11-2013

# Self-blindable credential: Towards anonymous entity authentication upon resource-constrained devices

Yanjiang YANG

Xuhua DING
*Singapore Management University*, xhding@smu.edu.sg

Haibing LU

Jian WENG

Jianying ZHOU

# Self-blindable Credential: Towards LightWeight Anonymous Entity Authentication

Yanjiang Yang*, Xuhua Ding†, Haibing Lu‡, Jian Weng♯

*Institute for Infocomm Research, Singapore
†Singapore Management University
‡Santa Clara University, USA
♯JiNan University, China

**Abstract.** We are witnessing the rapid expansion of smart devices in our daily life. The need for individual privacy protection calls for anonymous entity authentication techniques with affordable efficiency upon the resource-constrained smart devices. Towards this objective, in this paper we propose *self-blindable credential*, a lightweight anonymous entity authentication primitive. We provide a formulation of the primitive and present two concrete instantiations. The first scheme implements verifier-local revocation and the second scheme enhances the former with forward security. Our analytical performance results show that our schemes outperform relevant existing schemes.

## 1 Introduction

The recent advances of hardware technology such as mobile phones, embedded devices, and sensors, coupled with modern networking technology, set off an explosive growth of applications using digital credentials for authentication purposes. For instance, vehicles can be embedded with smart sensors carrying an electronic plate number issued from the DMV, such that, they can communicate with road-side devices for the road condition or traffic alerts. A mobile phone can carry its owner's personal credential, e.g., an electronic driver license or an e-identity, such that the owner can use the mobile phone to prove his identity or certain attribute, e.g., above 18 years.

Two security related issues emerge from the adoption of digital credentials. One is personal privacy protection. Users today are more concerned than ever about their personal information leakage. Unlike a physical credential such as a plastic driver license card which is presented by the user herself, digital credentials are often automatically applied in many applications, without the user's notice or explicit approval. This difference implies that digital credentials residing on smart devices may backfire on the user's personal privacy, despite its easiness of use.

The other issue is credential revocation. Different from their physical counterparts, digital credentials can be easily duplicated/stored by the users. This makes it a challenge for the authority to revoke a credential since it is the verifier's burden (not the credential holder's) to check the revocation status. This becomes even harder if the credentials are used in a privacy-preserving manner (to address the above privacy concern) where the verifier cannot link credentials with identities.

Moreover, these two issues are aggravated by the limited computation and communication capabilities of smart devices. The resource constraints of smart devices dictate that the solution for privacy preservation and revocation support should have lightweight computation and communication in order to minimize the time delay and energy consumption.

Among a myriad of privacy protection techniques in the literature, anonymous credentials [6, 7, 14, 16–19] address user privacy in entity authentication where credentials are used anonymously such that two authentication transactions using the same credential cannot be linked by the verifier and the public. Nonetheless, existing anonymous credential schemes are *not* suitable for smart devices due to high communication or computation overhead. The RSA-based schemes (e.g., [17, 18]) normally involve costly zero-knowledge range proofs, which entail tens of kilobits in communication and tens of modular exponentiation operations in an RSA group. Although the IDEMIX anonymous credential and the Direct Anonymous Attestation (both are RSA-based) have been implemented on JAVA card, e.g., [5, 9, 16, 29], the implementations either are not practically efficient enough or do not well address credential revocation. The state-of-the-art bilinear map-based schemes such as [7, 8, 19] rely on bilinear pairing operations or computations over large bilinear groups. The resulting

heavy computation and bulky data to communicate have a undesirable toll on performance and would drain the device battery power significantly.

The *self-blindable certificate* scheme in [31] is another primitive that can be used for anonymous entity authentication. The notion of self-blindable certificate considers a privacy-preserving variant of the conventional public key certificate: the certificate holder can *blind* the public key in the certificate by herself, such that multiple uses of the same certificate cannot be linked, while the validity of the CA signature is preserved. The scheme in [31] achieves this by adapting a digital signature scheme with certain homomorphic property. When used for entity authentication, a self-blindable certificate essentially functions as an anonymous credential, but with lighter computation. The high efficiency of the self-blindable certificate comes from the fact that the certificate holder only needs to perform computation in the compact base group of bilinear map. Nonetheless, despite of its high performance for smart devices [11], it does not provide a satisfactory credential revocation mechanism (see Section 2 for more details).

**Contributions**. Towards achieving anonymous entity authentication upon resource-constrained devices, we propose a new primitive called *self-blindable credential*. It follows the same working mechanism of self-blindable certificate, i.e., the computation at the user side works entirely in $G_1$ of an asymmetric bilinear map $e : G_1 \times G_2 \rightarrow G_T$, and the resulting authentication data only consist of group elements of $G_1$. As $G_1$ is much more compact than $G_2$ and $G_T$, and the computation in $G_1$ is orders of magnitude faster than in $G_2, G_T$ and pairing operation, the new primitive achieves better performance than existing anonymous credential schemes. In particular, our contributions consist of the following.

1. We formulate the notion of self-blindable credential, and the associated security requirements.
2. We present two concrete constructions: one is self-blindable credential with verifier-local revocation, and the other is forward secure self-blindable credential. The latter strengthens the former to achieve forward unlinkability, while the former has better performance.
3. We analyze the performance of our schemes, by comparing with the relevant existing schemes. The analytical performance results show that our constructions outperform relevant schemes, e.g., our second scheme saves 70% computation overhead and 50% communication overhead, over [3].

**Organization**. In Section 2, we review the related work. Formulation of self-blindable credential is presented in Section 3. Section 4 introduces preliminary knowledge. The two constructions together with security analysis are presented in Section 5 and 6, respectively. Section 7 provides analytical performance results.

## 2 Related Work

Anonymous credentials [6, 14, 16, 17, 19] are a cryptographic primitive allowing for entity authentication to proceed in an anonymous manner where unlinkability of credential uses is achieved. For some of the schemes in [6, 17], a credential can only be used once, and any reuse would be linkable. These one-use anonymous credentials have good performance, but the credential holders have to acquire new credentials from the credential issuer repetitively. $k$-TAA ($k$-Times Anonymous Authentication) [3, 30] improves one-use anonymous credential such that a credential can be used unlinkably for up to $k$ times.

Group signatures, e.g., [1, 4, 8, 12, 15, 20], are a primitive akin to anonymous credentials, allowing signatures to be generated in an unlinkable fashion. The difference between the two is that in group signatures, a designated authority can nullify the anonymity of the signatures and trace the actual signers. Ring signatures such as [28] are another primitive generating unlinkable signatures whose unlinkability is protected among a set of dynamically chosen users.

The self-blindable certificate scheme [31] can be viewed as lightweight anonymous credential by nature, and the light-weightiness comes from its distinctive working mechanisms introduced earlier. Despite its efficiency advantage, it lacks satisfactory credential revocation support. Two

revocation solutions are proposed in [31]. One is to enforce short-term certificates by embedding with each certificate an expiry date. However, there are several drawbacks with this approach: (1) it downgrades privacy protection, e.g., a unique expiry date identifies its bearer. In the general case, an expiry date links all certificates with the same expiry date, thus narrowing the space for linking a certificate; (2) it does not allow for immediate revocation, and a certificate can only be invalidated upon the expiry date, even if it needs to be revoked instantly. In addition, it has the problem of determining a proper life span for a certificate. The other revocation approach resembles Online Certificate Status Protocol (OCSP), where the certificate holder has to encrypt the blinding factors used in generating a blinded certificate to a trusted third party (TTP). The TTP helps the verifier determine the revocation status of the blinded certificate by de-blinding the blinded certificate. However, it is widely known that establishing an online trusted server implies a myriad of difficulties and drawbacks. Worse yet, it seems difficult to integrate more satisfactory revocation mechanisms with the scheme to achieve *revocable* self-blindable certificate. This somehow could be seen from the fact that the follow-up proposals such as [22, 25] are not secure (see Section 5.1). Our schemes presented in this work solve this problem[1].

Recently, structure preserving anonymous credentials (e.g., [2, 10]) have been proposed, making use of the non-interactive zero-knowledge proof systems [24]. An anonymous credential scheme is *structure preserving* if all of its public keys, messages, credentials, and authentication data (generated when showing a credential) are group elements of $G_1$ and $G_2$. Structure preserving anonymous credentials are intended to show credentials non-interactively while avoid the usual practice of the Fiat-Shamir heuristics [23]. In comparison, the authentication data generated from showing a self-blindable credential include only group elements of $G_1$, more promising from the efficiency perspective. The objective of our proposal of self-blindable credential is an anonymous authentication primitive efficient enough for resource-constrained devices, rather than strengthened security. Thus we still recur to the Fiat-Shamir heuristics to achieve no-interactive self-blindable credentials, and we believe that this is not an issue in practice if care is taken in implementing the corresponding cryptographic hash function.

## 3 Revocable Self-blindable Credential: Concept and Security Notions

**Problem Statement.** We consider a credential system that enables smart devices to authenticate themselves to a *device reader* anonymously, such that repeated uses of a credential cannot be linked by the reader. The primary issue to be addressed is that smart devices and the reader are asymmetric in terms of resources: smart devices are restraint with limited computation capability and energy resources, while the reader is relatively powerful and has no resource constraints. For the credential system to work under such a scenario, it should inflict upon the smart devices as less overhead as possible, and push heavy computations to the reader side. Ideally, a smart device performs lightweight computation and generate short authentication data, while the reader executes relatively heavy computation for verification. In the rest, we do not differentiate human users and their smart devices; "user" and "device" are used interchangeably, both referring to the client side of the authentication scheme.

**Self-blindable Credential and Security Notions.** We propose the concept of *self-blindable credential* as the solution to the asymmetric scenario stated above. A self-blindable credential system comprises a set of users, each obtaining a special credential called *self-blindable credential* from a credential issuer. These credentials are used for anonymous authentication to the verifiers, e.g., service providers, who trust the credential issuer. To use her self-blindable credential, a user generates a *blinded credential* by blinding it with random factors. The verifier checks the blinded credential to determine its validity. It is mandated that blinded credentials produced from the same self-blindable credential cannot be linked by the verifier or a third party. Our system supports credential revoca-

[1] A self-blindable credential can certainly be used as a self-blindable certificate: for each use, select a random public/private key pair, and then generate "zero-knowledge proof of credential" upon the public key in a non-interactive fashion by employing the Fiat-Shamir heuristics [23].

tion, such that the blinded credentials resulting from a revoked credential cannot be accepted by the verifier. Formally, a revocable self-blindable credential scheme is defined below.

**Definition 1.** *[Revocable Self-Blindable Credential] Let $\mathcal{C}$ denote the domain of the self-blindable credentials, and $\mathcal{BC}$ denote the collection of all blinded credentials. A revocable self-blindable credential scheme is composed of five algorithms {Setup, CredIssue, Blind, CredVerify, Revoke} as follows.*

**Setup**$(1^\kappa) \rightarrow (params, msk)$**:** *The setup algorithm takes as input a security parameter $1^\kappa$, and outputs a set of public system parameters params, and a master secret key msk. Below we assume that params is implicitly included as input to the rest four algorithms.*

**CredIssue**$(msk, u) \rightarrow c$**:** *The credential issuance algorithm takes as input msk and user identity $u$, and outputs a self-blindable credential $c \in \mathcal{C}$ for $u$.*

**Blind**$(c) \rightarrow bc$**:** *The blinding algorithm takes as input a self-blindable credential $c \in \mathcal{C}$ produced by CredIssue(), and outputs a blinded credential $bc \in \mathcal{BC}$. This is normally an interactive algorithm between a user and a verifier (who inputs a challenge message). Its non-interactive version can be derived by using the standard Fiat-Shamir Heuristics [23]. For the sake of generality, we omit the verifier's challenge which should be clear from the context.*

**Revoke**$(\lceil msk \rfloor, c) \rightarrow CRL \cup \{c\}$**:** *The credential revocation algorithm takes as input a self-blindable credential $c \in \mathcal{C}$ and optionally master secret key msk, and outputs the updated Credential Revocation List (CRL), which is initially empty.*

**CredVerify**$(bc, CRL) \rightarrow \{0, 1\}$**:** *The credential verification algorithm takes as input a blinded credential bc and CRL, and outputs either 1 (accept) or 0 (reject).*

*Correctness*. For all $c \leftarrow \mathsf{CredIssue}(msk, u)$, where $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa)$, it holds that $\mathsf{CredVerify}(\mathsf{Blind}(c), CRL) = 1$ on the condition that $c \notin CRL$.

*Security Notions*. We impose the following security requirements upon the revocable self-blindable credential scheme. Due to the length limit, the requirements are described informally below and the formulation is in Appendix A.

- **Unforgeability**. Credential unforgeability is a fundamental requirement for any credential system. For the self-blindable credential scheme, it mandates that an adversary cannot forge either self-blindable credentials or blinded credentials.
- **Unlinkability**. For privacy protection, blinded credentials must be unlinkable. In other words, it is not possible for the adversary (including the verifier) to determine whether two blinded credentials are produced from the same self-blindable credential.
- **Revocability**. The ability to revoke user credentials is crucial for practical credential systems. For example, if an employee resigns from a company, her previous privileges within the company must be revoked. For self-blindable credentials, revocability mandates that once a self-blindable credential is revoked, any resulting blinded credential must be rejected by the verifier.

In addition, possession of a credential represents the holder's access rights. Thus, sharing of a credential with other users who do not have the access rights should be disallowed. Ideally, a self-blindable credential scheme should deter a credential holder from sharing her credential with others. We call this property as *non-shareability*.

## 4 Preliminaries

**Notations.** Throughout the paper, we use $s \in_R S$ to denote that an element $s$ is randomly chosen from a set $S$. We will also use the following notations.

*Bilinear map*. Let $G_1, G_2, G_T$ be multiplicative cyclic groups of prime order $q$. A bilinear map $e : G_1 \times G_2 \rightarrow G_T$ has the following properties.

- Bilinear: $\forall u \in G_1, v \in G_2$ and $x, y \in_R Z_q$, $e(u^x, v^y) = e(u, v)^{xy} = e(u^y, v^x)$.

- Non-degenerate: let $g$ be a generator of $G_1$, and $h$ be a generator of $G_2$, $e(g,h) \neq 1$.

*Proof of Knowledge.* A Proof of Knowledge (PoK) protocol is a zero-knowledge two-party three-round protocol, whereby a prover proves to a verifier about the knowledge of a secret without disclosing any information on the secret. The three-round is "commit-challenge-response", which has been well studied in the literature. For simplicity, we use $PoK\{(x) : y = g^x\}$ to denote the zero-knowledge protocol that proves the knowledge of $x$ satisfying $y = g^x$. In fact, more complex relations can be proved, e.g., $PoK\{(x_1, x_2, ..., x_\ell) : y = g_1^{x_1} g_2^{x_2} ... g_l^{x_\ell}\}$, and $PoK\{(x) : y_1 = g_1^x \wedge y_2 = g_2^x\}$. A zero-knowledge proof of knowledge protocol can be made non-interactive by means of the Fiat-Shamir Heuristics [23], whereby the prover herself generates the challenge by applying the commitment to a collision-free hash function.

**Review of ASM Signatures.** Our constructions of self-blindable credential will be based on the ASM signature scheme [3][2]. A brief review of the ASM signature scheme is as follows. A signer's public key is $(Z = h^z, h \in G_2, a, b, d \in G_1)$ and the private key is $z \in Z_q^*$. An ASM signature upon message $m$ is defined as $(M, k, s)$ where $k, s \in_R Z_q$, and $M = (a^m b^s d)^{\frac{1}{k+z}} \in G_1$. The resulting signature can be verified by testing whether

$$e(M, Z \cdot h^k) = e(A, h)$$

where $A = a^m b^s d$.

The signature verification can also be conducted in a zero-knowledge proof of knowledge protocol that enables the holder of the signature to prove the possession of $(M, k, s, m)$ to a verifier, without revealing any information on the signature. The protocol is denoted as $PoK\{(M, k, s, m) : e(M, Z \cdot h^k) = e(a, h)^m e(b, h)^s e(d, h)\}$ for brevity. More details can be found in [3]. Note that the computation of this protocol works mostly in $G_T$ which is a much larger group than $G_1$.

## 5 Self-blindable credential with verifier-local revocation

In this section, we first explain the rationale of our design, and then present a self-blindable credential scheme with verifier-local revocation.

### 5.1 Design Rationale

The basic approach of our design is to use the ASM signature scheme as the self-blindable credential issuance algorithm. In other words, a self-blindable credential $c$ is an ASM signature on the credential holder's attribute. We then design a blinding algorithm, such that the credential holder can randomize her ASM signature into a blinded credential $bc$. To convince the verifier that her $bc$ is valid, the credential holder proves that $bc$ is well formed in the sense that it is derived from a valid $c$. Our key innovation centers around designing a highly efficient blinding algorithm, replacing the original costly protocol $PoK\{(M, k, s, m) : e(M, Z \cdot h^k) = e(a, h)^m e(b, h)^s e(d, h)\}$ in [3].

Given an ASM signature $\sigma = (M, k, s)$ on message $m$, we observe that it can be blinded as follows. Select a blinding factor $f \in_R Z_q$ and compute $M' = M^f$. Namely, $M' = (a^{m \cdot f} \cdot b^{s \cdot f} \cdot d^f)^{\frac{1}{k+z}}$. Thus, $M'$ and $A' = (a^m b^s d)^f$ can be verified by

$$e(M', Z \cdot h^k) = e(A', h) \tag{1}$$

To fully hide the original signature $\sigma$, we need to hide $k$ as well. From (1), we get

$$\begin{aligned} & e(M', Z \cdot h^k) = e(A', h) \\ \Leftrightarrow\ & e(M', Z) e(M', h^k) = e(A', h) \\ \Leftrightarrow\ & e(M', Z) e(M'^k, h) = e(A', h) \end{aligned} \tag{2}$$

[2] In [3], the authors call their scheme BBS+ signature, as the scheme is developed from the well accepted BBS short group signature scheme in [8].

Let $M'' = M'^k = M^{f \cdot k}$, then $k$ is blinded by $f$. As such, we get a blinded credential $bc = (M', M'', A')$. To show its validity, a proof of knowledge protocol is needed to attest that $M', M''$ and $A'$ are in correct forms. To this point, an important notice is that such a protocol works exclusively in $G_1$, accounting for higher efficiency. We leave the details to Section 5.2.

CAVEAT. One may argue that $k$ can be blinded in a different way as follows. From Equation (1) we get

$$e(M', Z^{f'})e(M', h^{k \cdot f'}) = e(A'^{f'}, h)$$

where $f' \in_R Z_q$ is another random factor. Consequently, $(M' = M^f, Z' = Z^{f'}, H' = h^{k \cdot f'}, A' = (a^m b^s d)^{f \cdot f'})$ constitute a blinded credential, and can be validated by checking $e(M', Z')e(M', H') = e(A', h)$. However, this method does not maintain unlinkability. Consider two blinded credentials $(M'_1, Z'_1, H'_1, A'_1)$, $(X'_2, Z'_2, H'_2, A'_2)$. If they are generated from the same credential, then it is easy to verify that $e(M'_1 \cdot M'_2, Z'_1 \cdot Z'_2)e(M'_1 \cdot M'_2, H'_1 \cdot H'_2) = e(A'_1 \cdot A'_2, h)$; otherwise, this equation does not hold with an overwhelming probability. This is exactly the reason why the schemes in [22, 25] are not secure, although they do not use the ASM signature scheme.

### 5.2 Construction Details

We proceed to present the details of our construction of a self-blindable credential scheme with verifier-local revocation, whereby the verifier checks the revocation status of blinded credentials against the revoked items in the $CRL$. The verifier-local revocation approach has been widely used in the literature, e.g., [12, 16].

**Setup**$(1^\kappa)$**:** Given a system parameter $1^\kappa$, determine a bilinear map $e : G_1 \times G_2 \to G_T$. Select $a, b, d \in G_1$, $h \in G_2$. Compute $Z = h^z$, where $z \in_R Z_q^*$. Note that $(a, b, d, h, Z)$ are the parameters of the ASM signature scheme. Set the public parameters $params = (e, a, b, d, Z, CRL = \emptyset)$ and master secret key $msk = (z)$.

**CredIssue**$(msk = (z), u)$**:** The credential issuer computes an ASM signature $(M, k, s)$ on a user's identity $u$, where $M = (a^u b^s d)^{\frac{1}{k+z}} \in G_1$, and $k, s \in_R Z_q$. Set $c = (M, k, s, u)$.

**Blind**$(c = (M, k, s, u))$**:** User $u$ computes a blinded credential as follows.

1. Select $f \in_R Z_q$, and compute $M' = M^f, M'' = M'^k, A' = (a^u \cdot b^s \cdot d)^f$.
2. Construct the following proof of knowledge protocol, denoted as $PoK$.
$$PoK\{(k, \mu, \varsigma, f) : M'' = M'^k \wedge A' = a^\mu \cdot b^\varsigma \cdot d^f\}$$
where $\mu = u \cdot f, \varsigma = s \cdot f$.
3. Set $bc = (M', M'', A', PoK)$.

**Revoke**$(\lceil msk \rfloor, c = (M, k, s, u))$**:** To revoke credential $c$, the credential issuer updates $CRL$ with $k$ such that $CRL = CRL \cup \{k\}$. Similar to a $CRL$ in the traditional PKI, $CRL$ is signed by the credential issuer.

**CredVerify**$(bc, CRL)$**:** Given the $CRL$ and a blinded credential $bc = (M', M'', A', PoK)$, the verifier outputs 1 if *all* of the following are *True*; otherwise outputs 0.

$$\begin{cases} M' \neq 1 \in G_1 \\ PoK \text{ is valid} \\ e(M', Z)e(M'', h) \stackrel{?}{=} e(A', h) \\ \forall k \in CRL : M'' \neq M'^k \end{cases}$$

Remark. The reason why $M'$ is not allowed to be the unity in $G_1$ is the following: if $M'$ is the unity, then $f = 0 \pmod q$ and any value of $k$ would trivially satisfy the proof $PoK$. The check actually provides a guarantee that $f \neq 0 \pmod q$.

*Correctness.* Correctness of the construction is straightforward. If $bc$ is well-formed based on a valid $c$, then $e(M', Z)e(M'', h) = e(M^f, Z)e(M^{f \cdot k}, h)$, which is exactly $e(A', h)$.

### 5.3 Security Analysis

Next, we present the security analysis based on the requirements set out in Section 3.

*Unforgeability.* For unforgeability as defined in Appendix A, we have the following theorem.

**Theorem 1.** *The proposed construction achieves unforgeability, given that the ASM signature scheme [3] is existentially unforgeable.*

*Proof.* We show that there is a direct reduction from the ASM signature scheme to our scheme. Intuitively, the proof of knowledge in a blinded credential $bc$ guarantees the existence of $(k, u', s', f \neq 0)$ such that $e(M', Z)e(M', h^k) = e(a^{u'}, h)e(b^{s'}, h)e(d^f, h)$. Therefor, the tuple $(M'^{\frac{1}{f}}, k, \frac{s'}{f}, \frac{u'}{f})$ constitutes a valid ASM signature. If $bc$ is a forgery, then the tuple is a forgery for the ASM signature scheme. More formally, given an adversary $\mathcal{A}$ for our scheme, the forger $\mathcal{F}_{ASM}$ for the ASM signature scheme is constructed as follow. $\mathcal{F}_{ASM}$ invokes $\mathcal{A}$, and answers the latter's $\mathsf{CredIssue}$ oracle using its own signature signing oracle. Upon the output of $\mathcal{A}$, $\mathcal{F}_{ASM}$ executes the knowledge extractor of the proof of knowledge protocol to get $(k, u', s', f)$, and outputs $(M'^{\frac{1}{f}}, k, \frac{s'}{f}, \frac{u'}{f})$ as the forgery for the ASM signature.

*Unlinkability.* Unlinkability as defined in Appendix A requires the DDH assumption to hold in $G_1$. Let $g_1, g_2 \in G_1$. The DDH assumption states that it is computationally infeasible to distinguish between $(g_1, g_2, g_1^x, g_2^x)$ and $(g_1, g_2, g_1^x, C)$, where $x \in_R Z_q$ and $C \in_R G_1$. In the context of bilinear map, the DDH assumption is commonly referred to as XDH (external Diffie-Hellman) assumption. It is widely accepted that the XDH assumption holds in certain subgroups of MNT curves.

**Theorem 2.** *If the XDH assumption holds, then the proposed construction achieves unlinkability*[3]*.*

*Proof.* To prove the theorem, it suffices to show how to construct the simulator function (i.e., $\mathtt{Simulate}(\cdot)$) for a blinded credential $(M', M'', A', PoK)$. Select $M^* \in_R G_1$ and $A^* \in_R G_1$, and invoke the knowledge extractor for $PoK\{(k^*) : e(M^*, Z)e(M^*, h)^{k^*} = e(A^*, h)\}$ to extract $k^*$. Then set $M^{**} = M^{*k^*}$. Next, invoke the zero-knowledge proof simulator to produce $PoK^* = PoK\{(k^*, \mu^*, \varsigma^*, f^*) : M^{**} = M^{*k^*} \wedge A^* = a^{\mu^*} \cdot b^{\varsigma^*} \cdot d^{f^*}\}$. The simulated blinded credential is $(M^*, M^{**}, A^*, PoK^*)$. Due to the XDH assumption, $(M, a^u b^s d, M^*, A^*)$ is indistinguishable from $(M, a^u b^s d, M', A')$; in addition, $(M^*, M', M^{**}, M'')$ also constitutes a DDH tuple. Thus, $(M^*, M^{**}, A^*, PoK^*)$ by the simulator function is computationally indistinguishable from the real blinded credential $(M', M'', A', PoK)$.

*Revocability.* As shown from our construction, revocability is attained by using the $CRL$. A valid $(M', M'')$ pair in a blinded credential $bc$ must satisfy $M'' = M'^k$. Therefore, the verifier can check the revocation status of a blinded credential by examining it against all $k$-values in the $CRL$.

*Non-shareability.* The above scheme does not provide non-shareability. A typical method to achieve non-shareability in the anonymous credential literature is the all-or-nothing approach: an important secret (e.g., a user's long term signing key) is encoded in the credential, such that sharing of the credential leads to sharing of that important secret. It is easy to modify the above scheme to implement this all-or-nothing approach.

Suppose that user $u$'s long term signing key is an ElGamal-type key pair $(m, y = g^m)$, where $y$ is the public key certified by a CA and $m$ is the private key. To get a credential from the credential issuer, the user submits $a^m$ and $PoK\{(m) : \mathbb{A} = a^m \wedge y = g^m\}$. Then, the credential issuer computes an ASM signature on $m$ (instead of on user identity $u$). Our scheme ensures that the user must know $m$ in order to construct the proof of knowledge for $A' = a^{m \cdot f} b^{s \cdot f} d^f$ in running the $\mathsf{Blind}$ algorithm. As a result, user $u$ is enforced to share her private key $m$ in order to share her credential with another user.

[3] We actually cannot achieve unlinkability as defined in Definition 3 where the adversary/distinguisher is allowed to know an entire self-blindable credential. Here we assume that the adversary does not learn $k$, which in fact makes the achieved unlinkability weaker than defined in Definition 3.

## 6 Self-blindable Credential With Forward Unlinkability and Scalable Revocation

The above construction suffers from two weaknesses due to the verifier-local revocation mechanism. Firstly, it lacks *forward unlinkability*. Specifically, once a credential is revoked and the corresponding $k$ is published in the $CRL$, those blinded credentials generated prior to revocation become linkable. This may be undesirable for some applications where the revocation is only applied to subsequent credential usage. Secondly, the verifier's computation is linear to the total number of revoked credentials, which grows over the time. Even though the verifier does not have resource constraints, the unscalable revocation cost may hinder the practical deployment of the scheme. In this section, we propose another construction to address these two issues.

### 6.1 Forward Unlinkable Self-blindable Credential

We continue to use the ASM signature scheme for credential issuance. To achieve forward unlinkability, we cannot simply use $M'' = M^{f \cdot k}$ to hide $k$ since $k$ can be exposed in the $CRL$. We can further blind $k$ as $M'' = M^{f \cdot k} \cdot t^r$, together with $h^r \in G_2$ (which is used to cancel out the randomness $t^r$ in credential verification), where $t \in_R G_1, r \in_R Z_q$. However, this requires $h^r$ to be an element in $G_2$ which imposes heavy computation on the user end because computation in $G_2$ is even more expensive than in $G_T$ (let alone $G_1$). The challenge is to make the user computation work entirely over $G_1$.

We get over this problem by introducing two pairs of public parameters $(t_1, T_1 = t_1^z, )$ and $(t_2, T_2 = t_2^z)$ into the ASM signature setting, where $t_1, t_2 \in G_1$. Specifically, the new public key of the credential system becomes $(Z = h^z, h, a, b, d, t_1, t_2, T_1, T_2)$ and the private key remains as $z \in Z_q^*$. The signature generation and verification algorithm remain the same. The newly introduced $(t_1, T_1, t_2, T_2)$ are used for blinding purposes only.

Specifically, a blinded credential is generated from $(M, k, s, m)$ as follows, where $f, r_1, r_2 \in_R Z_q$:

$$\begin{aligned} M' &= M^f \cdot t_1^{r_1} \in G_1 \\ M'' &= M^{f \cdot k} \cdot T_2^{r_2} \in G_1 \\ A' &= (a^m b^s d)^f \in G_1 \\ T_1' &= T_1^{r_1} \in G_1 \\ T_2' &= t_2^{r_2} \in G_1. \end{aligned} \tag{3}$$

together with $PoK\{(k, \mu, \varsigma, f, \gamma, r_1, r_2) : M'' = M'^k t_1^{-\gamma} T_2^{r_2} \wedge A' = a^\mu b^\varsigma d^f \wedge T_1' = T_1^{r_1} \wedge T_2' = t_2^{r_2}\}$, where $\gamma = k \cdot r_1, \mu = m \cdot f, \varsigma = s \cdot f$. The blinded credential $(M', M'', A', T_1', T_2', PoK)$ can be verified by checking:

$$\begin{cases} A' \neq 1 \in G_1 \\ PoK \text{ is valid} \\ e(M', Z) e(M'', h) \stackrel{?}{=} e(A', h) e(T_1', h) e(T_2', Z) \end{cases}$$

As in the previous construction, the proof of knowledge $PoK$ ensures that these blinded elements are well formed.

**Security**. We analyze unforgeability and unlinkability of this scheme, and in particular, the introduction of $(t_1, t_2, T_1, T_2)$ does not affect unforgeability of credentials. We have the following theorems with their proofs left in Appendix B.1 and B.2, respectively.

**Theorem 3.** *If the original ASM signature [3] is existentially unforgeable, then the above forward unlinkable self-blindable credential scheme achieves unforgeability.*

**Theorem 4.** *If the XDH assumption holds, then the above forward unlinkable self-blindable credential scheme achieves unlinkability.*

### 6.2 Scalable Revocation

To avoid the linear computation at the verifier side, we take advantage of the dynamic accumulator in [27], a revocation technique widely used in anonymous credentials and group signatures.

*Review of Dynamic Accumulator* [27]. A dynamic accumulator scheme allows a large set of values to be accumulated into a single value called the *accumulator*. For each accumulated value, there exists a *witness*, which is the evidence attesting that the accumulated value is indeed contained in the accumulator. The proof can be carried out in a zero-knowledge fashion such that no information is exposed about the witness and the value.

The details of Nguyen's dynamic accumulator [27] are as follows. Let $e : G_1 \times G_2 \rightarrow G_T$ be a bilinear map as above, and $h$ be generators of $G_2$. The public parameters include $(Z = h^z, h)$, and the private key is $z \in Z_q^*$. The accumulator, denoted as $\Lambda$, is initially assigned with an random value[4] in $G_1$. The witness for a value $k$ accumulated in $\Lambda$ is $W = \Lambda^{\frac{1}{k+z}}$. As such, $(W, k)$ can be verified by $e(W, Z \cdot h^k) = e(\Lambda, h)$. For brevity, the proof of knowledge protocol showing that $k$ is accumulated in $\Lambda$ is denoted by $PoK\{(W, k) : e(W, Z \cdot h^k) = e(\Lambda, h)\}$. More details can be found in [27, 3].

Nguyen's accumulator supports revocation of values from the accumulator. Specifically, the revocation of value $k_j$ in the present accumulator $\Lambda_{old}$ is performed as follows. The authority computes the new accumulator as $\Lambda_{new} = \Lambda_{old}^{\frac{1}{k_j+z}}$, and publishes a new entry $\langle \Lambda_{new}, k_j \rangle$ in a public board. To respond to this event, every witness holder needs to update her witness in order to keep consistent with the new accumulator. In particular, for a holder with witness $W_i$ (corresponding to $k_i$), the witness can be updated by computing $W_i^{new} = W_i^{\frac{1}{k_j+z}} = (\Lambda_{old}^{\frac{1}{k_i+z}})^{\frac{1}{k_j+z}} = \Lambda_{old}^{(\frac{1}{k_i+z} - \frac{1}{k_j+z}) \cdot \frac{1}{k_j-k_i}} = (\frac{W_i}{\Lambda_{new}})^{\frac{1}{k_j-k_i}}$. It is clear that each witness holder updates her witness without the knowledge of $z$.

*Blinded Proof.* To integrate Nguyen's dynamic accumulator with the above forward unlinkable self-blindable credential scheme, we have to avoid the original proof of knowledge protocol $PoK\{(W, k) : e(W, Z \cdot h^k) = e(\Lambda, h)\}$ in [3, 27]. Our observation is that the structure of a witness/value pair for the dynamic accumulator and that for an ASM signature are almost identical, except the difference between $\Lambda$ and $A = a^m b^s d$. We thus can use a blinded proof following exactly the same rationale for the above forward unlinkable self-blindable credential scheme: blinding of $(W, k, \Lambda)$ works exactly in the same way as blinding of $(M, k, s, m)$, with $\Lambda^f$ replacing $(a^m b^s d)^f$. We omit the details, as they are simply a repetition of Equation 3.

### 6.3 Construction Details

The complete construction is a combination of the above forward unlinkable credential scheme and Nguyen's dynamic accumulator scheme (where the two primitives share system parameters). The basic idea is that the "$k$" element of each credential is accumulated into the accumulator, so that if a credential is revoked, then the corresponding "$k$" element is removed from the accumulator. Below are the details.

**Setup**$(1^\kappa)$**:** Determine a bilinear map $e : G_1 \times G_2 \rightarrow G_T$. Select $a, b, d, t_1, t_2 \in G_1$, $h \in G_2$. Compute $Z = h^z, T_1 = t_1^z, T_2 = t_2^z$, where $z \in_R Z_q^*$. Initialize an accumulator $\Lambda \in_R G_1$. Set public parameters $params = (e, a, b, d, t_1, t_2, h, Z, T_1, T_2, \Lambda, CRL = \emptyset)$, and master secret key $msk = (z)$.

**CredIssue**$(msk = (z), u)$**:** Compute an ASM signature $(M, k, s, m)$ on the user's private key $m$, as discussed in Section 5.3 for achieving non-shareability, such that $M = (a^m b^s d)^{\frac{1}{k+z}}$. Compute the witness $W$ corresponding to $k$, such that $W = \Lambda^{\frac{1}{k+z}}$, with $\Lambda$ being the latest accumulator. Set the credential to be $c = (M, k, s, m, W)$.

[4] In [27], the accumulator $\Lambda$ for a set of values $(k_1, k_2, \cdots, k_\ell)$ is actually calculated as $\Lambda = \bar{g}^{\prod_{j=1}^{\ell}(k_j+z)}$, where $\bar{g} \in G_1$ is a public parameter. As such, both accumulation of new values and revocation of existing values require witness updates. The advantage of our assigning $\Lambda$ a random initial value is that only revocation of values involves witness updates.

**Blind**$(c = (M, k, s, m, W)$**:** Generate a blinded credential $bc$ as follows. Let $\Lambda$ be the latest accumulator. Select $f, r_1, r_2 \in_R Z_q$, and compute

1. $M' = (M \cdot W)^f \cdot t_1^{r_1}$;
2. $M'' = (M \cdot W)^{f \cdot k} \cdot T_2^{r_2}$;
3. $A' = (a^m b^s d \cdot \Lambda)^f$;
4. $T_1' = T_1^{r_1}$;
5. $T_2' = t_2^{r_2}$;
6. $PoK\{(k, \mu, \varsigma, f, \gamma, r_1, r_2) : M'' = M'^k t_1^{-\gamma} T_2^{r_2} \wedge A' = a^\mu b^\varsigma d^f \Lambda^f \wedge T_1' = T_1^{r_1} \wedge T_2' = t_2^{r_2}\}$, where $\gamma = k \cdot r_1, \mu = m \cdot f, \varsigma = s \cdot f$.

Set $bc = (M', M'', A', T_1', T_2', PoK)$.

**Revoke**$(msk = (z), c = (M, k, s, m, W))$**:** To revoke a credential $c$, update the present accumulator $\Lambda$ as $\Lambda_{new} = \Lambda^{\frac{1}{k+z}}$, and publish a new entry $(k, \Lambda_{new})$ in the $CRL$ such that $CRL = CRL \cup (k, \Lambda_{new})$[5]. Existing self-blindable credential holders update their respective witnesses as described earlier.

**CredVerify**$(bc, CRL)$**:** The verifier gets the latest accumulator from the $CRL$, and checks all the following for a given blinded credentials $bc = (M', M'', A', T_1', T_2', PoK)$.

$$\begin{cases} A' \neq 1 \in G_1; \\ PoK \text{ is valid}; \\ e(M', Z)e(M'', h) \stackrel{?}{=} e(A', h)e(T_1', h)e(T_2', Z); \end{cases}$$

*Correctness.* Correctness can be easily shown. By cancelling out the elements associated with $r_1, r_2$, we have $e(M', Z)e(M'', h) = e(A', h)e(T_1', h)e(T_2', Z) \Rightarrow e((M \cdot W)^f, Z)e(M \cdot W)^{f \cdot k}, h) = e((a^m b^s d \cdot \Lambda)^f, h)$, which is clearly a combination of an ASM signature and an accumulator.

### 6.4 Security Analysis

Revocability and non-shareability of the construction are clearly achieved. So is forward unlinkability, which can be proved similarly as in Theorem 4. It remains to analyze unforgeability: it seems hard to directly relate unforgeability to that of the forward unlinkable scheme in Section 6.1, although intuitively it is clear that the extra information $W = \Lambda^{\frac{1}{k+z}}$ does not affect unforgeability.

We actually relate unforgeability to the $\ell$-Strong Bilinear Collusion Attack Assumption ($\ell$-SBCAA), which is implied in [7, 8]. Let $G_1, G_2$ be as in the bilinear map, and $g \in G_1, h \in G_2$. $\ell$-SBCAA asserts that for $z \in Z_q^*$, it is hard to compute $(k_0 \in Z_q, g^{\frac{1}{k_0+z}})$, given a $(\ell + 3)$-tuple $(g, h, h^z, (k_1, g^{\frac{1}{k_1+z}}), (k_2, g^{\frac{1}{k_2+z}}), \cdots, (k_\ell, g^{\frac{1}{k_\ell+z}}))$, $\forall k_i \in Z_q$. We have the following theorem, and the proof can be found in Appendix B.3.

**Theorem 5.** *If the $\ell$-SBCAA holds, then our construction achieves unforgeability.*

### 6.5 Witness Update Outsourcing

We rely upon Nguyen's dynamic accumulator for credential revocation. Recall that it requires all users to update their witnesses in the event of credential revocation. To get her witness updated, a user should either keep connected to the revocation server around the clock or do a batch update after a period of time. For resource-constrained devices, batch update is a more viable choice.

In a large-scale system where credential revocations are frequent, a batch update consists of a plethora of witness update events. Then, the out-of-the-band witness update step may severely impact the system performance. Fortunately, in our scheme the credential component corresponding to the accumulator (i.e., $W, k$) can be public, as its role is simply a testimony that the credential is not revoked. Based on this observation, users can outsource the witness update task to an *untrusted*

[5] It may seem strange to publish accumulator values $\Lambda$ in the $CRL$. As introduced earlier, $(k, \Lambda)$ should be published in a public board in order for credential holders to update their witnesses. Here we actually abuse the $CRL$, using it as the public board.

third party, who helps update their witnesses. A user simply needs to retrieve her witness back at the time of authentication. Note that the third party is not necessarily trusted, as $W, k$ are deemed not secret. We stress that the disclosure of $W, k$ does not compromise unlinkability of our construction, as the security analysis already assumes that the adversary knows user credentials.

## 7 Performance Analysis

We provide an analysis on the performance of our constructions. Let $|G|$ denote the bit length of an element in group $G$, and $\texttt{EXP}(G)$ denote an exponentiation operation in $G$. The efficiency advantage of our constructions is attributed to the fact that the user computes entirely over $G_1$, whereas the verifier performs bilinear pairings and group computation over $G_T$. This feature distinguishes our proposal from virtually all existing bilinear map-based anonymous credentials and group signatures, e.g., those mentioned in Section 2.[6] Note that in the context of bilinear map, $|G_1|$ is much shorter than $|G_2|$ and $|G_T|$, and $\texttt{EXP}(G_1)$ is an order of magnitude faster than $\texttt{EXP}(G_2)$, $\texttt{EXP}(G_T)$, and pairing operation.

In our second construction described in Section 6.3, the proof of knowledge $PoK$ in the blinded credential incurs $4 \cdot \texttt{EXP}(G_1)$ in computation (for commitment) and $4|G_1| + 6|Z_q|$ bits in communication (for response). Consequently, the total overhead of generating a blinded credential includes $9 \cdot \texttt{EXP}(G_1)$ and $9|G_1| + 6|Z_q|$ bits. Similarly, the overhead of our first construction is $5 \cdot \texttt{EXP}(G_1)$ in computation and $5|G_1| + 4|Z_q|$ in communication, respectively. Note that for computation cost, we only count the number of exponentiations. Furthermore, by applying optimized algorithms for multi-exponentiations, e.g., [26, 21], the computation overhead for a multi-exponentiation is just slightly more than a single-exponentiation. We thus do not distinguish multi-exponentiation and single-exponentiation by $\texttt{EXP}(G)$ in the calculations.

We list in Table 1 the comparison results of the user side performance between each of our schemes with a representative existing anonymous credential scheme that implements similar functionalities. Specifically, we compare our first scheme with DAA (see e.g., [9]), both supporting verifier-local revocation; for our second scheme, we compare it with the relevant part of the k-TAA scheme in [3]. Note that in [3], the k-TAA scheme includes, as building blocks, concrete proof of knowledge protocols for both the ASM signature and Nguyen's dynamic accumulator. The combination of the two protocols actually achieves the same revocation support as our second construction, and represents the state of the art of bilinear map-based anonymous credentials. We thus compare our construction with that combination (note that we do not compare with k-TAA)[7].

**Table 1.** Performance Comparison

| | **Computation** | **Communication** |
|---|---|---|
| DAA in [9] | $5 \cdot \texttt{EXP}(F_n)$[a.] | $> 5\|n\|$ |
| Our first construction | $5 \cdot \texttt{EXP}(G_1)$ | $5\|G_1\| + 3\|Z_q\|$ |
| Combined scheme in [3] | $8 \cdot \texttt{EXP}(G_1) + 2 \cdot \texttt{EXP}(G_T)$ | $7\|G_1\| + 2\|G_T\| + 12\|Z_q\|$ |
| Our second construction | $9 \cdot \texttt{EXP}(G_1)$ | $9\|G_1\| + 6\|Z_q\|$ |

[a.] $n$ is a RSA modulus and $F_n$ denotes the finite filed over $n$.

When instantiating the bilinear map over MNT curves with 80-bit security, it is estimated in [13] that $|G_T| = 1026$, $|G_1| = 171$, and $|Z_q| = 160$, and $1 \cdot \texttt{EXP}(G_T) = 10 \cdot \texttt{EXP}(G_1)$. In such a case, $|n| = 1024$, and $1 \cdot \texttt{EXP}(F_n) \approx 1 \cdot \texttt{EXP}(G_T)$. Under such a setting, we can see that the computation and communication overhead of our first scheme is about 10% and 25% of DAA, respectively. For

[6] While claiming efficiency advantage over existing anonymous credential and group signature schemes, we acknowledge that different schemes may depend on different cryptographic assumptions. Anyway, our objective is to develop an efficient anonymous authentication primitive with reasonable security guarantees, suitable for weak devices.

[7] While we do not compare with other bilinear map-based anonymous credential schemes such as [7, 8], they are no better in efficiency than [3].

our second construction, its computation cost is around 30% of that of [3]. Note that we have assumed various optimizations for the computation in [3], e.g., transfer pairing operations into fixed-base exponentiations in $G_T$ by pre-computing. We believe that this efficiency gap should be even larger in practice, especially upon resource-constrained devices, due to the fact that the multi-exponentiations in [3] contain too many elements. The communication overhead of our second construction is about 50% of [3].

## 8 Conclusion

Due to high computation/communication overheads, existing anonymous entity authentication techniques such as anonymous credentials and group signatures are not suitable for for resource-constraint smart devices. To address this problem, we have proposed two self-blindable credential schemes with superior performance and revocation support. The efficiency advantage of our constructions stems from the fact that the computations at the weak device side work entirely on $G_1$ of bilinear map, which is an order of magnitude faster than pairing operations. We have formally proved their security and also analyzed their performance against relevant existing schemes.

## References


1. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. *A Practical and Provably Secure Coalition-Resistant Group Signature Scheme*, Proc. Advances in Cryptology, Crypto'00, LNCS 1880, pp. 255-270.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. *Structure-Preserving Signatures and Commitments to Group Elements*, Proc. Advances in Cryptology, Crypto'10, LNCS 6223, pp. 209-236.
3. M.H. Au, W. Susilo, and Y. Mu. *Constant-Size Dynamic k-TAA*, Proc. Security and Cryptography for Networks, SCN'06, LNCS 4116, pp. 111-125.
4. G. Ateniese, and B. Medeiros. *Efficient Group Signatures without Trapdoors*, Proc. Advances in Cryptology, Asiacrypt'03, pp. 246-268, 2003.
5. J. Balasch. *Smart Card Implementation of Anonymous Credentials*, Master thesis, K. U. Leuven, 2008.
6. S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privay*, MIT Press, 2000.
7. D. Boneh, X. Boyen. *Short Signatures Without Random Oracles*, Proc. Advances in Cryptology, Enrocrypt'04, LNCS 3027, pp. 56-73.
8. D. Boneh, X. Boyen, and H. Shacham. *Short Group Signatures*, Proc. Advances in Cryptology, Crypto'04, LNCS 3152, pp. 41-55.
9. P. Bichsel, J. Camenisch, T. Gro, and V. Shoup. *Anonymous Credentials on a Standard Java Card*, Proc. ACM Conference on Computer and Communication Security, CCS'09, pp. 600-610.
10. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. *P-signatures and Noninteractive Anonymous Credentials*, Proc. the $5^{th}$ Conference on Theory of Cryptography, TCC'08, pp. 356-374.
11. L. Batina, J.H. Hoepman, B. Jacobs, W. Mostowski, P. Vullers. *Developing Efficient Blinded Attribute Certificates on Smart Cards via Pairings*, Proc. 9th IFIP International Conference on Smart Card Research and Advanced Application, CARDIS'10, LNCS 6035, pp. 209-222.
12. D. Boneh, and H. Shacham. *Group Signatures with Verifier-Local Revocation*, Proc. ACM conference on Computer and Communications Security, CCS'04, pp. 168-177.
13. X. Boyen. *A Tapestry of Identity-based Encryption: Practical Frameworks Compared*, Journal of Appllied Cryptography, pp. 3-19, Vol. 1, No. 1, 2008.
14. D. Chaum. *Security Without Identification: Transaction Systems to Make Big Brother Obsolete*, Communications of the ACM, Vol 28, No. 10, pp. 1030-1044, 1985.
15. D. Chaum, E. van Heyst. *Group Signatures*, Proc. Advances in Cryptology, Eurocrypt'91, LNCS 547, pp. 257-265.
16. J. Camenisch and E. V. Herreweghen. *Design and Implementation of the Idemix Anonymous Credential System.* Proc. ACM Conference on Computer and Communication Security, CCS'02.
17. J. Camenisch, and A. Lysyanskaya. *An Efficient Sysem for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation*, Proc. Advances in Cryptology, Eurocrypt'01, pp. 93-118.
18. J. Camenisch, A. Lysyanskaya. *A Signature Scheme with Efficient Protocols*, Proc. Security and Cryptography for Networks, SCN'02, LNCS 2576, pp. 268-289.
19. J. Camenisch, M. Kohlweiss, and C. Soriente. *An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials*, Proc. Public Key Cryptography, PKC'09, LNCS 5443, pp. 481-500.
20. J. Camenisch, M. Michels. *A Group Signature Scheme with Improved Efficiency*, Proc. Advances in Cryptology, Asiacrypt'98, LNCS 1514, pp. 160-174.
21. V. S. Dimitrov, G. A. Jullien, and W. C. Miller, *Complexity and Fast Algorithms for Multi-exponentiations*, IEEE Transactions on Computers, vol 49, no 2, pp. 141-147, 2000.

22. K. Emura, A. Miyaji, K. Omote. *A Certificate Revocable Anonymous Authentication Scheme with Designated Verifier*, Proc. International Conference on Availability, Reliability and Security, ARES'09, pp. 769-773.
23. A. Fiat, A. Shamir. *How to Prove Yourself: Practical Solution to Identification and Signature Problems*, Proc. Advances in Cryptology, Crypto'86, LNCS 263, pp. 186-194.
24. J. Groth, A. Sahai. *Efficient Non-interactive Proof Systems for Bilinear Groups*, Proc. Advances in Cryptology, Eurocrypt'08, LNCS 4965, pp. 415-432.
25. S. Kiyomoto, T. Tanaka. *Anonymous Attribute Authentication Scheme using Self-blindable Certificates*, Proc. IEEE International Conference on Intelligence and Security Informatics, pp. 215-217, 2008.
26. B. Moeller. *Algorithm for Multi-exponentiation*, Proc. Selected Areas in Cryptography, SAC'01, pp. 165-180.
27. L. Nguyen. *Accumulators from Bilinear Pairings and Applications*, Proc. CT-RSA'05, LNCS 3376, pp. 275-292.
28. R.L. Rivest, A. Shamir, Y. Tauman. *How to Leak a Secret*, Proc. Advances in Cryptology, Asiacrypt'01, LNCS 2248, pp. 552-565.
29. M. Sterckx, B. Gierlichs, B. Preneel, T. Verbauwhede. *Efficient Implementation of Anonymous Credentials on Java Card Smart Cards*, Proc. Information Forensics and Security, IEEE, WIFS'09. pp. 106-110.
30. I. Teranishi, J. Furukawa, and K. Sako. *k-Times Anonymous Authentication (Ex-tended Abstract)*. Proc. Advances in Cryptology, Asiacrypt'04, pp. 308-322.
31. E. R. Verheul. *Self-Blindable Credential Certificates from the Weil Pairing*, Proc. Advances in Cryptology, Asiacrypt'01, LNCS 2248, pp. 533-551.

## A Formulation of Security Notions

The formulation of unforgeability, unlinkability is presented below.

**Definition 2.** *[Unforgeability] A revocable self-blindable credential scheme satisfies unforgeability if for any PPT adversary $\mathcal{A}$, the probability of the following "unforgeability game" returning* 1 *is negligible in $\kappa$.*

1. $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa)$;
2. $bc \leftarrow \mathcal{A}^{\mathsf{CredIssue}(msk,\cdot)}(params)$;
3. *if* $\mathsf{CredVerify}(bc, CRL) = 1 \wedge \mathsf{Origin}(bc) \notin \{c_1, c_2 \ldots c_\ell\}$ *then return* 1*; and returns* 0 *otherwise, whereby* $\{c_i\}_{i=1}^{\ell}$ *are the set of credentials returned by the* $\mathsf{CredIssue}(msk, \cdot)$ *oracle, and* $\mathsf{Origin} : \mathcal{BC} \rightarrow \mathcal{C}$ *is a function taking as input a blinded credential and outputting the original credential from which the former is generated.*

Specifically, the definition asserts that given a set of credentials, the adversary cannot produce a valid blinded credential, which is not generated from any given credential. In the definition, unforgeability of the original credentials is implicit, as if the adversary can forge a self-blindable credential, it can certainly generate a valid blinded credential.

**Definition 3.** *[Unlinkability] A revocable self-blindable credential scheme satisfies unlinkability if for any PPT adversary $\mathcal{A}$, the probability of the following "unlinkability game" returns* 1 *is* $1/2 + \nu(\kappa)$*, where $\nu(.)$ is a negligible function.*

1. $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa)$; $c \leftarrow \mathsf{CredIssue}(msk, u)$;
2. $\sigma \xleftarrow{R} \{0, 1\}$;
3. *if* $\sigma = 1$ *then* $bc \leftarrow \mathsf{Blind}(c)$;
   *else* $bc \leftarrow \mathsf{Simulate}(params)$;
4. $\sigma^* \leftarrow \mathcal{A}(params, c, bc)$;
5. *if* $\sigma = \sigma^*$ *then return* 1 *else return* 0;

Specifically, the "unlinkability game" proceeds with the adversary $\mathcal{A}$ in such a way that $\mathcal{A}$ is given either a blinded credential $bc$ generated from a genuine credential $c$ or a simulation result independent of $c$, depending on the random bit $\sigma$. Above, $\mathsf{Simulate}(\cdot)$ denotes the simulator function that outputs a simulation of a blinded credential. Finally, $\mathcal{A}$ outputs a guess on $\sigma$. Unlinkability requires that $\mathcal{A}$ should not gain any significant advantage over a random guess, even though it is given the original credential $c$ in the game.

## B Security Proofs

### B.1 Proof of Theorem 3

*Proof.* We show by contradiction that if there exists a PPT adversary $\mathcal{A}$ against unforgeability of our forward unlinkable self-blindable credential signature scheme, then there exists a PPT forger $\mathcal{F}$ that compromises unforgeability of the ASM signature scheme [3]. Specifically, the construction of $\mathcal{F}$ is as follows. $\mathcal{F}$ is challenged with the ASM signature scheme and is given oracle access to the corresponding signing algorithm of the scheme. $\mathcal{F}$ first gets two ASM signatures $(M_1, k_1, s_1, m_1)$ and $(M_2, k_2, s_2, m_2)$ by invoking the signing algorithm. It holds that $M_1 = (a^{m_1} b^{s_1} d)^{\frac{1}{k_1+z}} \Rightarrow M_1^z = a^{m_1} b^{s_1} d M_1^{-k_1}$; so is $M_2^z = a^{m_2} b^{s_2} d M_2^{-k_2}$. $\mathcal{F}$ sets $t_1 = M_1, T_1 = a^{m_1} b^{s_1} d M_1^{-k_1}, t_2 = M_2, T_2 = a^{m_2} b^{s_2} d M_2^{-k_2}$, and gives them to $\mathcal{A}$ together with the public parameters of the ASM signature scheme, in order to simulate the public parameters of our construction. Clearly the simulation is perfect.

From then on, it is straightforward for $\mathcal{F}$ to answer the queries of $\mathcal{A}$ to the CredIssue oracle using its own ASM signing oracle. The simulation is again perfect, given that the simulation of $t_1, T_1, t_2, T_2$ is perfect. Upon receipt of the output $(M', M'', A', T_1', T_2', PoK)$ of $\mathcal{A}$, $\mathcal{F}$ executes the knowledge extractor for $PoK$ to get $(k, \mu, \varsigma, f, \gamma, r_1, r_2)$, and outputs $((M' \cdot t_1^{-r_1})^{\frac{1}{f}}, k, \frac{\varsigma}{f}, \frac{\mu}{f})$ as the forgery for the ASM signature scheme. This is because

$$
\begin{aligned}
& e(M', Z) e(M'', h) = e(A', h) e(T_1', h) e(T_2', Z) \\
\Rightarrow\ & e(M', Z) e(M'^k t_1^{-\gamma} T_2^{r_2}, h) = e(A', h) e(T_1', h) e(T_2', Z) \\
\Rightarrow\ & e(M', Z) e(M'^k t_1^{-\gamma}, h) = e(A', h) e(T_1', h) \\
\Rightarrow\ & e(M', Z) e(M'^k t_1^{-\gamma}, h) = e(A', h) e(T_1^{r_1}, h) \\
\Rightarrow\ & e(M', Z) e(M'^k t_1^{-\gamma}, h) = e(A', h) e(t_1^{r_1}, Z) \\
\Rightarrow\ & e(M' t_1^{-r_1}, Z) e(M'^k t_1^{-\gamma}, h) = e(A', h) \\
\Rightarrow\ & e(M' t_1^{-r_1}, Z) e(M' t_1^{-\gamma/k}, h^k) = e(a^\mu b^\varsigma d^f, h)
\end{aligned}
$$

When $r_1 = \gamma/k$, the output is an ASM signature.

### B.2 Proof of Theorem 4

*Proof.* Construction of the simulator function $\texttt{Simulate}(\cdot)$ for a blinded credential follows the idea of the proof for Theorem 2. Select $M^*, A^* \in_R G_1$, $k^*, \gamma^* \in_R Z_q$, and let $r_1^* = \gamma^*/k^*$. Then invokes the knowledge extractor for $PoK\{(r_2^*) : e(M^*, Z) e(M^*, h)^{k^*} e(t_1, h)^{-\gamma^*} e(T_2, h)^{r_2^*} = e(A^*, h) e(T_1, h)^{r_1^*} e(t_2, Z)^{r_2^*}\}$ to extract $r_2^*$. Then set $M^{**} = M^{*k^*} t_1^{-\gamma^*} T_2^{r_2^*}, T_1^* = T_1^{r_1^*}, T_2^* = t_2^{r_2^*}$. Next, invoke the zero-knowledge proof simulator to produce $PoK^* = PoK\{(k^*, \mu^*, \varsigma^*, f^*, \gamma^*, r_1^*, r_2^*) : M^{**} = M^{*k^*} t_1^{-\gamma^*} T_2^{r_2^*} \wedge A^* = a^{\mu^*} b^{\varsigma^*} d^{f^*} \wedge T_1^* = T_1^{r_1^*} \wedge T_2^* = t_2^{r_2^*}\}$. Due to the XDH assumption, $(M^*, M^{**}, A^*, T_1^*, T_2^*, PoK^*)$ are distributed computationally indistinguishable from the actual blinded credential $(M', M'', A', T_1', T_2', PoK)$.

### B.3 Security Proof for Theorem 5

*Proof.* The proof follows the idea in the proof for Theorem 3. To avoid repetition, we only briefly explain the main idea on how the adversary $\mathcal{A}$ for $\ell$-SBCCA simulates the adversary $\mathcal{A}'$ for our construction. Given a $(\ell+3)$-tuple $(g, h, h^z, (k_1, g^{\frac{1}{k_1+z}}), \cdots, (k_\ell, g^{\frac{1}{k_\ell+z}}))$, $\mathcal{A}$ sets $d = g, a = g^{r_a}, b = g^{r_b}, \Lambda = g^r$, where $r_a, r_b, r \in_R Z_q$. Now $\mathcal{A}$ can easily transform the $\ell$-tuple $(k_1, g^{\frac{1}{k_1+z}}), \cdots, (k_\ell, g^{\frac{1}{k_\ell+z}})$ into $\ell$ self-blindable credentials $(M_i = a^{m_i} b^{s_i} d = (g_1^{(m_i \cdot r_a + s_i \cdot r_b + 1) \cdot \frac{1}{k_i+z}}, k_i, s_i \in_R Z_q, m_i \in_R Z_q, W_i = g^{r \cdot \frac{1}{k_i+z}}), i \in [1..\ell]$. With the $\ell$ credentials in place, $\mathcal{A}$ can compute $t_1, T_1, t_2, T_2$ and answer the queries of $\mathcal{A}'$ to the CredIssue oracle in a similar way as in the proof for Theorem 3.