mathtt{Adv}_{Linear} \leftarrow |Pr[\mathcal{A}\left(\mathbb{G}, u, v, h, u^a, v^b, h^{a+b}\right) = \mathtt{yes} : u, v, h \overset{\cup}{\leftarrow} \mathbb{G}, a, b \overset{\cup}{\leftarrow} \mathbb{Z}_p^*] - Pr[\mathcal{A}\left(\mathbb{G}, u, v, h, u^a, v^b, \eta\right) = \mathtt{yes} : u, v, h, \eta \overset{\cup}{\leftarrow} \mathbb{G}, a, b \overset{\cup}{\leftarrow} \mathbb{Z}_p^*]|$.

Definition 2. *The* (τ, ϵ)*-Decision Linear Diffie-Hellman Assumption (the Decision Linear Assumption) holds in* $\mathbb{G}$ *if no* τ*-time algorithm has advantage of at least* ϵ *in solving the Decision Linear Problem in* $\mathbb{G}$*.*

3.3 Basic Signature Scheme

We now describe a signature scheme [8] that is strongly existentially unforgeable against chosen plaintext attacks. This scheme is a fundamental element of the credential issuing protocol of our proposed anonymous credential systems.

Key Generation:
Randomly select generators $g_2, u_2, v_2 \xleftarrow{\cup} \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, and $v_1 \leftarrow \psi(v_2)$. Randomly select $x \xleftarrow{\cup} \mathbb{Z}_p^*$ and compute $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \psi, e, g_1, g_2, u_2, v_2)$ is the system parameter, w_2 is the public-key, and x is the secret-key.
Signature Generation:
Let $m \in \mathbb{Z}_p^*$ be the message to be signed. Signer $\mathcal{S}$ randomly selects $r, s \xleftarrow{\cup} \mathbb{Z}_p^*$, 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 system parameters (g_1, g_2, u_2, v_2) and public-key w_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 $e\left(\sigma, w_2 g_2^r\right) \stackrel{?}{=} e\left(g_1, g_2^m u_2 v_2^s\right)$. If they hold, the verification result is `valid`, otherwise `invalid`.

Proposition 1 (Security of the Basic Signature Scheme [8]).
If the $(q_S + 1, \tau', \epsilon')$*-SDH assumption holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the basic signature scheme is* (τ, q_S, ϵ)*-strongly existentially-unforgeable against adaptively chosen message attacks, provided that*

$$\epsilon \geq 3 q_S \epsilon', \tau \leq \tau' - \Theta\left(q_S^2 T\right),$$

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

4 Proposed Basic Anonymous Credential System

In this section, we describe the construction of the proposed basic anonymous credential system. We use a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with a computable isomorphism ψ, as in Section 2.2. We assume the basic signature scheme is strongly existentially unforgeable against chosen message attacks and the Strong Diffie-Hellman assumption holds in $\mathbb{G}_2$. We use the basic signature scheme in the credential issuing protocol of our proposed system.

4.1 Key Generation

Authority `Auth` generates public-key w_2 and secret-key x in the same way as in the signature scheme in Section 3.3.

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 `Auth`. When message m is received from $\mathcal{U}$, `Auth` signs m by using the signature scheme described in Section 3.3. $\mathcal{A}$ then sends triple signature (σ, r, s), to $\mathcal{U}$ as `Cred`, where $\sigma = \left(g_1^m u_1 v_1^s\right)^{1/(x+r)}$. $\mathcal{U}$ then verifies whether `Cred` is a valid signature on m. $\mathcal{U}$ calculates $\alpha \leftarrow w_2 g_2^r, \beta \leftarrow g_2^m u_2 v_2^s$ and verifies $e(\sigma, \alpha) \stackrel{?}{=} e(g_1, \beta)$.

4.3 Credential Proving Protocol

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

First, $\mathcal{U}$ randomises its credential, and sends the data including the randomised credential to $\mathcal{V}$ as follows: Prover $\mathcal{U}$ randomly selects $t, \theta \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$, 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 $(\sigma', \alpha', \beta')$ to the verifier $\mathcal{V}$. $\mathcal{V}$ then checks the equation $e(\sigma', \alpha') \stackrel{?}{=} e(g_1, \beta')$.

Second, $\mathcal{U}$ has to prove to $\mathcal{V}$ that $\mathcal{U}$ fairly created $(\sigma', \alpha', \beta')$. Therefore $\mathcal{U}$ proves knowledge for the following statement:

$$PK\{(\theta, r\theta) : \alpha' = w_2^\theta g_2^{r\theta}, \theta \neq 0\},\ PK\{(t, st) : \beta' = \left(g_2^m\right)^t u_2^t v_2^{st}, t \neq 0\}.$$

Details of this proof of knowledge are shown in **Figure.1**.

Figure.1 $PK\{(\theta, r\theta) : \alpha' = w_2^\theta g_2^{r\theta}, \theta \neq 0\}$

Common input: Public-key and α' **Prover's input:** $(\theta \neq 0, r\theta)$
Protocol:
Step1: $\mathcal{U}$ randomly selects R_1, R_2, $R_3 \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$, and computes $\gamma \leftarrow \alpha'^{R_1} g_2^{R_2} u_2^{R_3}, \delta \leftarrow \theta R_1 \bmod p, \omega \leftarrow r\theta R_1 + R_2 \bmod p$ and sends (γ, δ) to $\mathcal{V}$. If $\delta \neq 0$ then $\mathcal{V}$ outputs `reject`. Otherwise, $\mathcal{U}$ and $\mathcal{V}$ executes

$$PK\{(R_1, R_2, R_3, \omega) : \gamma = \alpha'^{R_1} g_2^{R_2} u_2^{R_3},\ \gamma / w_2^\delta = g_2^\omega u_2^{R_3}\}$$

as follows.
Step2: $\mathcal{U}$ picks random numbers r_1, r_2, r_3, $r_4 \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$, computes $A = \alpha'^{r_1} g_2^{r_2} u_2^{r_3}$, $B = g_2^{r_4} u_2^{r_3}$, and sends (A, B) to $\mathcal{V}$.
Step3: $\mathcal{V}$ sends a random number $b \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$ to $\mathcal{U}$.
Step4: $\mathcal{U}$ sends (c_1, c_2, c_3, c_4) to $\mathcal{V}$ such that $c_1 \leftarrow r_1 + bR_1 \bmod p, c_2 \leftarrow r_2 + bR_2 \bmod p, c_3 \leftarrow r_3 + bR_3 \bmod p, c_4 \leftarrow r_4 + b\omega \bmod p$.
Step5: $\mathcal{V}$ checks that $\alpha'^{c_1} g_2^{c_2} u_2^{c_3} \stackrel{?}{=} A\gamma^b$, $g_2^{c_4} u_2^{c_3} \stackrel{?}{=} B\left(\gamma / w_2^\delta\right)^b$.

$PK\{(t, st) : \beta' = \left(g_2^m\right)^t u_2^t v_2^{st}, t \neq 0\}$ can be proved in the same way as above. If $\mathcal{V}$ succeeds in these two proofs of the knowledge, $\mathcal{V}$ outputs `accept`, otherwise outputs `reject`.

4.4 Security

Unforgeability

Theorem 1. *If the basic signature scheme is* $(q_{Auth}, \tau, \epsilon)$*-strongly existentially unforgeable against chosen message attacks, then our proposed basic anonymous credential system is* $\left(\tau', q'_{Auth}, \epsilon'\right)$*-unforgeable, provided that*

$$\frac{1}{2}\left(1 - 2e^{\frac{\epsilon'}{2(\epsilon'-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon'-4}{2(p\epsilon'-4-2p)}n}\right) \geq \epsilon, \quad 2n\tau' + \Theta(T) \leq \tau, \quad q'_{Auth} \leq q_{Auth}.$$

Proof. Let us assume our system is not $\left(\tau', q'_{Auth}, \epsilon'\right)$-unforgeable. We will then show that the basic signature scheme is not $(\tau, q_{Auth}, \epsilon)$ -unforgeable. Under this assumption, adversary $\mathcal{U}$ can prove the two protocols in Section 4.3 as a prover with success probability greater than ϵ. We will then construct extractor $\mathcal{E}$ that outputs (σ, r, s).

Let us focus on protocol PK in **Figure.1**. $\mathcal{E}$ uses $\mathcal{U}$ as a black-box. After receiving (A, B), $\mathcal{V}$ sends $b \xleftarrow{\cup} \mathbb{Z}_p^*$ to $\mathcal{U}$ and receives (c_1, c_2, c_3, c_4). $\mathcal{E}$ then resets $\mathcal{U}$, and after receiving the same (A, B), $\mathcal{E}$ sends $b' \xleftarrow{\cup} \mathbb{Z}_p^*/\{b\}$ to $\mathcal{U}$ and receives $\left(c'_1, c'_2, c'_3, c'_4\right)$. If both runs of the protocols are accepted, $\mathcal{E}$ calculates $R_1 \leftarrow \frac{c'_1 - c_1}{b'-b} \bmod p, R_2 \leftarrow \frac{c'_2 - c_2}{b'-b} \bmod p, R_3 \leftarrow \frac{c'_3 - c_3}{b'-b} \bmod p, \omega \leftarrow \frac{c'_4 - c_4}{b'-b} \bmod p$. Note that (R_1, R_2, R_3, ω) satisfies $\gamma = \alpha'^{R_1} g_2^{R_2} u_2^{R_3}$ and $\gamma = g_2^{\omega} u_2^{R_3} w_2^{\delta}$. Now $\mathcal{E}$ succeeds in extracting (R_1, R_2, R_3). $\mathcal{E}$ then calculates $\theta \leftarrow \frac{\delta}{R_1} \bmod p, r \leftarrow \frac{\omega - R_2}{\theta R_1} \bmod p$. Note that $\alpha' = w_2^{\theta} g_2^{r\theta}$ and $\theta \neq 0$ since $\delta \neq 0$. In the same way, $\mathcal{E}$ computes the value (s, t) such that $\beta' = \left(g_2^m\right)^t u_2^t v_2^{st}$ and $t \neq 0$ from $PK\{(t, st) : \beta' = \left(g_2^m\right)^t u_2^t v_2^{st}, t \neq 0\}$, and then computes $\sigma \leftarrow \sigma'^{\frac{\theta}{t}}$. (σ, r, s) is a valid signature of the basic signature scheme.

Therefore, $\mathcal{E}$, using black-box $\mathcal{U}$, can forge the basic signature scheme (σ, r, s) with probability of at least ϵ' such that $\frac{1}{2}\left(1 - 2e^{\frac{\epsilon'}{2(\epsilon'-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon'-4}{2(p\epsilon'-4-2p)}n}\right) \geq \epsilon$ (by using the heavy row lemma and Chernoff bound). $2n$ is the number of times which $\mathcal{E}$ uses $\mathcal{U}$ as a black-box. The running time is at most $2n\tau' + \Theta(T)$, and the number of chosen message attack queries is at most q'_{Auth}. □

Anonymity and Unlinkability

Theorem 2. *Our proposed basic anonymous system is information-theoretically anonymous-and-unlinkable.*

Proof. The game described in **Anonymity and Unlinkability** of Section 2.3 is used to assess our system. If the protocols of proving knowledge are witness-indistinguishable, the system is anonymous and unlinkable; that is, in this game, the view of Step.3(a) and that of Step.3(b) are information-theoretically independent. The Σ-protocol is witness-indistinguishable. We show that the distributions of $\left(\sigma'_0, \alpha'_0, \beta'_0\right)$ and $\left(\sigma'_1, \alpha'_1, \beta'_1\right)$ are the same.

Let $b \in \{0, 1\}$. Using some set of numbers (z_b, y_b, w_b), $\sigma'_b = \left(g_1^{z_b}\right)^{\frac{t_b}{\theta_b}}, \alpha'_b = \left(g_2^{y_b}\right)^{\theta_b}, \beta'_b = \left(g_2^{w_b}\right)^{t_b}$ holds. Since $e\left(\sigma'_b, \alpha'_b\right) = e\left(g_1, \beta'_b\right)$, $z_b y_b = w_b \bmod p$ is satisfied. Thus, when the

values of σ'_b, α'_b are fixed, the value of β'_b can be uniquely decided. Therefore, there are two independent values in $\left(\sigma'_b, \alpha'_b, \beta'_b\right)$ and there are two random values t_b and θ_b. The distribution of $\left(\sigma'_b, \alpha'_b\right)$ is the same as the distribution of $\sigma'_b \overset{\cup}{\leftarrow} \mathbb{G}_1$ and $\alpha'_b \overset{\cup}{\leftarrow} \mathbb{G}_2$. Therefore, the distributions of $\left(\sigma'_0, \alpha'_0, \beta'_0\right)$ and $\left(\sigma'_1, \alpha'_1, \beta'_1\right)$ are the same. □

5 Proposed Anonymous Credential System with Revocation

We next show our proposed anonymous credential system with revocation. In this section, we assume that the Decision Linear Diffie-Hellman assumption holds in $\mathbb{G}_2$.

5.1 Key Generation

In addition to the secret and public keys generated in our proposed basic anonymous credential system, randomly selected $h, \hat{h}, a_2 \overset{\cup}{\leftarrow} \mathbb{G}_2$ are also used as system parameters. `Auth` proves $PK\{x : w_2 = g_2^x\}$ to get a certificate.

Now, in our proposed system with revocation, user $\mathcal{U}$ and opener $\mathcal{O}$ also generate secret and public keys. $\mathcal{U}$ randomly selects its secret-key $q \overset{\cup}{\leftarrow} \mathbb{Z}_p^*$, and calculates g_2^q(thus $g_1^q = \psi\left(g_2^q\right)$). $\mathcal{U}$ also generates a pair (pk_U, sk_U) of public-key and secret-key for some signature scheme. $\mathcal{U}$ publishes pk_U as its public-key. $\mathcal{O}$ randomly selects $\xi_1, \xi_2 \overset{\cup}{\leftarrow} \mathbb{Z}_p^*$ as its secret-key and computes $U \leftarrow g_2^{\xi_1}$, $V \leftarrow g_2^{\xi_2}$. $\mathcal{O}$ also publishes (U, V) as its public-key.

5.2 Credential Issuing Protocol

First, user $\mathcal{U}$ creates signature of g_2^q, $sig_U\left(g_2^q\right)$, using sk_U. $\mathcal{U}$ then sends g_2^q, $sig_U\left(g_2^q\right)$, and m as a message, for which $\mathcal{U}$ wants to obtain a certificate, to authority `Auth`.

Upon receiving these data from $\mathcal{U}$, `Auth` verifies $sig_U\left(g_2^q\right)$ by using pk_U, then signs m together with q by using the signature scheme described in Section 3.3. Namely, `Auth` creates the following signature (σ, r, s), where $\sigma = \left(g_1^m g_1^q u_1 v_1^s\right)^{1/(x+r)}$. `Auth` then sends the signature to $\mathcal{U}$ as `Cred`.

$\mathcal{U}$ then verifies whether `Cred` is a valid signature on m and q, $\mathcal{U}$ calculates $\alpha \leftarrow w_2 g_2^r, \beta \leftarrow g_2^m g_2^q u_2 v_2^s$ and verifies $e(\sigma, \alpha) \overset{?}{=} e(g_1, \beta)$. `Auth` writes $\left(\sigma, r, s, m, g_2^q, sig_U\left(g_2^q\right)\right)$ in database DB whenever `Auth` engages in the credential issuing protocol with users.

5.3 Credential Proving Protocol

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

`BL` $= (b_1, b_2, \cdots, b_l)$ is $\mathcal{V}$'s current blacklist of users who did something wrong (`Auth` can write and read, while $\mathcal{V}$ can only read `BL`), where $b_i\,(1 \le i \le l) \leftarrow g_2^{q_i}$($q_i$ is the i-th blacklisted user's secret-key). $\mathcal{U}$ encrypts its credential, and sends the data,

including an encrypted credential, data unique to the user related to revocation to $\mathcal{V}$ as follows:

Step1: $\mathcal{U}$ randomly selects $t_1, t_2, \theta, \rho \xleftarrow{\cup} \mathbb{Z}_p^*$, $f, \hat{f} \xleftarrow{\cup} \mathbb{G}_1$, and computes $\sigma' \leftarrow \sigma \cdot g_1^{t_1+t_2} = \left(g_1^m g_1^q u_1 v_1^s\right)^{\frac{1}{x+r}} \cdot g_1^{t_1+t_2}, \alpha' \leftarrow \left(w_2 g_2^r\right)^{\theta}, \beta' \leftarrow \left(g_2^m g_2^q u_2 v_2^s\right)^{\theta} \cdot \alpha'^{t_1+t_2}, d_1 \leftarrow \psi(U)^{t_1}, d_2 \leftarrow \psi(V)^{t_2}, \chi \leftarrow f^q \hat{f}^{\rho}$ and sends $\left(\sigma', \alpha', \beta', d_1, d_2, \chi, f, \hat{f}, g_2^{\rho}\right)$ to $\mathcal{V}$.

Step2: Verifier $\mathcal{V}$ verifies $e(\sigma', \alpha') \stackrel{?}{=} e(g_1, \beta')$ and $e(\chi, g_2) \stackrel{?}{\neq} e(f, b_i)\, e\left(\hat{f}, g_2^{\rho}\right)$ for every $i\,(1 \le i \le l)$.

Step3: $\mathcal{U}$ has to prove to $\mathcal{V}$ that $\mathcal{U}$ fairly created $(\chi, \sigma', \alpha', \beta', d_1, d_2)$. Therefore, $\mathcal{U}$ proves knowledge for the following statement: $PK\{(q, \rho, \theta, r\theta, s\theta, t_1, t_2) : \chi = f^q \hat{f}^{\rho}, \alpha' = w_2^{\theta} g_2^{r\theta}, \beta' = \left(g_2^m\right)^{\theta} g_2^{q\theta} u_2^{\theta} v_2^{s\theta} \alpha'^{t_1+t_2}, d_1 = \psi(U)^{t_1}, d_2 = \psi(V)^{t_2}, \theta \neq 0\}$. We detail this proof of knowledge in **Figure.2**.

Step4: If all verifications in **step.2** hold and the proof of knowledge is accepted, $\mathcal{V}$ finally outputs `accept`, otherwise outputs `reject`. Because blacklisted users cannot satisfy the latter verification in **step.2** as well as succeed in the proof of knowledge in **Figure.2**, this protocol provides blacklisting.

Figure.2 $PK\{(q, \rho, \theta, r\theta, s\theta, t_1, t_2) : \chi = f^q \hat{f}^{\rho},\ \alpha' = w_2^{\theta} g_2^{r\theta},\ \beta' = \left(g_2^m\right)^{\theta} g_2^{q\theta} u_2^{\theta} v_2^{s\theta} \alpha'^{t_1+t_2},\ d_1 = \psi(U)^{t_1},\ d_2 = \psi(V)^{t_2},\ \theta \neq 0\}$

Common input: $(\chi, \alpha', \beta', d_1, d_2)$ and public-key
Prover's input: $(q, \rho, \theta, r\theta, s\theta, t_1, t_2)$
Protocol:
Step1: $\mathcal{U}$ requests $\mathcal{V}$ to start the protocol. $\mathcal{V}$ then picks random numbers $b, \lambda \xleftarrow{\cup} \mathbb{Z}_p^*$ and computes $z \leftarrow h^b \hat{h}^{\lambda}$ (commitment of b) and sends z to $\mathcal{U}$.
Step2: $\mathcal{U}$ randomly selects $R_1, R_2, R_3, R_4 \xleftarrow{\cup} \mathbb{Z}_p^*$, computes $\gamma \leftarrow \alpha'^{R_1} g_2^{R_2} u_2^{R_3}$, $\delta \leftarrow \theta R_1 \bmod p$, $\omega \leftarrow r\theta R_1 + R_2 \bmod p$, $\xi \leftarrow \alpha'^{R_1} a_2^{R_4}$, and sends (γ, δ, ξ) to $\mathcal{V}$. If $\delta \neq 0$ then $\mathcal{V}$ outputs `reject`. Otherwise, $\mathcal{U}$ and $\mathcal{V}$ execute $PK\{(R_1, R_2, R_3, R_4, \omega, q, \rho, s, t_1, t_2, (t_1+t_2)R_1, (t_1+t_2)R_4) : \gamma = \alpha'^{R_1} g_2^{R_2} u_2^{R_3},\ \gamma / w_2^{\delta} = g_2^{\omega} u_2^{R_3},\ \chi = f^q \hat{f}^{\rho},\ \xi = \alpha'^{R_1} a_2^{R_4},\ g_2^{m\delta} u_2^{\delta} = \beta'^{R_1} g_2^{-\delta q} v_2^{\delta s} \xi^{-(t_1+t_2)} a_2^{(t_1+t_2)R_4},\ g_2^{m\delta} u_2^{\delta} = \beta'^{R_1} g_2^{-\delta q} v_2^{\delta s} \alpha'^{-(t_1+t_2)R_1})\}$, as follows.
Step3: $\mathcal{U}$ picks random numbers $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12} \xleftarrow{\cup} \mathbb{Z}_p^*$, computes $A = \alpha'^{r_1} g_2^{r_2} u_2^{r_3}$, $B = g_2^{r_5} u_2^{r_3}$, $C = f^{r_6} \hat{f}^{r_7}$, $D = \alpha'^{r_1} a_2^{r_4}$, $E = \beta'^{r_1} g_2^{-\delta r_6} v_2^{-\delta r_8} \xi^{-(r_9+r_{10})} a_2^{r_{12}}$, $F = \beta'^{r_1} g_2^{-\delta r_6} v_2^{-\delta r_8} \alpha'^{-r_{11}}$, $G = \psi(U)^{r_9}$, $H = \psi(V)^{r_{10}}$, and sends (A, B, C, D, E, F, G, H) to $\mathcal{V}$.
Step4: $\mathcal{V}$ sends b, λ to $\mathcal{U}$ in order to open the commitment.
Step5: $\mathcal{U}$ sends $(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12})$ to $\mathcal{V}$ such that $c_1 \leftarrow r_1 + bR_1 \bmod p, c_2 \leftarrow r_2 + bR_2 \bmod p$, $c_3 \leftarrow r_3 + bR_3 \bmod p, c_4 \leftarrow r_4 + bR_4 \bmod p$, $c_5 \leftarrow r_5 + b\omega \bmod p, c_6 \leftarrow r_6 + bq \bmod p, c_7 \leftarrow r_7 + b\rho \bmod p, c_8 \leftarrow r_8 + bs \bmod p$, $c_9 \leftarrow r_9 + bt_1 \bmod p$, $c_{10} \leftarrow r_{10} + bt_2 \bmod p$, $c_{11} \leftarrow r_{11} + b(t_1+t_2)R_1 \bmod p$, $c_{12} \leftarrow r_{12} + b(t_1+t_2)R_4 \bmod p$.

Step6: $\mathcal{V}$ checks that $\alpha'^{c_1} g_2^{c_2} u_2^{c_3} \stackrel{?}{=} A\gamma^b$, $g_2^{c_5} u_2^{c_3} \stackrel{?}{=} B\left(\gamma / w_2^{\delta}\right)^b$, $f^{c_6} \hat{f}^{c_7} \stackrel{?}{=} C\chi^b$, $\alpha'^{c_1} a_2^{c_4} \stackrel{?}{=} D\xi^b$, $\beta'^{c_1} g_2^{-\delta c_6} v_2^{-\delta c_8} \xi^{-(c_9+c_{10})} a_2^{c_{12}} \stackrel{?}{=} E\left(g_2^{m\delta} u_2^{\delta}\right)^b$, $\beta'^{c_1} g_2^{-\delta c_6} v_2^{-\delta c_8} \alpha'^{-c_{11}} \stackrel{?}{=} F\left(g_2^{m\delta} u_2^{\delta}\right)^b$, $\psi(U)^{c_9} \stackrel{?}{=} G d_1^b$, $\psi(U)^{c_{10}} \stackrel{?}{=} H d_2^b$.

If $\mathcal{V}$ succeeds in this proof of knowledge, $\mathcal{V}$ outputs `accept`, otherwise outputs `reject`.

5.4 Identity Revealing Protocol

If verifier $\mathcal{V}$ finds that a user has misused his credential, $\mathcal{V}$ informs O. O then reveals the credential of the user as follows:

Step1: $\mathcal{V}$ sends σ', d_1, and d_2 to O, and asks O to reveal the user who created σ'.

Step2: O computes $\sigma = \frac{\sigma'}{{d_1}^{1/\xi_1} {d_2}^{1/\xi_2}}$ and searches the database DB to identify the user $\mathcal{U}$. O then finds $\left(r, s, m, g_2^q, sig_U\left(g_2^q\right)\right)$ in DB (they are related to σ) and sends $\left(\sigma, r, s, m, g_2^q, sig_U\left(g_2^q\right)\right)$ to $\mathcal{V}$.

Step3: O proves knowledge for the following statement: $PK\{(\xi_1, \xi_2) : U = g_2^{\xi_1}, V = g_2^{\xi_2}, \sigma = \frac{\sigma'}{{d_1}^{1/\xi_1} {d_2}^{1/\xi_2}}\}$. We detail this proof of knowledge in **Figure.3**. $\mathcal{V}$ checks $e\left(\sigma, w_2 g_2^r\right) \stackrel{?}{=} e\left(g_1, g_2^m g_2^q u_2 v_2^s\right)$.

$\mathcal{V}$ then finally can find that σ' was created fairly by $\mathcal{U}$, by using pk_U and checking whether $sig_U\left(g_2^q\right)$ is a valid signature on g_2^q. This protocol provides the identity revealing.

Figure.3 $PK\{(\xi_1, \xi_2) : U = g_1^{\xi_1}, V = g_2^{\xi_2}, \sigma = \sigma' / \left(d_1^{1/\xi_1} d_2^{1/\xi_2}\right)\}$.

Common input: Public key and $(d_1, d_2, \sigma, \sigma')$
Prover's input: (ξ_1, ξ_2)
Protocol:
Step1: O picks random numbers R_1, $R_2 \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$, computes $Y_1 = g_1^{R_1}, Y_2 = g_1^{R_2}, X_1 = d_1^{1/\xi_1}, X_2 = d_2^{1/\xi_2}, Y_3 = X_1^{R_1}, Y_4 = X_2^{R_2}$, and sends these data to $\mathcal{V}$.
Step2: $\mathcal{V}$ sends a random number $b \stackrel{\cup}{\leftarrow} \mathbb{Z}_p^*$ to O.
Step3: O sends (c_1, c_2) to $\mathcal{V}$ such that $c_1 \leftarrow R_1 + b\xi_1 \bmod p$, $c_2 \leftarrow R_2 + b\xi_2 \bmod p$.
Step4: $\mathcal{V}$ checks that $g_1^{c_1} \stackrel{?}{=} Y_1 U^b, g_2^{c_2} \stackrel{?}{=} Y_2 V^b, X_1^{c_1} \stackrel{?}{=} Y_3 d_1^b, X_2^{c_2} \stackrel{?}{=} Y_4 d_2^b, \sigma \stackrel{?}{=} \sigma' / X_1 X_2$. If it holds, $\mathcal{V}$ outputs `accept`, otherwise outputs `reject`.

Remark: If we require a stronger non-frameability where verifier $\mathcal{V}$ as well as an opener is dishonest, $\mathcal{V}$ should publish a transcript of the credential proving protocol in which $\mathcal{V}$'s challenge is a hashed value of prover's first message in a Σ-protocol. However, the protocol in **Figure.2** is not a Σ-protocol as challenge b is committed in **Step.1**. Hence, in order to guarantee the stronger non-frameability, we should change the protocol in **Figure.2** to a standard Σ-protocol, and challenge message, b, by $\mathcal{V}$ is a hash value of (A, B, C, D, E, F, G, H). Instead, to prove the anonymity-and-unlinkability, an oracle-linear assumption is needed (it will be shown in the full version of this paper).

5.5 Security

Unforgeability

Theorem 3. *If the basic signature scheme is* $(q_{Auth}, \tau, \epsilon)$*-strongly existentially unforgeable against chosen message attacks, our proposed anonymous credential system with revocation is* $\left(\tau', q'_{Auth}, \epsilon'\right)$*-unforgeable, provided that*

$$\frac{1}{2}\left(1 - 2e^{\frac{\epsilon'}{2(\epsilon'-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon'-2}{2(p\epsilon'-2-2p)}n}\right) \geq \epsilon, \;\; 2n\tau'' + \Theta(T) \leq \tau, \;\; q'_{Auth} \leq q_{Auth} \; .$$

Proof. The proof follows the same approach used in our proposed basic system. Assuming our system is not $(\tau', q_{Auth}, \epsilon')$-unforgeable, $\mathcal{U}$ can forge $(\sigma', \alpha', \beta', d_1, d_2)$ that satisfies verifier $\mathcal{V}$'s equation in the credential proving protocol with $(\tau', q_{Auth}, \epsilon')$. We then construct extractor $\mathcal{E}$ that outputs the original credential (σ, r, s) (and U, V). □

Anonymity and Unlinkability

Theorem 4. *If the* (τ, ϵ)*-Decision Linear Assumption holds in* $\mathbb{G}_2$ *then our proposed anonymous credential system with revocation is* (τ', ϵ')*-anonymous-and unlinkable, provided that* $\epsilon' \geq \epsilon, \tau' \leq \tau$.

Proof. Assume Adv is an adversary that (τ', ϵ')-breaks the anonymity and unlinkability of our proposed anonymous credential system with revocation. We construct an algorithm $\mathcal{A}$ that, by interacting with Adv, solves the Decision Linear Problem in time τ with advantage ϵ.

Algorithm $\mathcal{A}$ is given random instance $(\mathbb{G}_2, U, V, g_2, U^{t_1}, V^{t_2}, \eta)$ of the Decision Linear Problem. It randomly selects $u_2, v_2 \xleftarrow{\cup} \mathbb{G}_2$ and gives $(\mathbb{G}_2, g_2, u_2, v_2)$ to Adv as a system parameter. Adv outputs public key w_2 and proves $PK\{x : w_2 = g_2^x\}$. $\mathcal{A}$ extracts x by using Adv as a black-box prover. $\mathcal{A}$ then generates two users'($\mathcal{U}_0$ and $\mathcal{U}_1$) secret-key i.e., selects random $q_0, q_1 \xleftarrow{\cup} \mathbb{Z}_p^*$ and users' signature key pair $sk_{\mathcal{U}_0}, pk_{\mathcal{U}_0}, sk_{\mathcal{U}_1}, pk_{\mathcal{U}_1}$. It then sends $\left(g_2^{q_0}, g_2^{q_1}, pk_{\mathcal{U}_0}, pk_{\mathcal{U}_1}\right)$ to Adv and carries out the credential issuing protocol with Adv, as $\mathcal{U}_0$ and $\mathcal{U}_1$. $\mathcal{A}$ obtains (σ_0, r_0, s_0) and (σ_1, r_1, s_1), where $\sigma_0 = \left(g_1^m g_1^{q_0} u_1 v_1^{s_0}\right)^{1/(x+r_0)}$, and $\sigma_1 = \left(g_1^m g_1^{q_1} u_1 v_1^{s_1}\right)^{1/(x+r_1)}$.

Next, $\mathcal{A}$ can execute the credential proving protocol with $\mathcal{U}_0$ and $\mathcal{U}_1$ polynomial-times. When Adv queries $\mathcal{U}_{b'}$ $(b' \in \{0,1\})$, $\mathcal{A}$ selects $\theta, r_1, r_2 \xleftarrow{\cup} Z_p^*$, and computes $\sigma' \leftarrow \sigma_{b'} \cdot \psi(\eta) \cdot g_1^{r_1+r_2}, \alpha' \leftarrow \left(w_2 g_2^{r_{b'}}\right)^{\theta}, \beta' \leftarrow \left(g_2^m g_2^{q_{b'}} u_2 v_2^{s_{b'}}\right)^{\theta} \cdot \eta^{\theta(x+r_d)} g_2^{r_1+r_2}, d_1 \leftarrow \psi(U^{t_1}) g_2^{r_1}, d_2 \leftarrow \psi(V^{t_2}) g_2^{r_2}$. $\mathcal{A}$ randomly chooses $\rho_{b'} \xleftarrow{\cup} \mathbb{Z}_p^*$ and $f_{b'}, \hat{f}_{b'} \xleftarrow{\cup} \mathbb{G}_1$, and calculates $\chi_{b'} \leftarrow f_{b'}^{q_{b'}} \hat{f}_{b'}^{\rho_{b'}}$, and sends them to Adv as $\mathcal{U}_{b'}$. $\mathcal{A}$ first executes the protocol and obtains the value of b in **Step.3**, and resets Adv. $\mathcal{A}$ then re-executes the proof of knowledge protocol. Now $\mathcal{A}$ knows the value of b, so $\mathcal{A}$ can successfully finish the proof of knowledge protocol without knowing the witness. $\mathcal{A}$ and Adv then engage in the credential proving protocol. Adv now requests its anonymity challenge. $\mathcal{A}$ chooses uniformly random bit of $d \in \{0,1\}$, selects random $\theta \xleftarrow{\cup} \mathbb{Z}_p^*$ and computes $\sigma' \leftarrow \sigma_d \cdot \psi(\eta) \cdot g_1^{r_1+r_2}$, $\alpha' \leftarrow \left(w_2 g_2^{r_d}\right)^{\theta}, \beta' \leftarrow \left(g_2^m g_2^{q_d} u_2 v_2^{s_d}\right)^{\theta} \cdot \eta^{\theta(x+r_d)} g_2^{r_1+r_2}, d_1 \leftarrow \psi(U^{t_1}) g_2^{r_1}, d_2 \leftarrow \psi(V^{t_2}) g_2^{r_2}$.

$\mathcal{A}$ and Adv then engage in the credential proving knowledge of σ_d. After this, Adv can query $\mathcal{U}_0$ and $\mathcal{U}_1$ polynomial-times. The procedure is just the same as the above.

Finally, Adv outputs bit d'. If $d' = d$, $\mathcal{A}$ outputs yes(guesses $\eta = g_2^{t_1+t_2}$). Else(if $d' \neq d$), $\mathcal{A}$ outputs no. If $\eta = g_2^{t_1+t_2}$, $Pr[\mathcal{A}\left(\mathbb{G}_2, U, V, g_2, U^{t_1}, V^{t_2}, g_2^{t_1+t_2}\right) = \mathtt{yes} : U, V, g_2, \xleftarrow{\cup} \mathbb{G}_2, t_1, t_2 \xleftarrow{\cup} \mathbb{Z}_p^*] = Pr[d' = d]$. If $\eta \neq g_2^{t_1+t_2}$, let $\eta = g_2^{\zeta}$. $\sigma' = \sigma_b \cdot g_1^{\zeta}$ holds. $\alpha' = \left(w_2 g_2^{r_b}\right)^{\theta}$ and $\beta' = \left(g_2^{m} g_2^{q_b} u_2 v_2^{s_b}\right)^{\theta} \cdot \alpha'^{\zeta}$ are satisfied. Since there are two independent elements in $(\sigma', \alpha', \beta')$ and these are randomised by θ and ζ, the distribution of (α', β') is just the same as the following distribution $\alpha' \xleftarrow{\cup} \mathbb{G}_2, \beta' \xleftarrow{\cup} \mathbb{G}_2$. Therefore, the distribution is independent of the value of d, thus $Pr[\mathcal{A}(\mathbb{G}_2, U, V, g_2, U^{t_1}, V^{t_2}, \eta) = \mathtt{yes} : U, V, g_2, \eta \xleftarrow{\cup} \mathbb{G}_2, t_1, t_2 \xleftarrow{\cup} \mathbb{Z}_p^*] = \frac{1}{2}$. □

Traceability

Theorem 5. *If the basic signature scheme is $(q_{Auth}, \tau, \epsilon)$-strongly existentially unforgeable against chosen message attacks, our proposed anonymous credential system is $\left(\tau', q'_{Auth}, \epsilon'\right)$-traceable, provided that*

$$\frac{1}{2}\left(1 - 2e^{\frac{\epsilon'}{2(\epsilon'-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon'-2}{2(p\epsilon'-2-2p)}n}\right) \geq \epsilon,\ 2n\tau'' + \Theta(T) \leq \tau,\ q_{Auth'} \leq q_{Auth}\ .$$

Proof. Assume Adv is an adversary that $\left(\tau', q'_{Auth}, \epsilon'\right)$-breaks the traceability of our proposed anonymous credential system with revocation. We construct an extractor $\mathcal{E}$ that, by interacting with Adv, can forge the basic signature scheme in time τ with advantage ϵ, where q'_{Auth} is the maximum number of queries made by Adv.

Adv succeeds in generating such $(\sigma', \alpha', \beta', d_1, d_2)$ that is accepted by $\mathcal{V}$, but $\mathcal{O}$ fails in revealing the original credential stored in DB. $\mathcal{E}$ then extracts (σ, r, s) by using Adv as a black-box in the same way as in the proof of **Unforgeability**. Since (σ, r, s) is not in DB, it is a forged signature of the basic signature scheme. □

Non-frameability

Theorem 6. *If the user's signature scheme is $(q_{Auth}, \tau, \epsilon)$-existentially unforgeable against chosen message attacks and the discrete logarithm problem in $\mathbb{G}_1$ is (τ', ϵ')-hard, then our proposed anonymous credential system with revocation is $\left(\tau'', q''_{Auth}, \epsilon''\right)$-non-frameable, provided that*

$$\frac{1}{2}\left(1 - 2e^{\frac{\epsilon''}{2(\epsilon''-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon''-2}{2(p\epsilon''-2-2p)}n}\right) \geq \epsilon',\ \epsilon'' \geq \epsilon,\ min\left(\frac{\tau' - \Theta(T)}{2n}, \tau\right) \geq \tau'',\ q_{Auth'} \leq q_{Auth}\ .$$

Proof. Assume Adv is an adversary that (τ', ϵ')-breaks the non-frameability of our proposed anonymous credential system with revocation. We then construct an algorithm $\mathcal{A}$ that, by interacting with Adv, breaks the unforgeability of the user's signature scheme or the discrete logarithm problem.

Algorithm $\mathcal{A}$ is given public-key pk_U of the user's signature scheme and instance $g_2, g_2^q \in \mathbb{G}_2$ of the discrete logarithm problem. $\mathcal{A}$ gives Adv $\mathbb{G}_2, g_2$ as a system parameter. Adv generates authority's public-keys and opener's public keys. Adv then generates

its secret-key. $\mathcal{A}$ concurrently executes the following two procedures. The first one is breaking the unforgeability of the user's signature scheme. $\mathcal{A}$ generates a user $\mathcal{U}$ and registers pk_U as the public-key of $\mathcal{U}$. The second one is breaking the discrete logarithm problem. $\mathcal{A}$ generates a user $\mathcal{U}$, generates a new key $\left(pk'_U, sk'_U\right)$, and uses g_2^q as the value given to `Adv` (`Auth`) at credential issuing protocol.

`Adv` first generates its secret-key as a user, and creates its credential $\texttt{Cred}_{Adv}$ on m. `Adv` then executes the credential proving protocol of σ_{Adv} with an honest verifier $\mathcal{V}$. Eventually, `Adv` employs the identity revealing protocol with $\mathcal{V}$, and creates accepted proof for $\mathcal{V}$ that $\mathcal{U}$, who is an honest user, produced the proof of $\texttt{Cred}_{Adv}$. This means `Adv` outputs $\left(\sigma, r, s, sig_U\left(g_2^q\right), g_2^q, m\right)$ that is accepted by $\mathcal{V}$ as $\mathcal{U}$'s proof of $\texttt{Cred}_{Adv}$.

If `Adv` outputs in the first procedure, $\left(g_2^q, sig_U\left(g_2^q\right)\right)$ is a forged signature of the user's signature scheme. If `Adv` outputs in the second procedure, $\mathcal{A}$ extracts q in the same manner as in the proof of **Unforgeability** by using `Adv` as a black-box. Thus, $\mathcal{A}$ can forge the signature scheme or break the discrete logarithm problem, with the maximum time $\tau' \geq 2n\tau'' + \Theta(T)$ and the advantage $\frac{1}{2}\left(1 - 2e^{\frac{\epsilon''}{2(\epsilon''-1)}n}\right)\left(1 - 2e^{\frac{p\epsilon''-2}{2(p\epsilon''-2-2p)}n}\right) \geq \epsilon'$. □

5.6 Comparison

We turn now to the efficiency of our anonymous credential system. The upper table in **Table.1** is a comparison of our basic system and an existing system [3]. "pk" means the public-key specific to each user (excluding the system parameters), and "sk" means the secret-key. "Size of `Prov`" means communication complexity between $\mathcal{U}$ and $\mathcal{V}$ in the credential proving protocol (`Prov` denotes a credential proving protocol). "Ops" means the number of operations.

We show a comparison of our system with revocation and the existing system [5] in the lower table in **Table.1**. "Size of `Reveal`" means communication complexity between $\mathcal{O}$ and $\mathcal{V}$ in the identity revealing protocol (`Reveal` denotes an identity revealing protocol). N is the size of an RSA modulus. A number l means the number of blacklisted users.

6 Conclusion

We presented two anonymous credential systems. The basic anonymous credential system is unforgeable under the Strong Diffie-Hellman assumption and is information-theoretically anonymous-and-unlinkable. It also seems more efficient than an existing system [3] (See **Table.1**). Our proposed anonymous credential system with revocation is secure under the Strong Diffie-Hellman assumption and the Decision Linear assumption. Our system, however, offers two revocation schemes: Blacklisting and identity revealing of users who act wrongly. Our system is also secure in the standard model.

References

1. Chaum, D: Security without identification: transaction systems to make big brother obsolete. Commun.ACM. **28**(10) (1985) 1030-1044

Table 1. Comparison

	CL04 [3]	Our proposed basic system
Assumption	LRSW	SDH
Size of pk	3 elements in $\mathbb{G}_1$	1 element in $\mathbb{G}_1$
Size of sk	3 elements in $\mathbb{Z}_p$	1 element in $\mathbb{Z}_p$
Size of `Cred`	5 elements in $\mathbb{G}_1$	1 element in $\mathbb{G}_1$, 2 elements in $\mathbb{Z}_p$
Size of `Prov`	5 elements in $\mathbb{G}_1$, 1 element in $\mathbb{G}_T$, 4 elements in $\mathbb{Z}_p$	9 elements in $\mathbb{G}_1$, 12 elements in $\mathbb{Z}_p$
Ops to issue `Cred`	4.3 exps in $\mathbb{G}_1$	1.3 exps in $\mathbb{G}_1$
Ops to verify `Cred`	4.3 exps in $\mathbb{G}_1$, 8 pairings	2.6 exps in $\mathbb{G}_1$, 2 pairings
Ops to prove in `Prov`	4 pairings, 5 exps in $\mathbb{G}_1$, 1.3 exps in $\mathbb{G}_T$	11.4 exps in $\mathbb{G}_1$
Ops to verify in `Prov`	10 pairings, 1.3 exps in $\mathbb{G}_1$	2 pairings, 5.2 exps in $\mathbb{G}_1$
	CL01 [5]	Our proposed system with revocation
Assumption	strong RSA, DDH	SDH
Size of pk	10 elements in $\mathbb{Z}_N^*$	3 elements in $\mathbb{G}_1$, size of sk_U
Size of sk	7 elements in $\mathbb{Z}_N^*$	4 elements in $\mathbb{Z}_p$, size of pk_U
Size of `Cred`	3 elements in $\mathbb{Z}_N^*$	1 element in $\mathbb{G}_1$, 2 elements in $\mathbb{Z}_p$
Size of `Prov`	9 elements in $\mathbb{Z}_N^*$	20 elements in $\mathbb{G}_1$, 15 elements in $\mathbb{Z}_p$
Size of `Reveal`	15 elements in $\mathbb{Z}_N^*$	12 elements in $\mathbb{G}_1$, 3 elements in $\mathbb{Z}_p$
Ops to issue `Cred`	1 exp in $\mathbb{Z}_N^*$	1.3 exps in $\mathbb{G}_1$, Ops to issue $sig_U\left(g_2^q\right)$
Ops to verify `Cred`	1 exp in $\mathbb{Z}_N^*$	2.6 exps in $\mathbb{G}_1$, 2 pairings
Ops to prove in `Prov`	6.5 exps in $\mathbb{Z}_N^*$	20.6 exps in $\mathbb{G}_1$
Ops to verify in `Prov`	3.9 exps in $\mathbb{Z}_N^*$	$(3l+2)$ pairings, 10.4 exps in $\mathbb{G}_1$
Ops to open in `Reveal`	10.2 exps in $\mathbb{Z}_N^*$	7.3 exps in $\mathbb{G}_1$, Ops to verify $sig_U\left(g_2^q\right)$
Ops to verify in `Reveal`	5.9 exps in $\mathbb{Z}_N^*$	2 pairings, 7.5 exps in $\mathbb{G}_1$
Blacklisting	Not available	Available
Identity revealing	Available	Available

2. Lysyanskaya, A.: Signature Schemes and Applications to Cryptographic Protocol Design. Ph.D thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2002)
3. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. CRYPTO'04, LNCS, Vol.3152 (2004) 56–72
4. Lysyanskaya, A., Rivest, R. L., Sahai, A., Wolf, S.: Pseudonym Systems. The 6th Annual International Workshop on Selected Areas in Cryptography, LNCS, Vol.1758 (2000) 184–199
5. Camenisch, J., Lysyanskaya, A.: An efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. EUROCRYPT'01, LNCS, Vol.2045 (2001) 93–118
6. Tsang, P., Au, M. H., Kapadia, A., Smith, S.: Blacklistable Anonymous Credentials: Blocking Misbehaving Users without TTPs, CCS'07, 14th ACM conf. on computer and communications security (2007) 72–81
7. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. CRYPTO'04, LNCS, Vol.3152 (2004) 41–55
8. Okamoto, T.: Efficient Blind and Partially Blind Signatures Without Random Oracles. TCC'06, LNCS, Vol.3876 (2006) 80–99