Access with Fast Batch Verifiable Anonymous Credentials

Ke Zeng

NEC Laboratories, China
zengke@research.nec.com.cn

Abstract. An anonymous credential-based access control system allows the user to prove possession of credentials to a resource guard that enforce access policies on one or more resources, whereby interactions involving the same user are unlinkable by the resource guard. This paper proposes three fast batch verifiable anonymous credential schemes. With all three schemes, the user can arbitrarily choose a portion of his access rights to prove possession of credentials while the number of expensive cryptographic computations spent is independent of the number of accessx rights being chosen. Moreover, the third anonymous credential scheme is not only fast batch verifiable but also fast fine-grained revocable, which means that to verify whether an arbitrarily chosen subset of credentials is revoked entails constant computation cost.

Keywords: anonymous credential, batch verification, fine-grained revocation, pairing.

1 Introduction

The protection of consumer privacy in access-control-based applications is a challenge that can be postponed no longer [1]. Access control simply means the act of determining if a particular right, such as access to some resource, can be granted to the presenter of a particular credential [2]. The access control system that this paper describes has a pseudonym authority (PA), resource holder (RH), resource guard (RG), and user as types of players. The PA issues a pseudonym to the user. The RH manages resources and, by issuing credentials, grants resource access rights to the user. The RG enforces access policies on the resources of one or more RHs and, by verifying the pseudonym and the credentials of a user, admits or denies the user access to the resources according to the RG's access control policies.

It's interesting to note the common practice that the applications try to collect as much personal information as possible from users, due to the incentive to price discriminate [3]. Hence, the user will frequently confront challenges in deciding how many resources are accessible and how much privacy is compromised. The user's decision may vary from service to service, from time to time, and from person to person. It's thus important to devise efficient anonymous credential schemes that enable verifying an arbitrarily chosen set of user credentials such that only absolutely necessary access rights (that are sensitive user information) are exposed to the access-control-based applications, with as few as possible computational cost.

L. Chen, M.D. Ryan, and G. Wang (Eds.): ICICS 2008, LNCS 5308, pp. 65–80, 2008.

Camenisch et al. [4, p. 2] have proposed seven desirable properties of an anonymous credential system, i.e., existential unforgeability, unlinkability, traceability, separability for resource holders[1], revocability, multi-show/one-show credentials, non-transferability. In addition to these, this paper highlights four desirable properties: batch verifiable, fine-grained revocable, and their *fast* versions.

(i) Batch verifiable. A user can arbitrarily select a portion of his access rights and prove possession of credentials to the RG without exposing other access rights.

(ii) *Fast* batch verifiable. In addition to (i), the number of expensive cryptographic computations spent and the pieces of data generated by proving possession of credentials are independent of the number of access rights being chosen. Scalar multiplication, modular exponentiation, and pairing evaluation are in general considered expensive.

(iii) Fine-grained revocable. Any one credential of the user can be revoked while leaving his other credentials untouched. In other words, revocation of credentials is on a per-user per-access-right basis.

(iv) *Fast* fine-grained revocable. The RG may need to ascertain whether a subset of the user's credentials has been revoked. In such case, in addition to (iii), the number of expensive cryptographic computations spent and the pieces of data generated by proving that the subset of credentials are not revoked are independent of the size of the subset being chosen.

Our Contribution

To the best of our knowledge, this work is the first that addresses an anonymous credential system achieving the *fast* batch verifiable property and the *fast* fine-grained revocable property. Three pairing-based anonymous credential schemes are presented. The first scheme achieves the *fast* batch verifiable property in the random oracle model. The second and third schemes achieve the *fas*t batch verifiable property in the standard model. In particular, the third scheme achieves the *fast* fine-grained revocable property as well.

2 Related Work

The anonymous credential system has been extensively studied in academia, resulting in many schemes, including those of [1, 4 - 14], just to name a few.

Chaum et al. [5] first introduced the scenario with multiple users that request credentials from resource holders then anonymously present credentials to resource guards without involving the resource holders online. The schemes proposed in [6] and [7] are based on having a trusted third party involved in all interactions.

Persiano et al. [10,11] proposed two anonymous credential schemes. The work of [10] is based on a chameleon certificate. The work of [11] is based on Strong RSA assumption. However, these schemes rely on inefficient zero-knowledge proofs, such

[1] Here it should be noted that the term Organization is originally utilized in [4]. The Organization not only issues credentials but also verifies credentials. In this regard, we logically divide Organization into Resource Holder and Resource Guard.

as proving knowledge of a double discrete logarithm, which is too expensive to be adoptable in practice [15].

Camenisch et al. [4, 12] proposed two efficient anonymous credential schemes, wherein [4] is based on Strong RSA assumption and [12] is based on LRSW assumption.

Most recently, Akagi et al. 14 proposed a q-SDH assumption-based anonymous credential scheme. This scheme is more efficient than the above ones because it utilizes a simplified system model (we will elaborate this simplified system model in Section 5.1).

However, none of the previous work presents a fast batch verifiable anonymous credential scheme. As Camenisch et al. recently pointed out, it is as yet an open problem to find a fast batch verification scheme for anonymous credentials [16].

Moreover, none of the previous work explicitly addresses the fine-grained revocable property. And none presents a fast fine-grained revocable anonymous credential scheme.

3 Preliminaries

3.1 Notations and Number-Theoretic Preliminaries

If $\mathcal{S}$ is a finite set, $x \in_R \mathcal{S}$ denotes that x is chosen from $\mathcal{S}$ uniformly at random. Let $\Omega(\cdot)$ be an arbitrary Boolean predicate, i.e., a function that, upon input of some string ς, outputs either TRUE or FALSE. By $\varsigma \leftarrow A(x) : \Omega(\varsigma)$ we denote that $\Omega(\varsigma)$ is TRUE after ς was obtained by running algorithm $A(\cdot)$ on input x. A function $adv(k)$ is said to be negligible if for every positive polynomial $p(\cdot)$ and sufficiently large k, $adv(k) < 1 / p(k)$.

Throughout this paper, we use the traditional multiplicative group notation, instead of the additive notation often used in elliptic curve settings.

Let $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ be two finite cyclic groups with additional group $\mathcal{G} = \langle g \rangle$ such that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathcal{G}| = p$ where p is a large prime. Let $\mathbb{G}_1^*$ denote $\mathbb{G}_1 \setminus \mathcal{O}$ where $\mathcal{O}$ is the identity of $\mathbb{G}_1$. Bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathcal{G}$ is a function, such that: Bilinear, for all $h_1 \in \mathbb{G}_1$, $h_2 \in \mathbb{G}_2$, and for all $a, b \in \mathbb{Z}_p$, $e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$; Non-degenerate, $\exists h_1 \in \mathbb{G}_1$, $\exists h_2 \in \mathbb{G}_2$ such that $e(h_1, h_2) \neq \mathcal{I}$ where $\mathcal{I}$ is the identity element of $\mathcal{G}$; and Computable: there exists an efficient algorithm for computing e.

We suppose there is a setup algorithm $Setup(\cdot)$ that, upon input of security parameter 1^k, outputs the above settings of the bilinear map and writes this as $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k)$.

q-SDH Assumption. For all probabilistic polynomial-time (p.p.t.) adversaries $\mathcal{A}$, $adv(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \in_R \mathbb{Z}_p;$$
$$\Pr\left[(x,y) \leftarrow \mathcal{A}(g_2{}^a, g_2{}^{a^2}, \cdots, g_2{}^{a^q}) : x \in \mathbb{Z}_p \wedge y = g_1{}^{1/(a+x)}\right] = adv(k)$$

The q-SDH assumption has been shown to hold in generic bilinear groups by Boneh et al. [17].

3.2 Honest-Verifier Zero-Knowledge (HVZK) Proof

Let $KP\left\{(\varsigma): \ \Omega(\varsigma)\right\}$ denote a zero-knowledge proof instance between a prover and a verifier, where their common input is a predicate $\Omega(\cdot)$, and the prover's secret input is a string ς. $KP\left\{(\varsigma): \ \Omega(\varsigma)\right\} = \text{TRUE}$ denotes the case that the verifier is convinced that $\Omega(\varsigma){=}\text{TRUE}$, and $KP\left\{(\varsigma): \ \Omega(\varsigma)\right\} = \text{FALSE}$ denotes the case otherwise. The honest-verifier zero-knowledge (HVZK) proof has been extensively studied during the past two decades, resulting in many efficient techniques [19 - 21].

In particular, we elaborate $KP\left\{(t,\tau,z): \ e(\mathrm{T}, A) = e(h^\tau \cdot \mathrm{T}^{-z}, g_2) \wedge \mathrm{T} = t^\tau\right\}$ as below Protocol 1, which is an HVZK proof of knowledge of $t \in \mathbb{G}_1$, $\tau \in \mathbb{Z}_p$, and $z \in \mathrm{Z} \subseteq \mathbb{Z}_p$ such that congruence $e(t^\tau, A \cdot g_2{}^z) = e(h^\tau, g_2)$ holds..

Protocol 1

Bilinear map $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k)$ and $A \in \mathbb{G}_2$ are system parameters. All these plus $h \in \mathbb{G}_1$ and $\mathrm{T} = t^\tau \in \mathbb{G}_1$ are common inputs to the prover and verifier.

The prover's secret input is $t \in \mathbb{G}_1$, $\tau \in \mathbb{Z}_p$, and $z \in \mathrm{Z} \subseteq \mathbb{Z}_p$.

The goal of Protocol 1 is to prove knowledge of t, τ, and z, such that $e(\mathrm{T}, A) = e(h^\tau \cdot \mathrm{T}^{-z}, g_2)$ and $\mathrm{T} = t^\tau$, i.e., $e(t^\tau, A \cdot g_2{}^z) = e(h^\tau, g_2)$.

1) The prover selects $(\alpha, \beta) \in_R \mathbb{Z}_p{}^2$, computes $\mathrm{R} = e(h^\alpha \cdot \mathrm{T}^{-\beta}, g_2) \in \mathcal{G}$, and sends R to the verifier.
2) The verifier selects a challenge $c \in_R \mathbb{Z}_p$ and sends c to the prover.
3) The prover computes $(s_\tau = \alpha - c \cdot \tau, s_z = \beta - c \cdot z) \in \mathbb{Z}_p{}^2$, and sends (s_τ, s_z) to the verifier.

The verifier is convinced iff $\mathrm{T} \in \mathbb{G}_1^*$ and $\mathrm{R} = e(h^{s_\tau} \cdot \mathrm{T}^{-s_z}, g_2) \cdot e(\mathrm{T}, A)^c$.

Also notice that Protocol 1 is a secure three-move identification scheme [22].

Claim 1. The number of expensive cryptographic computations spent and the pieces of data generated by Protocol 1 are constant.

4 Fast Batch Verifiable Anonymous Credential

Describing anonymous credential scheme as five algorithms: $PAKeyGen(\cdot)$, $RHKeyGen(\cdot)$, $UserEnroll(\cdot)$, $GCred(\cdot)$, and $VCred(\cdot)$, we formalize the notions of batch verification and fast batch verification.

$PAKeyGen(\cdot)$. This probabilistic algorithm takes as input the security parameter 1^k, and returns the PA public key pk^{PA} and private key sk^{PA}.

$RHKeyGen(\cdot)$. This probabilistic algorithm takes as input pk^{PA} and access right R_{ij} that is under control of RH_i, and returns access-right public key pk_{ij}^{RH} and private key sk_{ij}^{RH} of access right R_{ij}.

$UserEnroll(\cdot)$. This probabilistic algorithm takes as input pk^{PA}; sk^{PA}; $n_{\max}$, which is the maximum number of admissible users; and a user identity U_l; and returns user key x_l and user pseudonym nym_l.

$GCred(\cdot)$. The credential issuance algorithm takes as input pk^{PA}, nym_l, x_l, access right R_{ij}, corresponding access-right public key pk_{ij}^{RH} and private key sk_{ij}^{RH}, returns credential $Cred_{ijl}$.

$VCred(\cdot)$. The credential verification algorithm takes as input pk^{PA}, nym_l, x_l, a set of access rights $\{R_{ij}\}$, corresponding access-right public keys $\{pk_{ij}^{RH}\}$, and purported credentials $\{Cred_{ijl}\}$. It decides whether to accept or to reject the credentials, and returns TRUE or FALSE, respectively.

Definition (Batch Verifiable Anonymous Credentials). An HVZK knowledge proof instance $KP\left\{(nym_l, x_l, \{Cred_{ijl}\}) : VCred(pk^{PA}, nym_l, x_l, \{R_{ij}\}, \{pk_{ij}^{RH}\}, \{Cred_{ijl}\})\right\}$ is a batch verification of anonymous credentials if the following two conditions hold:

- If for all R_{ij}, $VCred(pk^{PA}, nym_l, x_l, R_{ij}, pk_{ij}^{RH}, Cred_{ijl}) = \mathrm{TRUE}$, then given the honest prover $KP\left\{\begin{matrix}(nym_l, x_l, \{Cred_{ijl}\}) : \\ VCred(pk^{PA}, nym_l, x_l, \{R_{ij}\}, \{pk_{ij}^{RH}\}, \{Cred_{ijl}\})\end{matrix}\right\} = \mathrm{TRUE}$
- If for any R_{ij}, $VCred(pk^{PA}, nym_l, x_l, R_{ij}, pk_{ij}^{RH}, Cred_{ijl}) = \mathrm{FALSE}$, then $adv(k)$ defined as follows is a negligible function:

$$\Pr\left[KP\left\{\begin{matrix}(nym_l, x_l, \{Cred_{ijl}\}) : \\ VCred(pk^{PA}, nym_l, x_l, \{R_{ij}\}, \{pk_{ij}^{RH}\}, \{Cred_{ijl}\})\end{matrix}\right\} = \mathrm{TRUE}\right] = adv(k)$$

With this definition, it's easy to see that most existing anonymous credential schemes can conduct batch verification of anonymous credentials, e.g. the scheme as per [9]. Specifically for some of the existing schemes, adopting batch verification method for modular exponentiations due to Bellare et al. [23] may yield alternative approaches, see for instance [24] which presented general approach to batch verification of short signatures. Whereas, neither approach is *fast*, as analyzed below, as per our definition for fast batch verification of anonymous credentials.

Definition (Fast Batch Verifiable Anonymous Credentials). The above batch verification of anonymous credentials is said to be fast if, for all $\tilde{\mathcal{R}} \subseteq \left\{R_{ij}\right\}$, the number of expensive cryptographic computations spent and the pieces of data generated by

$$KP\left\{(nym_l, x_l, \left\{Cred_{ijl}\right\}) : VCred(pk^{PA}, nym_l, x_l, \tilde{\mathcal{R}}, \left\{pk_{ij}^{RH}\right\}, \left\{Cred_{ijl}\right\})\right\} = \mathrm{TRUE}$$

are independent of $\left|\tilde{\mathcal{R}}\right|$.

5 Fast Batch Verifiable Anonymous Credential Scheme

5.1 Scheme I

Our first fast batch verifiable anonymous credential scheme (Scheme I) is based on a simplified system model. This simplified system model has four types of players: the portal service (PS) that not only manages pseudonyms but also manages access rights on behalf of the RHs, the RH that is transparent to the user, the RG, and the user. It's notable that the work of [14] is based on the same simplified system model.

Notice that with this simplified system model, algorithm $RHKeyGen(\cdot)$ merges with algorithm $PAKeyGen(\cdot)$ as $PSKeyGen(\cdot)$, and algorithm $UserEnroll(\cdot)$ merges with algorithm $GCred(\cdot)$.

***PSKeyGen(·)*:** The PS calls $Setup(\cdot)$ according to the security parameter 1^k, chooses a full-domain hash function $Hash(\cdot) : \left\{0,1\right\}^* \to \mathbb{G}_1$ that is viewed as a random oracle by the security analysis; and chooses $a \in_R \mathbb{Z}_p$; and computes $A = g_2^{\ a} \in \mathbb{G}_2$.

The PS's public key is $pk^{PS} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, A, Hash(\cdot))$ and private key is $sk^{PS} = a$.

***GCred(·)*:** In order to be granted access rights R_i by the PS, a user who has trustworthy identity U_a carries out the following access-rights-granting protocol with the PS:

2.a) The user sends its identity U_a to the PS and the PS queries its database for stored key z of U_a. Iff it does not find a match, the PS selects $z \in_R \mathbb{Z}_p$ for U_a and stores (U_a, z) in its database.

2.b) The PS computes $t_i = Hash(R_i)^{1/(a+z)}$, and sends user key z and credential t_i to the user.

2.c) User U_a verifies that $e(t_i, A \cdot g_2^{\ z}) = e(Hash(R_i), g_2)$.

Fast Batch Verification of Anonymous Credentials: Suppose the user U_a has been granted access rights $\left\{R_1, R_2, \ldots, R_\Upsilon\right\}$. Without loss of generality, suppose one subset $\left\{r_j\right\} \subseteq \left\{R_1, R_2, \ldots, R_\Upsilon\right\}$ of the user's access rights matches one of the RG's policies $Pol = \left(\left\{r_j\right\}, \mathrm{H} = \prod_j Hash(r_j), \ldots\right)$. Let t_j denote the user's credential that corresponds to access right r_j.

Define $VCred(\cdot)$ as $e(\prod_j t_j, A \cdot g_2^{\,z}) \overset{?}{=} e(\prod_j Hash(r_j), g_2)$. In order to prove to the RG that he meets policy Pol, the user carries out the following anonymous access control protocol with the RG:

3.a) The user U_a computes $\mathrm{t} = \prod_j t_j$, selects $\tau \in_R \mathbb{Z}_p$, and computes batch verifiable anonymous credential $\mathrm{T} = \mathrm{t}^\tau$.

3.b) User U_a notifies the RG of the matching of policy Pol and sends T to the RG.

3.c) RG interacts with U_a for $KP\left\{(\mathrm{t}, \tau, z):\ e(\mathrm{T}, A) = e(\mathrm{H}^\tau \cdot \mathrm{T}^{-z}, g_2) \wedge \mathrm{T} = \mathrm{t}^\tau\right\}$ utilizing Protocol 1.

3.d) If $KP\left\{(t, \tau, z):\ e(\mathrm{T}, A) = e(\mathrm{H}^\tau \cdot \mathrm{T}^{-z}, g_2) \wedge \mathrm{T} = \mathrm{t}^\tau\right\} = \mathrm{TRUE}$, the RG is convinced.

5.1.1 Scheme I Security

We formalize the existential unforgeability of Scheme I as an adaptive chosen-key and adaptive chosen-access-right game. In this model, the adversary $\mathcal{A}$ is given a single public key. His goal is the existential forgery of a batch verifiable anonymous credential. The adversary's advantage, $\mathrm{ADV}_{\mathcal{A}}^{SchemeI}$, is defined as his probability of success in the game.

Definition (Existential Unforgeability of Scheme I). An adversary $\mathcal{A}$ $(N, t, \mathrm{K}, \Gamma, \varepsilon)$-breaks the existential unforgeability of the N-user Scheme I in the adaptive chosen-key and adaptive chosen-access-right model if $\mathcal{A}$ runs in time at most t, makes at most K queries to the hash function, issues at most Γ credential queries to the challenger, and $\mathrm{ADV}_{\mathcal{A}}^{SchemeI}$ is at least ε.

Theorem 1.1. Proposed Scheme I is secure against existential forgery in the random oracle model under q-SDH assumption.

Corollary 1.1. Proposed Scheme I is a batch verification scheme for anonymous credentials.

Corollary 1.2. Based on Claim 1, proposed Scheme I is a *fast* batch verification scheme for anonymous credentials.

Now we turn to formalizing the unlinkability of Scheme I. We basically rephrase the CPA-full-anonymity model defined by Boneh et al. [25], which is a slightly weaker version of the full-anonymity model given by Bellare et al. [28]. In this model, the adversary $\mathcal{A}$ is given a single public key. His goal is to determine which of two users is involved in an instance of the anonymous access control protocol. His success probability, $\mathrm{SUCC}_{\mathcal{A}}^{SchemeI}$, is defined as his probability of success in the game.

Definition (CPA-Full-Anonymity of Scheme I). An adversary $\mathcal{A}$ $(N, t, \mathrm{K}, \Gamma, \varepsilon)$-breaks the CPA-full-anonymity of the N-user Scheme I if $\mathcal{A}$ runs in time at most t, makes at most K queries to the hash function, issues at most Γ credential queries to the challenger, and $\mathrm{SUCC}_{\mathcal{A}}^{SchemeI}$ is at least ε.

Definition (Security of Scheme I). Scheme I is secure if no algorithm $(N, t, \mathrm{K}, \Gamma, \varepsilon)$-breaks its existential unforgeability and no algorithm $(N, t, \mathrm{K}, \Gamma, \varepsilon)$-breaks its CPA-full-anonymity.

Lemma 1.3. Proposed Scheme I achieves CPA-full-anonymity.

Corollary 1.3. Based on Theorem 1.1 and Lemma 1.3, proposed Scheme I is secure.

5.2 Scheme II

Here we present our second fast batch verifiable anonymous credential scheme (Scheme II). Unlike Scheme I, Scheme II works in the general system model and the security of Scheme II does not rely on the random oracle.

***PAKeyGen(·)*:** The PA calls $Setup(\cdot)$ according to the security parameter 1^k and chooses $a \in_R \mathbb{Z}_p$ and computes $A = g_2^{\ a} \in \mathbb{G}_2$.

The PA's public key is $pk^{PA} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, A)$ and private key is $sk^{PA} = a$.

RHKeyGen(·): Given PA public key pk^{PA}, the resource holder RH_i that controls access rights R_{ij}, $j = 1, 2, \cdots, n_i$, executes the following:

2.a) For each R_{ij}, it chooses $b_{ij} \in_R \mathbb{Z}_p$ and computes $B_{ij} = g_2^{\ b_{ij}} \in \mathbb{G}_2$.

2.b) For each B_{ij}, it generates the signature of knowledge proof

$$\Sigma_{ij} = SKP\left\{(b_{ij}) : B_{ij} = g_2^{\ b_{ij}}\right\}.$$

The RH_i's access right public key is $pk_{ij}^{RH} = \left\{\left(R_{ij}, B_{ij}, \Sigma_{ij}\right)\right\}$ and the private key is $sk_{ij}^{RH} = b_{ij}$.

***UserEnroll(·)*:** In order to obtain a pseudonym, a user who has trustworthy identity U_u carries out the following with the PA:

3.a) The user sends its identity U_u to the PA and the PA queries its database for stored key z_u of U_u. Iff it does not find a match, the PA selects $z_u \in_R \mathbb{Z}_p$ for U_u and stores (U_u, z_u) in its database.

3.b) PA computes a BB signature [17] $t_u = g_1^{1/(a+z_u)}$, and sends user key z_u and pseudonym t_u to U_u.

3.c) User U_u verifies that $e(t_u, A \cdot g_2^{z}) = e(g_1, g_2)$.

GCred(·): In order to be granted a credential for access right R_{ij}, the user U_u carries out the following access-right-granting protocol with the resource holder RH_i that controls access right R_{ij}:

4.a) User U_u interacts with the RH_i for $KP\left\{(z):\ e(t_u, A) / e(g_1, g_2) = e(t_u^{-z}, g_2)\right\}$.

4.b) If $KP\left\{(z):\ e(t_u, A) / e(g_1, g_2) = e(t_u^{-z}, g_2)\right\} = \text{TRUE}$, the RH_i computes $v_{ij} = t_u^{b_{ij}}$ and sends v_{ij} to the user.

4.c) The user verifies that $e(t_u, B_{ij}) = e(v_{ij}, g_2)$ holds and stores v_{ij} as his credential for access right R_{ij}.

Fast Batch Verification of Anonymous Credentials: Suppose the user U_u has been granted pseudonym t_u and credentials for access rights $\left\{R_{ij}\right\}$ whose corresponding access-right public keys are $\left\{B_{ij}\right\}$. Without loss of generality, suppose one subset $\left\{r_j\right\} \subseteq \left\{R_{ij}\right\}$ of the user's access rights matches one of the RG's policies $Pol = \left(\left\{r_j\right\}, \mathrm{B} = \prod_j B_j, \ldots\right)$, where B_j is the access-right public key that corresponds to resource r_j. Since $r_j \in \left\{R_{ij}\right\}$, we have that $B_j \in \left\{B_{ij}\right\}$. Let v_j denote the user's credential that corresponds to access right r_j.

Define $VCred(\cdot)$ as: $e(t_u, A \cdot g_2^{z}) \overset{?}{=} e(g_1, g_2)$ and $e(t_u, B_j) \overset{?}{=} e(v_j, g_2)$. In order to prove to the RG that he meets policy Pol, the user U_u carries out the following anonymous access control protocol with the RG:

5.a) The user U_u selects $\tau \in_R \mathbb{Z}_p$, computes pseudonym $\mathrm{T} = t_u^{\tau}$ and batch verifiable anonymous credential $\mathrm{V} = \mathrm{v}^{\tau} = \left(\prod_j v_j\right)^{\tau}$.

5.b) User U_u notifies the RG of the matching of policy Pol and sends T, V to RG.

5.c) The RG interacts with the user U_a for

$$KP\left\{(t_u,\tau,z):\ e(\mathrm{T},A)=e(g_1^{\tau}\cdot\mathrm{T}^{-z},g_2)\wedge\mathrm{T}=t_u^{\tau}\right\}$$

utilizing Protocol 1.

5.d) If $KP\left\{(t_u,\tau,z):\ e(\mathrm{T},A)=e(g_1^{\tau}\cdot\mathrm{T}^{-z},g_2)\wedge\mathrm{T}=t_u^{\tau}\right\}=\mathrm{TRUE}$, the RG next verifies that $e(\mathrm{T},\mathrm{B})\overset{?}{=}e(\mathrm{V},g_2)$. If the congruence holds, the RG is convinced.

5.2.1 Scheme II Security

We formalize the existential unforgeability of Scheme II as an adaptive chosen-key and adaptive chosen-access-right game. In this model, the adversary $\mathcal{A}$ is given the PA public key and an access-right public key. His goal is the existential forgery of a pseudonym and a batch verifiable anonymous credential. The adversary's advantage, $\mathrm{ADV}_{\mathcal{A}}^{SchemeII}$, is defined as his probability of success in the game.

Definition (Existential Unforgeability of Scheme II). An adversary $\mathcal{A}$ (N,t,Γ,ε)-breaks the existential unforgeability of the N-user Scheme II in the adaptive chosen-key and adaptive chosen-access-right model if $\mathcal{A}$ runs in time at most t, issues at most Γ queries to the challenger, and $\mathrm{ADV}_{\mathcal{A}}^{SchemeII}$ is at least ε.

Theorem 2.1. Proposed Scheme II is secure against existential forgery under q-SDH assumption.

Corollary 2.1. Proposed Scheme II is a batch verification scheme for anonymous credentials.

Corollary 2.2. Based on Claim 1, proposed Scheme II is a *fast* batch verification scheme for anonymous credentials.

Now we turn to formalizing the unlinkability of Scheme II, again in the CPA-full-anonymity model. In this model, the adversary $\mathcal{A}$ is given the PA public key and some access-right public keys. His goal is to determine which of two users is involved in an instance of the anonymous access control protocol. His success probability, $\mathrm{SUCC}_{\mathcal{A}}^{SchemeII}$, is defined as his probability of success in the game.

Definition (CPA-Full-Anonymity of Scheme II). An adversary $\mathcal{A}$ (N,t,Γ,ε)-breaks the CPA-full-anonymity of the N-user Scheme II if $\mathcal{A}$ runs in time at most t, issues at most Γ queries to the challenger, and $\mathrm{SUCC}_{\mathcal{A}}^{SchemeII}$ is at least ε.

Definition (Security of Scheme II). Scheme II is secure if no algorithm (N,t,Γ,ε)-breaks its existential unforgeability and no algorithm (N,t,Γ,ε)-breaks its CPA-full-anonymity.

Lemma 2.3. Proposed Scheme II achieves CPA-full-anonymity.

Corollary 2.3. Based on Theorem 2.1 and Lemma 2.3, proposed Scheme II is secure.

6 Revocation

In this section, we will present a fast batch verifiable as well as fast fine-grained revocable scheme (Scheme III) that is modified from Scheme II.

6.1 Fast Fine-Grained Revocation

In addition to the five algorithms described in Section 4, Scheme III also requires the $Revoke(\cdot)$ algorithm below.

$Revoke(\cdot)$. This deterministic algorithm takes as input sk^{PA}, R_{ij}, pk_{ij}^{RH}, x_l, $Cred_{ijl}$, and $\tilde{x}$ for which the credential for R_{ij} needs revocation, and returns the updated $pk_{ij}^{RH\prime}$ and credential $Cred_{ijl}{}'$.

Definition (Fast Batch Verifiable and Fast Fine-grained Revocable Anonymous Credentials). A batch verifiable and fine-grained revocable anonymous credential scheme is said to achieve fast batch verification and fast fine-grained revocation properties if, for all $\hat{\mathcal{R}} \subseteq \tilde{\mathcal{R}} \subseteq \left\{R_{ij}\right\}$ where $\hat{\mathcal{R}}$ is the set of access rights whose corresponding credentials purportedly have not been revoked, the number of expensive cryptographic computations spent and the pieces of data generated by $KP\left\{(nym_l, x_l, \left\{Cred_{ijl}\right\}) : VCred(pk^{PA}, nym_l, x_l, \tilde{\mathcal{R}}, \left\{pk_{ij}^{RH}\right\}, \left\{Cred_{ijl}\right\})\right\} = \text{TRUE}$ are independent of $\left|\tilde{\mathcal{R}}\right|$ and $\left|\hat{\mathcal{R}}\right|$, where $VCred(\cdot)$ will return FALSE if any one credential for $\hat{\mathcal{R}}$ has been revoked.

In order to support fast fine-grained revocation, Scheme II's procedures for $RHKeyGen(\cdot)$, $GCred(\cdot)$, and anonymous access control protocol need to be slightly modified, as depicted below.

***RHKeyGen(·)*:** Given PA public key pk^{PA}, the resource holder RH_i that controls access rights R_{ij}, $j = 1, 2, \cdots, n_i$, executes the following:

2.a) For each R_{ij}, it chooses $b_{ij} \in_R \mathbb{Z}_p$ and computes $B_{ij} = g_2^{b_{ij}} \in \mathbb{G}_2$.

2.b) For each R_{ij}, it computes access right revocation data $h_{ij} = g_1^{b_{ij}} \in \mathbb{G}_1$ and initializes revocation list $RL_{ij} = \left\{(h_{ij}, \Delta)\right\}$, where Δ denotes that the two-tuple (h_{ij}, Δ) is the first row in RL_{ij}, i.e., no revocation happens yet.

2.c) For each B_{ij} and h_{ij}, it generates a signature of knowledge proof

$$\Sigma_{ij} = SKP\left\{(b_{ij}) : B_{ij} = g_2^{b_{ij}} \wedge h_{ij} = g_1^{b_{ij}}\right\}.$$

The RH_i 's public key is $pk_{ij}^{RH} = \left\{\left(R_{ij}, B_{ij}, \Sigma_{ij}, RL_{ij}\right)\right\}$ and the private key is $sk_{ij}^{RH} = b_{ij}$.

***GCred(·)*:** In order to be granted access right R_{ij}, the user U_u carries out the following access-right-granting protocol with the resource holder RH_i that controls access right R_{ij}:

4.a) User U_u interacts with the RH_i for $KP\left\{(z) :\ e(t_u, A) / e(g_1, g_2) = e(t_u^{-z}, g_2)\right\}$.

4.b) If $KP\left\{(z) :\ e(t_u, A) / e(g_1, g_2) = e(t_u^{-z}, g_2)\right\} = \text{TRUE}$, the RH_i computes $v_{ij} = t_u^{b_{ij}}$ and sends v_{ij} to the user.

4.c) The user U_u verifies that $e(t_u, B_{ij}) = e(v_{ij}, g_2)$ holds, and stores v_{ij} as his credential and $w_{ij} = v_{ij}$ as his validity data for access right R_{ij}.

***Revoke(·)*:** Given a misbehaving user $\tilde{U}$ that has user key $\tilde{z}$, in order to revoke his credential for access right R_{ij}, PA needs to do the following:

6.a) PA retrieves revocation data h_{ij} from the last (latest) row of RL_{ij}.

6.b) PA computes $\tilde{h}_{ij} = h_{ij}^{1/(a+\tilde{z})}$ and appends $(\tilde{h}_{ij}, \tilde{z})$ to RL_{ij}.

Consider user U_u that has user key z. User U_u has credential v_{ij} and validity data w_{ij} for access right R_{ij} as well. As a consequence of user $\tilde{U}$'s credential for access right R_{ij} being revoked, user U_u needs to execute the following to update his credential:

6.c) The user U_u computes $\tilde{w}_{ij} = (\tilde{h}_{ij} / w_{ij})^{1/(z-\tilde{z})}$ and updates his credential for access right R_{ij} to $(v_{ij}, \tilde{w}_{ij})$.

Fast Batch Verification & Fast Fine-Grained Revocation of Anonymous Credentials: Suppose the user U_u has been granted credentials for access rights $\left\{R_{ij}\right\}$ whose corresponding access-right public keys are $\left\{B_{ij}\right\}$. Without loss of generality, suppose one subset $\left\{r_j\right\} \subseteq \left\{R_{ij}\right\}$ of the user's access rights matches one of the RG's policies $Pol = \left(\left\{r_j\right\}, \mathrm{B} = \prod_j B_j, \ldots\right)$, where B_j is the access-right public key that corresponds to resource r_j. Since $r_j \in \left\{R_{ij}\right\}$, we have that $B_j \in \left\{B_{ij}\right\}$. Let v_{ij} denote the user's credential for credential access right r_j. Let w_{ij} denote the user's current validity data, i.e., data that has been updated to include the latest revocation, on access right r_j.

The user U_u wants to prove to the RG that he meets policy Pol. Whereas, the RG is curious about whether the user's access rights on $\{r_k\} \subseteq \{r_j\}$ have been revoked. Let h_k denote the access right revocation data that is retrieved from the last row of RL_k.

Define $VCred(\cdot)$ as $e(t_u, A \cdot g_2^{\,z}) \overset{?}{=} e(g_1, g_2)$, $e(t_u, B_j) \overset{?}{=} e(v_j, g_2)$, and $e(w_k, A \cdot g_2^{\,z}) \overset{?}{=} e(h_k, g_2)$. In order to convince the RG, the user U_u carries out the following anonymous access control protocol with the RG:

5.a) The user U_u selects $\tau \in_R \mathbb{Z}_p$, and computes pseudonym $\mathrm{T} = t_u^{\,\tau}$ and batch verifiable anonymous credential $\mathrm{V} = \mathrm{v}^{\tau} = \left(\prod_j v_j\right)^{\tau}$. In addition, the user computes fine-grained validity data $\mathrm{W} = \mathrm{w}^{\tau} = \left(\prod_k w_k\right)^{\tau}$.

5.b) The user U_u notifies the RG of the matching of policy Pol and sends T, V, and W to the RG.

5.c) Utilizing a natural extension of Protocol 1, the RG interacts with the user U_u for

$$KP\left\{\begin{array}{l}(t_u, \mathrm{w}, \tau, z): \;\; e(\mathrm{T}, A) = e(g_1^{\,\tau} \cdot \mathrm{T}^{-z}, g_2) \wedge \mathrm{T} = t_u^{\,\tau} \\ \qquad\qquad\qquad \wedge\, e(\mathrm{W}, A) = e((\prod_k h_k)^{\tau} \cdot \mathrm{W}^{-z}, g_2) \wedge \mathrm{W} = \mathrm{w}^{\tau}\end{array}\right\}.$$

5.d) If the RG accepts the above knowledge proof, it further verifies that $e(\mathrm{T}, \mathrm{B}) \overset{?}{=} e(\mathrm{V}, g_2)$. If the congruence holds, the RG is convinced.

6.1.1 Security of Fine-Grained Revocation

Note that Scheme III exactly reuses the steps in Scheme II to achieve fast batch verification of anonymous credentials and that Protocol 1 is witness indistinguishable [33]. Therefore, Scheme III should be secure with respect to existential unforgeability and CPA-full-anonymity, as long as the fine-grained validity data is existentially unforgeable.

We formalize the existential unforgeability of the validity data as an adaptive chosen-key and adaptive chosen-access-right game. In this model, the adversary $\mathcal{A}$ is given the PA public key, an access-right public key, and access right revocation data. His goal is the existential forgery of the validity data. The adversary's advantage, $\mathrm{ADV}_{\mathcal{A}}^{VALID}$, is defined as his probability of success in the game.

Definition (Existential Unforgeability). An adversary $\mathcal{A}$ $(N, t, \Gamma, \varepsilon)$-breaks the existential unforgeability of validity data of the N-user Scheme III in the adaptive chosen-key and adaptive chosen-access-right model if $\mathcal{A}$ runs in time at most t, issues at most Γ queries to the challenger, and $\mathrm{ADV}_{\mathcal{A}}^{VALID}$ is at least ε.

Theorem 3.1. Proposed Scheme III is secure against existential forgery of validity data under q-SDH assumption.

Corollary 3.1. Based on Claim 1, proposed Scheme III attains *fast* fine-grained revocation of anonymous credentials.

7 Conclusions and Future Work

Three constructions of *fast* batch verifiable anonymous credential schemes were presented. The first scheme achieves the *fast* batch verifiable property in the random oracle model. The second and third schemes achieve the *fast* batch verifiable property in the standard model. The third scheme in addition achieves the *fast* fine-grained revocable property.

To attain the *fast* properties, our three schemes require linear storage consumption for the credentials and the credentials cannot be arbitrary statements. It is desirable to see anonymous credential scheme with *fast* properties that overcomes these two limitations.

According to the findings of [24] and [34], our three schemes after further modifications may be able to conduct batch verification of anonymous credentials from different users. But such modifications cannot be *fast* if we require the number of expensive cryptographic computations being independent of the number of users. It is thus very interesting to find a *fast* scheme for this usage scenario.

Acknowledgements

The author is grateful to Tomoyuki Fujita, Min-Yu Hsueh, and Toshikazu Fukushima for pointing to this research and for support to this research. The author thanks Yu-Liang Zheng, Yong Ding, Hao Lei, Ye Tian, and Li-Ming Wang for valuable discussions during the development of the ideas herein presented. The author also thanks the anonymous reviewers, whose comments and suggestions help to improve the quality of this paper and stimulate new thoughts. The author is particularly grateful to one reviewer for pointing out a mistake in an early version of this paper.

References

1. Verheul, E.R.: Self-Blindable Credential Certificates from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 533–551. Springer, Heidelberg (2001)
2. Westhoff, D., Lamparter, B.: Charging Related Mobile Device Authentication. In: Advanced Internet Charging and QoS Technologies (ICQT 2001), pp. 129–135 (2001), ISBN 3-85403-157-2

3. Odlyzko, A.: Privacy, Economics, and Price Discrimination on the Internet. In: 5th International Conference on Electronic Commerce, pp. 355–366. ACM Press, New York (2003)
4. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
5. Chaum, D.: Security without Identification: Transaction Systems to Make Big Brother Obsolete. Communications of the ACM 28(10), 1030–1044 (1985)
6. Chaum, D., Evertst, J.H.: A Secure and Privacy-protecting Protocol for Transmitting Personal Information Between Organizations. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 118–167. Springer, Heidelberg (1987)
7. Chen, L.: Access with Pseudonyms. In: Dawson, E.P., Golić, J.D. (eds.) Cryptography: Policy and Algorithms 1995. LNCS, vol. 1029, pp. 232–243. Springer, Heidelberg (1996)
8. Lysyanskaya, A., Rivest, R., Sahai, A., Wolf, S.: Pseudonym Systems. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 184–199. Springer, Heidelberg (2000)
9. Holt, J.E., Seamons, K.E.: Selective Disclosure Credential Sets. Report 2002/151, Cryptology ePrint Archive (2002)
10. Persiano, P., Visconti, I.: An Anonymous Credential System and a Privacy-Aware PKI. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 27–38. Springer, Heidelberg (2003)
11. Persiano, P., Visconti, I.: An Efficient and Usable Multi-show Non-transferable Anonymous Credential System. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 196–221. Springer, Heidelberg (2004)
12. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
13. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: Non-Interactive Anonymous Credentials. Report 2007/384, Cryptology ePrint Archive (2007)
14. Akagi, N., Manabe, Y., Okamoto, T.: An Efficient Anonymous Credential System. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143. Springer, Heidelberg (2008)
15. Ateniese, G., Song, D., Tsudik, G.: Quasi-efficient Revocation of Group Signatures. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 183–197. Springer, Heidelberg (2003)
16. Camenisch, J., Hohenberger, S., Pedersen, M.Ø.: Batch Verification of Short Signatures. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 246–263. Springer, Heidelberg (2007)
17. Boneh, D., Boyen, X.: Short Signatures without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
18. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solution to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
19. Chaum, D., Evertse, J.H., Graaf, J.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 127–141. Springer, Heidelberg (1988)
20. Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
21. Qin, B., Wu, Q., Susilo, W., Mu, Y.: Group Decryption. Report 2007/017, Cryptology ePrint Archive (2007)

22. Okamoto, T.: Provable Secure and Practical Identification Schemes and Corresponding Signature Schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993)
23. Bellare, M., Garay, J.A., Rabin, T.: Fast Batch Verification for Modular Exponentiation and Digital Signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998)
24. Ferrara, A.L., Green, M., Hohenberger, S., Pedersen, M.O.: On the Practicality of Short Signature Batch Verification. Report 2008/015, Cryptology ePrint Archive (2008)
25. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
26. Miyaji, A., Nakabayashi, M., Takano, S.: New Explicit Conditions of Elliptic Curves for FR-reduction. IEICE Trans. Fundamentals E84-A(5), 1234–1243 (2001)
27. MIRACL, Multi-precision Integer and Rational Arithmetic C Library, http://www.shamus.ie
28. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003)
29. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
30. Camenisch, J., Lysyanskaya, A.: Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–77. Springer, Heidelberg (2002)
31. Nguyen, L.: Accumulators from Bilinear Pairings and Applications to ID-based Ring Signatures and Group Membership Revocation. Report 2005/123, Cryptology ePrint Archive (2005)
32. Ateniese, G., Camenisch, J., Hohenberger, S., Medeiros, B.: Practical Group Signatures without Random Oracles. Report 2005/385, Cryptology ePrint Archive (2005)
33. Feige, U., Shamir, A.: Witness Indistinguishable and Witness Hiding Protocols. In: 22nd ACM Symposium on Theory of Computing, pp. 416–426. ACM Press, New York (1990)
34. Peng, K., Boyd, C., Dawson, E.: Batch Zero-Knowledge Proof and Verification and Its Applications. ACM Transactions on Information and System Security, Article 6 10(2), 1–28 (2007)