SCIS 2008 The 2008 Symposium on Cryptography and Information Security
Miyazaki, Japan, Jan. 22-25, 2008
The Institute of Electronics, Information and Communication Engineers

An Efficient Anonymous Credential System with Revocation

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). In our system, an user gets a proof of identity from an authority 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. Our system also has two credential revoking functions. First, there exists an opener that can reveal the identity of the user, who proved its credential to the verifier. Second, the verifier can reject the user's challenge of proof of its credential in advance if the user acted wrong and is blacklisted by the authority.

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

1 Introduction

The concept of anonymous credential systems was introduced by Chaum[5], and many researchers since then have proposed anonymous credential systems in order to counter some of the privacy difficulties related to identity certificates, and to implement ideal physical credentials, like entry certification, driver's licenses, and so on.

The basic properties of any anonymous credential system are as follows: It should be impossible for a user to forge a credential for it. Credentials also must be anonymous, 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. There are three basic protocols; Key Generation, Credential Issuing, Credential Proving. The details of the history and motivation behind anonymous credentials can be found here [7].

A credential revoking function is desirable in anonymous credential systems. One of the existing anonymous credential systems with revocation can reveal the user's identity if the user misuses the credential[3], and is based upon the strong RSA assumption and Decisional Diffie-Hellman assumption. There is another existing anonymous credential system with revocation[9], which enables a verifier to reject black-listed users. This system is based upon the q-SDH assumption and Decisional Diffie-Hellman assumption, and uses random oracle model. We also propose an anonymous credential system with revocation, which provides both of above function of revocation, and is unforgeable under the q-SDH assumption, and is anonymous-and-unlinkable under the Decision Linear Diffie-Hellman assumption(the Decision Linear assumption) without random oracle model.

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

2 Preliminaries

2.1 Notation

We will use the 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 $G = \langle g\rangle = \langle h\rangle$.

2.2 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 $\psi(g_2) = g_1$.
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, $b \in \mathbb{Z}_{\mathbb{P}}^*$, $e\left(u^a, v^b\right) = e(u,v)^{ab}$
   - (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.

2.3.1 Definition of Anonymous Credential System with Revocation

An anonymous credential system consists of parties which are users, authorities, verifiers, and openers. An anonymous credential system performs the following operations.

Key Generation:
An authority `Auth`, an user $\mathcal{U}$, and an opener O, given security parameter 1^k, outputs respectively a pair of public-key and secret-key.

Credential Issuing Protocol:
$\mathcal{U}$ 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 user's input m, user's secret-key, and authority's secret-key. At the end of the protocol, $\mathcal{U}$ obtains a credential `Cred`, corresponding to m. `Auth` has a database `DB` to record the data used in the credential issuing protocol with $\mathcal{U}$. An opener O can read but cannot write `DB`.

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 the user's possession of `Cred`.

`BL` is a $\mathcal{V}$'s current black-list of users who acted wrong outside the protocols (`Auth` can write and read, and $\mathcal{V}$ can only read `BL`). If $\mathcal{U}$ is black-listed, $\mathcal{V}$ outputs `reject`. Thus, $\mathcal{V}$ outputs `accept` if $\mathcal{U}$ is not listed on `BL` and can prove that it really has a valid `Cred`, otherwise outputs `reject` at the end of the protocol.

Credential Revealing Protocol:
O can trace a credential to the user that showed the credential. When $\mathcal{V}$ notices that $\mathcal{U}$ acts wrong after it finished proving knowledge of its credential `Cred` to $\mathcal{V}$, $\mathcal{V}$ requests to an opener O to identify the $\mathcal{U}$'s credential. O and $\mathcal{V}$ then executes the credential revealing protocol by using its secret-key and `DB` and reveals that $\mathcal{U}$ is the one who challenged the accepted proof of `Cred`.

2.3.2 Security of Anonymous Credential System with Revocation

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 below.

Let us consider the following game. Let `Adv` be an adversary, which has no information about the secret-key of `Auth`. `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.

`Adv` then forges `Cred` which is a credential of m. m is data which has not been chosen by `Adv` yet. Finally `Adv` and $\mathcal{V}$ execute the credential proving protocol of `Cred`, 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, 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 a verifier $\mathcal{V}$ and an authority `Auth` to find anything about a user $\mathcal{U}$, except the fact that $\mathcal{U}$ has some set of credentials, even if $\mathcal{V}$ cooperates with other verifiers or authorities(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.
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 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 (τ, ϵ)-anonymous-and-unlinkable if the advantage of any adversary, whose running time is at most τ, is at most ϵ.

Traceability:
Traceability demands that a user $\mathcal{U}$ is unable to produce a credential such that either the honest opener O declares itself unable to identify the origin of the credential, or, 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` issues credential `Cred` on m with `Auth`, and proves credential to $\mathcal{V}$. If the probability that O fails in the credential revoking protocol of `Cred` is at most ϵ, the anonymous credential system with revocation is (τ, ϵ)-traceable.

Non-frameability:
An 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, $\mathcal{U}$ 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 and an opener, whose running time is at most τ, first proves Cred to $\mathcal{V}$ in the credential proving protocol, and then tries to prove to $\mathcal{V}$ that $\mathcal{U}$ is the user that produced the accepted proof of Cred in the credential revoking protocol. If the probability of Adv's success is at most ϵ, the the anonymous credential system with revocation is (τ, ϵ)-non-frameable.

2.4 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 following game among adversary Adv 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 a, and (pk, sk) is given to signer $\mathcal{S}$.

2. Queries to signing oracle:
   Adv adaptively requests $\mathcal{S}$ (or signing oracle) to sign on at most q_S messages of his choice $m_1, ..., m_{q_S}$, $\mathcal{S}$ responds to m_i with a signature $\sigma_i = \mathcal{S}(sk, m_i)$.

3. Output:
   Eventually, Adv outputs pair (m, σ). a wins the game if
   (a) (m, σ) is not any of (m_i, σ_i) $(i = 1, ..., q_S)$
   (b) $\mathcal{V}$(pk, m, σ)=accept.
   We define $\mathtt{Adv}_{Sig}^{unforge}$ to be the probability that Adv wins the above game, taken over the coin tosses made by Adv, $\mathcal{G}$, and $\mathcal{S}$.

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

3 Assumptions and Basic Schemes

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 $\mathcal{A}$.

Difinition.2 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

Let $\mathbb{G}_1$ be the cyclic group of prime order p. Let u, v, h be generators of $\mathbb{G}_1$. The problem in $\mathbb{G}_1$ is defined as follows: Given $u, v, h, u^a, v^b, h^c \in \mathbb{G}_1$ 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 $\mathbb{G}_1$ if

$$\mathtt{Adv}_{Linear} \leftarrow |Pr[\mathcal{A}\left(u, v, h, u^a, v^b, h^{a+b}\right) = \mathtt{yes} : u, v, h \overset{R}{\leftarrow} \mathbb{G}_1, a, b \overset{R}{\leftarrow} \mathbb{Z}_p^*] - Pr[\mathcal{A}\left(u, v, h, u^a, v^b, \eta\right) = \mathtt{yes} : u, v, h, \eta \overset{R}{\leftarrow} \mathbb{G}_1, a, b \overset{R}{\leftarrow} \mathbb{Z}_p^*]|.$$

Difinition.3 The (τ, ϵ)-Decision Linear Diffie-Hellman Assumption (the Decision Linear Assumption) holds in $\mathbb{G}_1$ if no τ-time algorithm has advantage of at least ϵ in solving the Decision Linear Problem in $\mathbb{G}_1$.

3.3 Basic Signature Scheme

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

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 $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. $(g_1, g_2, w_2, u_2, v_2)$ 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) 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 $e\left(\sigma, w_2 g_2^r\right) \overset{?}{=} e\left(g_1, g_2^m u_2 v_2^s\right)$. If they hold, the verification result is valid, otherwise invalid.

Theorem.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 Anonymous Credential System with Revocation

In this section, we describe the construction of the proposed 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

First, an authority `Auth` randomly selects its secret-key $x \in \mathbb{Z}_p^*$. `Auth` then randomly selects generators $g_2, u_2, v_2 \in \mathbb{G}_2$ and sets $w_2 \leftarrow g_2^x$, $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, and $v_1 \leftarrow \psi(v_2)$. `Auth` then publishes $(g_1, g_2, u_2, v_2, w_2)$ as its public-key. `Auth` also publishes randomly selects $g, h \in \mathbb{G}_2$ and publishes them as public-key.

Second, a user $\mathcal{U}$ randomly selects its secret-key $q \in \mathbb{Z}_p^*$, 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.

Finally, an opener O randomly selects $\xi_1, \xi_2 \in \mathbb{Z}_p^*$ as its secret-key and computes $U \leftarrow g_2^{\xi_1}$, $V \leftarrow g_2^{\xi_2}$. O also publishes (U, V) as its public-key.

4.2 Credential Issuing Protocol

First, a user $\mathcal{U}$ creates signature sig_U on g_2^q using sk_U. $\mathcal{U}$ then sends g_2^q, sig_U, 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 by using pk_U, then signs m 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 $\mathcal{U}$. The tuple is received by $\mathcal{U}$ as its credential $Cred$ corresponding to m.

$\mathcal{U}$ then verifies whether the issued credential 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 $e(\sigma, \alpha) \stackrel{?}{=} e(g_1, \beta)$. `Auth` writes $\left(\sigma, r, s, m, g_2^q, sig_U\right)$ in database DB whenever `Auth` engages in the credential issuing protocol with users.

4.3 Credential Proving Protocol

After getting its credential, $\mathcal{U}$ proves the 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 a $\mathcal{V}$'s current black-list of users who acted something wrong (`Auth` can write and read, and $\mathcal{V}$ can only read `BL`), where $b_i\,(1 \le i \le l) \leftarrow g_2^{q_i}$($q_i$ is the i-th black-listed 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, θ, $\rho \in \mathbb{Z}_p^*$, $f, \hat{f}$ from $\mathbb{G}_1$, 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 $i\,(1 \le i \le l)$.

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 the knowledge for the following statement: $PK\{(q, \rho, \theta, r\theta, s\theta, t_1, t_2) : \chi = f^q \hat{f}^\rho, \alpha' = w_2^\theta g_2^{r\theta}, \beta' = \left(g_2^m\right)^\theta g_2^{q\theta} u_2^\theta v_2^{s\theta} \alpha'^{t_1+t_2}, d_1 = \psi(U)^{t_1}, d_2 = \psi(V)^{t_2}\}$. We detail this proof of the knowledge in **Figure.1**.

Furthermore, $\mathcal{U}$ has to prove that $\theta \neq 0$ to $\mathcal{V}$. Therefore, $\mathcal{U}$ proves the knowledge for the following statement: $PK\{(\theta, r\theta) : \alpha' = w_2^\theta g_2^{r\theta}, \theta \neq 0\}$. Details of this proof of the knowledge are the same in **Figure.1**.

Step4: If all verifications in **step.2** holds and the proof of knowledge is accepted, $\mathcal{V}$ finally outputs `accept`, otherwise outputs `reject`. Because blacklisted users cannot both satisfy the latter verification in **step.2** and succeed in the proof of knowledge in **Figure.1**, this protocol provides the former way of credential revoking function described in Section 2.3.1.

Figure.1

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

Common input: $\left(\chi, f, \hat{f}, \alpha', \beta', d_1, d_2\right)$ 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 a random number $b, \lambda \in \mathbb{Z}_p^*$ and computes $z \leftarrow g^b h^\lambda$(commitment of b) and sends z to $\mathcal{U}$.
Step2: $\mathcal{U}$ randomly selects R_1, R_2, R_3, R_4, R_5, R_6, and R_7 from $\mathbb{Z}_p^*$, and computes $A \leftarrow f^{R_1} \hat{f}^{R_2}, B \leftarrow w_2^{R_3} g_2^{R_4}, C \leftarrow \left(g_2^m\right)^{R_3} u_2^{R_3} v_2^{R_5} \alpha'^{R_6+R_7}, D \leftarrow \psi(U)^{R_6}, E \leftarrow \psi(V)^{R_7}, F \leftarrow g_2^{R_1}, G \leftarrow g_2^{R_3}, H \leftarrow g_2^{R_1 R_3}$ and sends (A, B, C, D, E, F, G, H) to $\mathcal{V}$.
Step3: $\mathcal{V}$ sends b, λ to $\mathcal{U}$ in order to open the commitment.
Step4: $\mathcal{U}$ sends $(Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7)$ to $\mathcal{V}$ such that $Q_1 \leftarrow R_1 + bq \bmod p, Q_2 \leftarrow R_2 + b\rho \bmod p, Q_3 \leftarrow R_3 + b\theta \bmod p, Q_4 \leftarrow R_4 + b(r\theta) \bmod p, Q_5 \leftarrow R_5 + b(s\theta) \bmod p, Q_6 \leftarrow R_6 + bt_1 \bmod p, Q_7 \leftarrow R_7 + bt_2 \bmod p$.
Step5: $\mathcal{V}$ checks that $f^{Q_1} \hat{f}^{Q_2} \stackrel{?}{=} A\chi^b, w_2^{Q_3} g_2^{Q_4} \stackrel{?}{=} B\alpha'^b, \left(g_2^m\right)^{Q_3} u_2^{Q_3} v_2^{Q_5} \alpha'^{Q_6+Q_7} g_2^{(Q_1 Q_3/b)} H^{1/b} F^{-Q_3/b} G^{-Q_1/b} \stackrel{?}{=} C(\beta')^b$,
$\psi(U)^{Q_6} \stackrel{?}{=} D d_1^b, \psi(V)^{Q_7} \stackrel{?}{=} E d_2^b$.

4.4 Credential Revealing Protocol

If verifier $\mathcal{V}$ finds that a user has misused its 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 sends σ to $\mathcal{V}$.
Step3: O proves the 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 the knowledge in **Figure.4**.

Step4: `Auth` finds $\left(r, s, m, g_2^q, sig_U\right)$ in `DB`(they are related to σ) and sends them to $\mathcal{V}$.

$\mathcal{V}$ 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 is a valid signature on ${g_2}^q$. This protocol provides the latter way of credential revoking function described in Section 2.3.1.

Figure.2

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

Common input: Public key and $(d_1, d_2, \sigma, \sigma')$
Prover's input: (ξ_1, ξ_2)
Protocol:
Step1: $\mathcal{O}$ picks random numbers R_1, $R_2 \in \mathbb{Z}_p^*$, 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 $b \in \mathbb{Z}_p^*$ to `Auth`.
Step3: `Auth` 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`.

5 Security

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

5.1 Unforgeability

Theorem.2 If the basic signature scheme is $(q_{Auth}, \tau, \epsilon)$-strongly existentially unforgeable against chosen message attacks, then our proposed anonymous credential system is $\left(\tau', q'_{Auth}, \epsilon'\right)$-unforgeable, provided that
$\frac{1}{16}\left(1 - 2e^{\frac{\epsilon'}{-2(1-\epsilon')}n}\right)^4 \left(1 - 2e^{\frac{\epsilon'}{-2(2-\epsilon')}n}\right)^4 \geq \epsilon$,
$2\tau'' + \Theta(T) \leq \tau, q'_{Auth} \leq q_{Auth}$.

Sketch of Proof: Let us assume that our system is $(\tau', q_{Auth}, \epsilon')$-forgeable. Thus, $\mathcal{U}$ can forge $(\sigma', \alpha', \beta', d_1, d_2)$ that satisfies a verifier $\mathcal{V}$'s equation in the credential proving protocol with $(\tau', q_{Auth}, \epsilon')$. We then construct an extractor $\mathcal{E}$ which outputs the original credential (σ, r, s) and U, V. The advantage is estimated above by using heavy-law lemmma and Chernoff bound.

5.2 Anonymity and Unlinkability

Theorem.3 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 $\epsilon' \geq 2\epsilon, \tau' \leq \tau$

Sketch of 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 $(U, V, g_2, U^{t_1}, V^{t_2}, \eta)$ of the Decision Linear Problem. Then $\mathcal{A}$ acts two users $\mathcal{U}_0$ and $\mathcal{U}_1$, and plays the game described in Section 2.3.2 with `Adv`.

Finally, `Adv` outputs bit d'. If $d' = d$, $\mathcal{A}$ outputs `yes`(guesses $\eta = g_2^{t_1+t_2}$). Else(if $l' \neq l$), $\mathcal{A}$ outputs `no`. $Pr[\mathcal{A}\left(U, V, g_2, U^{t_1}, V^{t_2}, g_2^{t_1+t_2}\right) =$ `yes` $: U, V, g_2, \stackrel{R}{\leftarrow} \mathbb{G}_1, t_1, t_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*]$ is at least $\frac{1}{2} \cdot \left(\frac{1}{2} + \epsilon'\right)$.
$Pr[\mathcal{A}(U, V, g_2, U^{t_1}, V^{t_2}, \eta) =$ `yes` $: U, V, g_2, \eta \stackrel{R}{\leftarrow} \mathbb{G}_1, t_1, t_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*]$ is $\frac{1}{2} \cdot \frac{1}{2}$.

Thus, $\mathtt{Adv}_{Linear}$ is at least $\frac{\epsilon'}{2}$.

5.3 Traceability

Theorem.4 If the basic signature scheme is $(q_{Auth}, \tau, \epsilon)$-strongly existentially unforgeable against chosen message attacks, then our proposed anonymous credential system is $\left(\tau', q'_{Auth}, \epsilon'\right)$-traceable, provided that $\frac{1}{8}\left(1 - 2e^{\frac{\epsilon'}{-2(1-\epsilon')}n}\right)^3 \left(1 - 2e^{\frac{\epsilon'}{-2(2-\epsilon')}n}\right)^3 \geq \epsilon, 2\tau'' + \Theta(T) \leq \tau$

Sketch of Proof: Assume `Adv` is an adversary that $\left(\tau', q'_{Auth}, \epsilon'\right)$-breaks the traceabiblity 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`.

$\mathcal{E}$ can extract (σ, r, s) in the same way as the proof of **Unforgeability**, and the tuple does not correspond to that of any user in the authority's `DB`. Therefore, (σ, r, s) is the forgery of the basic signature scheme.

5.4 Non-frameability

Theorem.5 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
$\frac{1}{16}\left(1 - 2e^{\frac{\epsilon''}{-2(1-\epsilon'')}n}\right)^4 \left(1 - 2e^{\frac{\epsilon''}{-2(2-\epsilon'')}n}\right)^4 \geq \epsilon'$,
$\epsilon'' \geq \epsilon, min\left(\frac{\tau' - \Theta(T)}{2}, \tau\right) \geq \tau''$.

Sketch of 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 random public-key pk_U of the user's signature scheme and parameters $g_2, g_2^q \in \mathbb{G}_2$ of the discrete logarithm problem. It generates the components of the credential public key, the authority's key, the opener's key,i.e., picks random $x, y, z, \xi_1, \xi_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $g_1 \leftarrow \psi(g_2)$, $w_2 \leftarrow g_2^x, u_2 \leftarrow g_2^y, v \leftarrow g_2^z, U \leftarrow g_2^{\xi_1}, V \leftarrow g_2^{\xi_2}$. It then provides to `Adv` the credential public-key $(g_1, g_2, w_2, u_2, v_2, U, V)$, the authority's secret-key x, and the opener's secret-key (ξ_1, ξ_2).

Table 1: Comparison

	CL01[3]	Our proposed system with revocation
Assumption	strong RSA, DDH	SDH
Size of pk	10 elements	8 elements
Size of sk	7 elements	5 elements
Size of Cred	3 elements	3 elements
Size of proof	9 elements	42 elements
Size of open	15 elements	15 elements
Operations to Issue	1 exp (creating pseudonym 8 exp, PoK of 12 values)	4 exp
Operations to Verify	1 exp	1 pairing + 2 exp
Operations to Prove	9 exp	20exp, $l+1$ pairings (l:number of black-listed user)
Operations to Reveal	14 exp	12 exp, 1 pairing

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

If sig_U is a forgery, the theorem is proven and advantage $\epsilon'' \geq \epsilon$. If sig_U is not a forgery, g_2^q is surely the value sent by $\mathcal{U}$ to the Adv. It is then possible for $\mathcal{A}$ to let Adv execute the credential proving knowledge twice and extract q in the same way as the proof of **Unforgeability** of our proposed anonymous credential systems, with the maximum time $\tau' \geq 2n\tau'' + \Theta(T)$ and the advantage

$$\frac{1}{16}\left(1-2e^{\frac{\epsilon''}{-2(1-\epsilon'')}n}\right)^4\left(1-2e^{\frac{\epsilon''}{-2(2-\epsilon'')}n}\right)^4 \geq \epsilon'.$$

5.5 Comparison

We turn now to the efficiency of our anonymous credential system. We show a comparison of our system with revocation and the existing system[3] in Table.1. "Size of proof" means the total numbers of data that $\mathcal{U}$ and $\mathcal{V}$ transmit to each other in the credential proving protocol. "Operations to Verify" means the number of operations user needs to verify the credential received from an authority. "Operations to Prove" means the number of operations user needs to prove the possession of the credential to a verifier. "Size of open" means the total numbers of data which $\mathcal{V}$ and $\mathcal{O}$ transmit to each other in the credential revealing protocol. "Operations to Reveal" is also added from Table.1 in our system with revocation. Our system with revocation seems to be less efficient than the existing system with revocation because of providing two types of revocation.

6 Conclusion

We presents the anonymous credential system with revocation. The system is secure under the standard model. It seems less efficient than the existing system[4](See Table.1), but we provide two ways of revocation: Rejecting black-listed users and Revealing the user's identity who acted wrong.

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. An efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. EUROCRYPT '01, LNCS, Vol.2045, pp.93–118, 2001.

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

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

[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] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC'06, LNCS, Vol.3876*, pp. 80–99.

[9] Patrick Tsang, Man Ho Au, Apu Kapadia, and Sean Smith. Blacklistable anonymous credentials: Blocking misbehaving users without ttps. Proceedings of CCS 2007.