This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Provably Secure Self-certified Partially Blind Signature Scheme from Bilinear Pairings

Xiaodong Lin, Rongxing Lu, Haojin Zhu, Pin-Han Ho and Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
Email: {xdlin,rxlu,h9zhu,pinhan,xshen}@bbcr.uwaterloo.ca

***Abstract*—To enable the practical electronic cash systems, significant attention has been paid to the partially blind signature because of its unlinkability and unforgeability. To the best of our knowledge, most of partially blind signature schemes are constructed under either the traditional public key certificate based system or the ID-based system, which may incur significant efforts in certification management and/or revocation. In this paper, we introduce a novel approach for partially blind signature with self-certified public keys. This is the first research effort for significantly reducing the certificate management and revocation in partially blind signature, and is characterized by the adoption of bilinear pairings and the analytic techniques of provable security.**

***Keywords* – Partially blind signature, self-certified, bilinear pairings, provable security.**

I. Introduction

Blind signature is a very useful cryptographic technique [1], [2], which allows a user to get a signature without giving the signer any information about the actual message or the resulting signature. A typical blind signature scheme must satisfy the following two fundamental characteristics: *Blindness* and *Unforgeability*.

- Blindness: A blind signature allows a user to obtain a valid signature on message m without revealing anything about the message m to the signer. In addition, the signature is not traceable in the sense that the signer cannot determine when he or she signed this message after the signature is published, although he or she knows that it is really signed by him or her.
- Unforgeability: Only the original signer can generate valid signatures, while anyone else cannot.

Because of the blindness and unforgeability, the blind signature has been extensively used as electronic cash and electronic votes in electronic commerce and electronic business systems [3], where individual's privacy (anonymity) is of particular importance. Over the past years, many blind signature schemes have been proposed in [4]–[6]. However, when the previously reported blind signature schemes are used in a practical electronic cash system, they are subject to some common problems, such as the unlimited growth of the bank's database which keeps all the spent e-coins for preventing double spending and the bank's uninscribing the value on the blindly issued e-coins. To cope with these problems, the concept of partially blind signature was introduced by Abe and Fujisaki [7]. A partially blind signature is an extension of blind signature that allows a signer to produce a blind signature on a message for a user. The signature also explicitly includes commonly agreed information which remains clearly visible despite of the blinding process. Because of the partial blindness property, a partially blind signature scheme is more efficient than ordinary blind signature schemes when it is used in an electronic cash system. Therefore, in recent years, many partially blind signature schemes have appeared [8]–[11].

However, for these partially blind signature schemes, most of them are constructed under either the traditional certificate based system [9], [10] or the ID-based system [11]. Recently, the notion of self-certified public key system was introduced in [12], where the private key of each user is only known to the user himself, while the corresponding public key is derived from the signature of the user's identity and private key, signed by a trusted system authority SA. The self-certified system can be taken as an intermediate between the traditional certificate based systems [13] and the ID-based systems [14]. Compared with the former, it can implicitly validate the user's public keys; while compared with the latter, it can get rid of the key escrow issues [15]. To the best of our knowledge, a partially blind signature under the self-certified public key system has never been explicitly reported, which is the focus of this research.

Concretely, our main contributions in this paper are in two-fold: (i) We formalize the definition and security model for self-certified partially blind signature; (ii) We present the first self-certified partially blind signature scheme based on the bilinear pairings [16], [17]; and use the techniques from provable security to analyze its security [18], [19].

The remainder of this paper is organized as follows. In Section II, we provide a formal definition and security model for self-certified partially blind signature. In Section III, we review the bilinear pairings and the underlying problems which form the basis of this study. Then, the proposed self-certified partially blind signature scheme will be presented in Section IV, followed by the security proof in Section V. Finally, we draw our conclusions in Section VI.

II. Notations and Definitions

In this section, we define the self-certified partially blind signature together with its security notions.

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

A. Notations

Let $\mathbb{N} = \{1, 2, 3, \ldots\}$ be the set of positive integers. If x, y are two strings, then $|x|$ denotes the length of x and $x\|y$ represents the concatenation of x and y. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If $\mathbb{S}$ is a set, then $|\mathbb{S}|$ denotes its size and $s \overset{R}{\longleftarrow} \mathbb{S}$ denotes the operation of picking a random element s of $\mathbb{S}$ uniformly.

B. Definition of self-certified partially blind signature

Definition 1 (Self-certified Partially Blind Signature): A self-certified partially blind signature $\mathcal{SCPBS}$ scheme consists of the following four algorithms $\{\mathsf{Setup}, \mathsf{KGen}, \mathsf{BSign}, \mathsf{Verify}\}$:

- System parameter setup algorithm Setup: it is a probabilistic algorithm which takes the security parameter k as input, and outputs the system parameters params and the master secret key $\mathsf{masterkey}$. The system parameters params are publicly known, while the $\mathsf{masterkey}$ is only known by the trusted system authority SA.
- Key generation algorithm KGen: on input of the system parameters params, a user with identity ID first generates partial public and private key pair (pk, sk), and submits (pk, ID) to the trusted system authority SA. SA then takes $(pk, ID, \mathsf{params}, \mathsf{masterkey})$ as inputs, and returns another partial private key sk' to the user via a secure channel.
- Blind signing algorithm BSign: on input of a message m and the system parameters params, a signer A with (ID_A, pk, sk, sk') and a user B first negotiates an agreed information $\mathsf{info} = \Delta$, then generates a blind signature σ on message m by interaction.
- Signature verifying algorithm Verify: On input of $(\sigma, m, \mathsf{info}, ID_A, pk, \mathsf{params})$, this algorithm outputs "1" if the signature-message pair (σ, m) is valid with respect to ID_A, pk, info, and "0" otherwise.

The four algorithms must satisfy the consistency constraint of the self-certified partial blind signature, i.e.,

$$\begin{aligned} \forall\ m\ \ &: \mathsf{Verify}(\sigma, m, \mathsf{info}, ID_A, pk, \mathsf{params}) = 1, \\ &\text{where } \sigma = \mathsf{BSign}(ID_A, pk, sk, sk', \mathsf{info}, \mathsf{params}). \end{aligned}$$

C. Security notions for self-certified partially blind signature

The security of a self-certified partially blind signature scheme should meet two requirements: the *partial blindness* and the *unforgeability*. We define that a self-certified partially blind signature scheme is secure if it satisfies these two requirements.

First, adopting the similar notion in [8], we define the *partial blindness* of the self-certified partially blind signature.

Definition 2 (Partial Blindness): Let B_0 and B_1 be two honest users that follow the blind signature issuing protocol in a self-certified partially blind signature $\mathcal{SCPBS}$ scheme, and let A be a signer that is involved in the following game with B_0 and B_1.

- $(ID_A, pk, sk, sk') \leftarrow \mathsf{KGen}$; the signer A and the trusted system authority SA first use the key generation algorithm to generate the public and private key pair (ID_A, pk, sk, sk') of A.
- $(m_0, m_1, \mathsf{info}) \leftarrow A$; the signer A produces two messages m_0 and m_1, together with an agreed information info.
- Choose a random bit $b \in \{0, 1\}$, and place m_b and m_{1-b} on the private input tapes of B_0 and B_1, respectively, where b is not disclosed to the signer A. Besides this, $(\mathsf{info}, ID_A, pk)$ are placed on the public input tapes of B_0 and B_1, respectively.
- The signer A engages in the blind signature issuing protocol with B_0 and B_1 in arbitrary order.
- If B_0 and B_1 output $(\mathsf{info}, m_b, \sigma_b)$ and $(\mathsf{info}, m_{1-b}, \sigma_{1-b})$ on their private tapes, respectively, then those outputs are given to A. Otherwise, $\perp$ is given to A.
- The signer A outputs $b' \in \{0, 1\}$ as the guess of b. A wins the game if $b' = b$.

We define the advantage of A as

$$\mathbf{Adv}^{\mathsf{PB}}_{\mathcal{SCPBS}}(A) = |2\Pr[b' = b] - 1|$$

where $\Pr[b' = b]$ denotes the probability that $b' = b$. We say a $\mathcal{SCPBS}$ scheme is partially blind if the advantage $\mathbf{Adv}^{\mathsf{PB}}_{\mathcal{SCPBS}}(A)$ is negligible in the game.

Security against existential forgery under chosen message attack. For digital signatures, the well-known strong security notion is *existential forgery against adaptive chosen message attack* ($\mathsf{EF\text{-}CMA}$) introduced in [20]. Therefore, with respect to the *unforgeability* of $\mathcal{SCPBS}$ scheme, we will define it in the same line. In the random oracle model, we consider an $\mathsf{EF\text{-}CMA}$ adversary $\mathcal{A}$ as follows: The adversary $\mathcal{A}$ is fed with the system parameters params and the signer A's identity ID_A and public key pk; also allowed to access to the signing oracle $\mathcal{O}_S$ and the random oracle $\mathcal{O}_H$. In the end, the adversary $\mathcal{A}$ returns a new valid signature-message pair $(\sigma^\star, m^\star)$. There is a natural restriction that the signature $\sigma^\star$ has not been obtained from the signing oracle $\mathcal{O}_S$ before.

Definition 3 (Unforgeability): Let $\mathcal{SCPBS}$ be a self-certified partially blind signature scheme, let $\mathcal{A}$ be an $\mathsf{EF\text{-}CMA}$ adversary against $\mathcal{SCPBS}$ scheme. We consider the following random experiment, where k is the security parameter:

```
Experiment $\mathbf{Exp}^{\mathsf{EF\text{-}CMA}}_{\mathcal{SCPBS},\mathcal{A}}(k)$
  1. $(\mathsf{params}, \mathsf{masterkey}) \leftarrow \mathsf{Setup}(k)$,
  2. $(ID_A, pk, sk, sk') \leftarrow \mathsf{KGen}(ID_A, \mathsf{params}, \mathsf{masterkey})$
  3. $(\sigma^\star, m^\star) \leftarrow \mathcal{A}^{\mathcal{O}_H, \mathcal{O}_S}(ID_A, pk, \mathsf{params}, \mathsf{info})$
  4. return $\mathsf{Verify}(\sigma^\star, m^\star, \mathsf{info}, ID_A, pk, \mathsf{params})$
```

We then define the success probability of $\mathcal{A}$ via

$$\mathbf{Succ}^{\mathsf{EF\text{-}CMA}}_{\mathcal{SCPBS},\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{EF\text{-}CMA}}_{\mathcal{SCPBS},\mathcal{A}}(k) = 1\right]$$

Let $\tau \in \mathbb{N}$ and $\epsilon \in [0, 1]$. We say that the $\mathcal{SCPBS}$ is (τ, ϵ)-secure if no $\mathsf{EF\text{-}CMA}$ adversary $\mathcal{A}$ running in time τ has a success $\mathbf{Succ}^{\mathsf{EF\text{-}CMA}}_{\mathcal{SCPBS},\mathcal{A}}(k) \geq \epsilon$.

III. Basic concepts on bilinear pairings

Bilinear pairing is an important cryptographic primitive and has recently been applied in many positive applications in cryptography [16], [17]. Let $\mathbb{G}_1$ be a cyclic additive group and $\mathbb{G}_2$ be a cyclic multiplicative group of the same prime order q. We assume that the discrete logarithm problems in both $\mathbb{G}_1$ and $\mathbb{G}_2$ are hard. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ which satisfies the following properties:

- *Bilinear*: $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in \mathbb{G}_1$, and $a, b \in \mathbb{Z}_q^*$.
- *Non-degenerate*: There exists $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1_{\mathbb{G}_2}$.
- *Computability*: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

From [16], we note that such a bilinear pairing can be realized using the modified Weil pairing associated with supersingular elliptic curve. For instance, let p be a prime such that $p = 6q - 1$ for some prime $q > 3$. Let $\mathbb{E}$ be a supersingular curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$. The group of rational points $\mathbb{E}(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : (x, y) \in \mathbb{E}\}$ forms a cyclic group of order $p+1$. Because the prime q satisfies the condition $6q = p + 1$, the group of points order q in $\mathbb{E}(\mathbb{F}_p)$ also form a cyclic subgroup, namely $\mathbb{G}_1$. Let P be the generator of $\mathbb{G}_1$, and $\mathbb{G}_2$ be the subgroup of $\mathbb{F}_{p^2}$ containing all elements of order q. Then, a bilinear pairing e is a computable map between $\mathbb{G}_1$ and $\mathbb{G}_2$. We define the general bilinear parameter generator $\mathcal{G}en$ as follows.

Definition 4 (Bilinear Parameter Generator): A bilinear parameter generator $\mathcal{G}en$ is a probabilistic algorithm that takes a security parameter k as input and outputs a 5-tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ as the bilinear parameters, including a prime number q with $|q| = k$, two cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ of the same order q, an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$.

Next, we review two related underlying mathematics problems in $\mathbb{G}_1$, which will serve as the basis for our proposed $\mathcal{SCPBS}$ scheme.

- **Computational Diffie-Hellman (CDH) Problem.** Given $P, aP, bP \in \mathbb{G}_1$, for unknown $a, b \in \mathbb{Z}_q^*$, compute $abP \in \mathbb{G}_1$.
- **Decisional Diffie-Hellman (DDH) Problem.** Given $P, aP, bP, cP \in \mathbb{G}_1$, for unknown $a, b, c \in \mathbb{Z}_q^*$, decide whether $c = ab \bmod q$. It is known that DDH problem in $\mathbb{G}_1$ is easy and can be solved in polynomial time according to $e(P, P)^{ab} {=} e(P, P)^c$ [16].

Definition 5 (CDH Assumption): Let $\mathcal{G}en$ be a bilinear parameter generator, and $\mathcal{A}$ an adversary that takes as input a 5-tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ generated by $\mathcal{G}en$ and $(X, Y) \in \mathbb{G}_1^2$. $\mathcal{A}$ returns an element $Z \in \mathbb{G}_1$. We consider the following random experiment, where k is a security parameter.

```
Experiment $\mathbf{Exp}^{\mathsf{CDH}}_{\mathcal{G}en,\mathcal{A}}(k)$
1. $(q, \mathbb{G}_1, \mathbb{G}_2, e, P) \leftarrow \mathcal{G}en(k)$,
2. $x \xleftarrow{R} \mathbb{Z}_q^*$, $X = xP$,
3. $y \xleftarrow{R} \mathbb{Z}_q^*$, $Y = yP$,
4. $Z \leftarrow \mathcal{A}(q, \mathbb{G}_1, \mathbb{G}_2, e, P, X, Y)$
5. return 1 if $Z = xyP$, 0 otherwise
```

We define the corresponding success probability of $\mathcal{A}$ in solving the CDH problem via

$$\mathbf{Succ}^{\mathsf{CDH}}_{\mathcal{G}en,\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{CDH}}_{\mathcal{G}en,\mathcal{A}}(k) = 1\right]$$

Let $\tau \in \mathbb{N}$ and $\epsilon \in [0, 1]$. We say that the CDH is (τ, ϵ)-secure if no polynomial algorithm $\mathcal{A}$ running in time τ has success $\mathbf{Succ}^{\mathsf{CDH}}_{\mathcal{G}en,\mathcal{A}}(k) \geq \epsilon$.

IV. Self-certified partially blind signature $\mathcal{SCPBS}$ scheme

In this section, we propose the self-certified partially blind signature $\mathcal{SCPBS}$ scheme. The details are as follows.

Setup: Given the security parameter k, the trusted system authority SA first generates 5-tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ by running the bilinear parameter generator $\mathcal{G}en(k)$, then chooses a random number $s \xleftarrow{R} \mathbb{Z}_q^*$ as $\mathsf{masterkey}$ kept by himself, and computes $P_{pub} = sP$. Next, SA also picks two cryptographic hash functions $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$, $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, and outputs the system parameters params=$(q, \mathbb{G}_1, \mathbb{G}_2, e, P, H, H_1, P_{pub})$ in the end.

KGen: When the signer A (with his identity ID_A) wants to join the system, A first chooses a random number $x_A \xleftarrow{R} \mathbb{Z}_q^*$ as his partial private key sk and computes the public key $pk = P_A = x_A P$. Then, A sends (ID_A, P_A) to the trusted system authority SA. After receiving (ID_A, P_A), SA uses the $\mathsf{masterkey}$ s to compute the partial private key $sk' = d_A = sH(ID_A \| P_A)$ and returns it to A via a secure channel[1]. Clearly, A can check the validity of d_A by the equation $e(d_A, P) = e(H(ID_A \| P_A), P_{pub})$, since the partial private key d_A is actually a BLS short signature due to Boneh et al. [17]. In the end, the private key of A is (x_A, d_A). Note that this algorithm here is very similar as the key generation algorithm of Shao's self-certified signature scheme proposed in [15].

BSign: The signer A and a user B first negotiate an agreed common information $\mathsf{info} = \Delta$. Then, to obtain a blind signature on message m, as shown in Figure. 1, they execute the following steps:

- Step 1: The signer A first chooses a random number $r \xleftarrow{R} \mathbb{Z}_q^*$, and computes R' and S', where

$$\begin{aligned} R' &= rP \\ S' &= rH(ID_A \| P_A) \end{aligned} \tag{1}$$

[1] SA can send d_A to A without using a secure channel as follows: he sends $D_A = d_A + sP_A$ to A, and then A recovers $d_A = D_A - x_A P_{pub}$, since $sP_A = x_A P_{pub} = x_A sP$.

Signer A		User B
	$\leftarrow$ info $= \Delta \rightarrow$	
$r \xleftarrow{R} \mathbb{Z}_q^*$ $R' = rP, S' = rH(ID_A \Vert P_A)$		
	$\xrightarrow{R',S'}$	
		$\alpha, \beta, \gamma \xleftarrow{R} \mathbb{Z}_q^*$ $R = \alpha R' + \gamma(P_{pub} + P_A)$ $S = \alpha S' + \alpha\beta H(ID_A \Vert P_A) - \gamma H(\Delta)$ $h = \alpha^{-1} H_1(m, R, S) + \beta \bmod q$
	$\xleftarrow{h}$	
$\bar{\varsigma} = (r + h)(x_A H(ID_A \Vert P_A) + d_A) + rH(\Delta)$		
	$\xrightarrow{\bar{\varsigma}}$	
		$\varsigma = \alpha\bar{\varsigma}$ $\sigma = (R, S, \varsigma)$ is the signature of m.

Fig. 1. Partially blind signature between A and B in $\mathcal{SCPBS}$ scheme

Then, A sends (R', S') to the user B.

- **Step 2**: The user B chooses three random numbers $\alpha, \beta, \gamma \xleftarrow{R} \mathbb{Z}_q^*$, computes R, S and h, where

$$\begin{array}{ll} R & = \alpha R' + \gamma(P_{pub} + P_A) \\ S & = \alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) \\ h & = \alpha^{-1} H_1(m, R, S) + \beta \bmod q \end{array} \tag{2}$$

  Then, B sends h to A.
- **Step 3**: The signer A computes $\bar{\varsigma}$ and sends it to user B, where

$$\begin{array}{ll} \bar{\varsigma} & = (r + h)(x_A H(ID_A \| P_A) + d_A) + rH(\Delta) \\ & = (r + h)(x_A + s)H(ID_A \| P_A) + rH(\Delta) \end{array} \tag{3}$$

- **Step 4**: Finally, the user B unblinds the received $\bar{\varsigma}$ as ς, where

$$\varsigma = \alpha\bar{\varsigma} = \alpha(r+h)(x_A+s)H(ID_A \| P_A) + \alpha r H(\Delta) \tag{4}$$

  In the end, the resulting signature for message m and the agreed information Δ is $\sigma = (R, S, \varsigma)$.

Verify: The validity of the signature $\sigma = (R, S, \varsigma)$ can be checked by the following equality

$$\begin{array}{ll} e(\varsigma, P) = & e(S + H_1(m, R, S) \\ & H(ID_A \| P_A), P_{pub} + P_A)e(H(\Delta), R) \end{array} \tag{5}$$

If it does hold, the signature $\sigma = (R, S, \varsigma)$ can be accepted as valid, otherwise it is rejected. Since

$$\begin{array}{l} \quad e(S + H_1(m, R, S)H(ID_A \| P_A), P_{pub} + P_A)e(H(\Delta), R) \\ = e(\alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) + H_1(m, R, S) \\ \quad H(ID_A \| P_A), (s + x_A)P)e(H(\Delta), \alpha R' + \gamma(P_{pub} + P_A)) \\ = e(\alpha r H(ID_A \| P_A) + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) + \\ \quad H_1(m, R, S)H(ID_A \| P_A), (s + x_A)P) \\ \quad e(H(\Delta), \alpha r P + \gamma(P_{pub} + P_A)) \\ = e(\alpha r H(ID_A \| P_A) + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) + \\ \quad H_1(m, R, S)H(ID_A \| P_A), (s + x_A)P) \\ \quad e(H(\Delta), \alpha r P) \cdot e(H(\Delta), \gamma(P_{pub} + P_A)) \\ = e(\alpha r H(ID_A \| P_A) + \alpha\beta H(ID_A \| P_A) + H_1(m, R, S) \\ \quad H(ID_A \| P_A), (s + x_A)P)e(H(\Delta), \alpha r P) \\ = e((\alpha r + \alpha\beta + H_1(m, R, S))H(ID_A \| P_A), (s + x_A)P) \\ \quad e(\alpha r H(\Delta), P) \\ = e((\alpha r + \alpha h)H(ID_A \| P_A), (s + x_A)P) \cdot e(\alpha r H(\Delta), P) \\ = e(\alpha(r + h)(s + x_A)H(ID_A \| P_A), P) \cdot e(\alpha r H(\Delta), P) \\ = e(\alpha(r + h)(s + x_A)H(ID_A \| P_A) + \alpha r H(\Delta), P) \\ = e(\varsigma, P) \qquad \square \end{array}$$

Next, we investigate the performance of the proposed $\mathcal{SCPBS}$ scheme in terms of its time complexity for KGen, BSign and Verify algorithms. For convenience, the notations used in the time complexity analysis are presented first as follows: T_{pmul} represents the time for one point multiplication computation in $\mathbb{G}_1$; T_{padd} represents the time for one point addition computation in $\mathbb{G}_1$; T_{pair} denotes the time for one pairing computation and T_{mhash} denotes the time for one Map2Point hash function. Note that the time complexity for other computation operations, such as the multiplication in $\mathbb{Z}_q^*$, the multiplication in $\mathbb{G}_2$, and ordinary hash operation H_1, are ignored, since they are much smaller than T_{pmul}, T_{padd}, T_{pair}, and T_{mhash}. We summarize the time complexity of our proposed $\mathcal{SCPBS}$ scheme in Table I, which shows that the time complexity in KGen, BSign and Verify algorithms is $2T_{\mathsf{smul}} + 2T_{\mathsf{pair}} + 2T_{\mathsf{mhash}}$, $11T_{\mathsf{smul}} + 6T_{\mathsf{padd}} + 2T_{\mathsf{mhash}}$ and $T_{\mathsf{pmul}} + 2T_{\mathsf{padd}} + 3T_{\mathsf{pair}} + 2T_{\mathsf{mhash}}$ respectively.

TABLE I
TIME COMPLEXITY FOR OUR PROPOSED $\mathcal{SCPBS}$ SCHEME

	KGen	BSign	Verify
SA	$T_{\mathsf{pmul}} + T_{\mathsf{mhash}}$		
Signer A	$T_{\mathsf{pmul}} + 2T_{\mathsf{pair}} + T_{\mathsf{mhash}}$	$5T_{\mathsf{pmul}} + 2T_{\mathsf{padd}} + T_{\mathsf{mhash}}$	
User B		$6T_{\mathsf{pmul}} + 4T_{\mathsf{padd}} + T_{\mathsf{mhash}}$	
Verifier			$T_{\mathsf{pmul}} + 2T_{\mathsf{padd}} + 3T_{\mathsf{pair}} + 2T_{\mathsf{mhash}}$

For many blind signature applications, the speed of signature verification is crucial for determining the feasibility of a blind signature scheme. As shown in Table I, It can be seen that the proposed $\mathcal{SCPBS}$ scheme is efficient.

V. SECURITY ANALYSIS

In this section, we study the security of the proposed $\mathcal{SCPBS}$ scheme, and the security results are given in the following two theorems.

Theorem 1. The proposed $\mathcal{SCPBS}$ scheme is partially blind.

Proof. We consider the signer A in the game defined in Definition 2. Suppose A is given $\perp$ in step 5 of the game, A determines b with a probability $\frac{1}{2}$, which is exactly the same as a random guess of b.

Suppose that A gets signatures (Δ, m_0, σ_0) and (Δ, m_1, σ_1) instead of $\perp$ in step 5 of the game. For $i \in \{0, 1\}$, let $(R_i', S_i', h_i, \bar{\varsigma}_i)$ be the view of data exchanged during the signature issuing protocol, and $(R_0, S_0, \varsigma_0, m_0, \Delta)$ and $(R_1, S_1, \varsigma_1, m_1, \Delta)$ are given to A.

In order to prove the partial blindness, we will show that given a valid signature (R, S, ς, m, c) and any view $(R', S', h, \bar{\varsigma})$, there always exists a unique tuple of blind factors $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$. And since the blind factors are chosen randomly, the partial blindness of the $\mathcal{SCPBS}$ scheme will naturally satisfy. Our proof is similar to that in [6]. Given a valid signature (R, S, ς, m, c) and any view $(R', S', h, \bar{\varsigma})$, the following must hold for $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$:

$$\varsigma = \alpha\bar{\varsigma} \tag{6}$$

$$h = \alpha^{-1}H_1(m, R, S) + \beta \bmod q \tag{7}$$

$$R = \alpha R' + \gamma(P_{pub} + P_A) \tag{8}$$

$$S = \alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) \tag{9}$$

From Eq. (6), it is obvious that an $\alpha = \log_{\bar{\varsigma}} \varsigma \in \mathbb{Z}_q^*$ exists uniquely. Then, we can also get a unique $\beta = h - \alpha^{-1}H_1(m, R, S) \bmod q$ from Eq. (7), and a unique $\gamma = \log_{(P_{pub}+P_A)}(R - \alpha R')$ from Eq. (8).

Next, we need to show that such α, β and γ will also satisfy Eq. (9). Here, due to the Non-degenerate of the bilinear pairing, we know

$$\begin{aligned}
& S = \alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta) \\
\Rightarrow\ & e(S, P_{pub} + P_A) \\
& = e(\alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta), P_{pub} + P_A)
\end{aligned}$$

Therefore, we only need to show α, β and γ satisfy the following equality

$$\begin{aligned}
& e(\alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta), P_{pub} + P_A) \\
& = e(S, P_{pub} + P_A)
\end{aligned}$$

On the other hand, because of the validity of signature (R, S, ς, m, c), we have

$$\begin{aligned}
e(\varsigma, P) = e(S + H_1(m, R, S) \\
H(ID_A \| P_A), P_{pub} + P_A) e(H(\Delta), R)
\end{aligned}$$

that is,

$$\begin{aligned}
& e\left(H_1(m, R, S) H(ID_A \| P_A), P_{pub} + P_A\right)^{-1} \\
= \quad & e\left(\varsigma, P\right)^{-1} e\left(S, P_{pub} + P_A\right) e\left(H(\Delta), R\right)
\end{aligned}$$

Then,

$$\begin{aligned}
& e(\alpha S' + \alpha\beta H(ID_A \| P_A) - \gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\alpha S' + \alpha(\beta H(ID_A \| P_A) - \alpha^{-1}\gamma H(\Delta)), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot S' + \log_{\bar{\varsigma}} \varsigma \cdot (\beta H(ID_A \| P_A) - (\log_{\bar{\varsigma}} \varsigma)^{-1} \cdot \\
& \gamma H(\Delta)), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot S' + \log_{\bar{\varsigma}} \varsigma \cdot \beta H(ID_A \| P_A), P_{pub} + P_A) \cdot \\
& e(-\gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot S' + \log_{\bar{\varsigma}} \varsigma \cdot (h - \alpha^{-1}H_1(m, R, S)) \\
& H(ID_A \| P_A), P_{pub} + P_A) \cdot e(-\gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot (r + h) H(ID_A \| P_A), P_{pub} + P_A) \\
& e(\log_{\bar{\varsigma}} \varsigma \cdot \alpha^{-1} H_1(m, R, S) H(ID_A \| P_A), P_{pub} + P_A)^{-1} \\
& e(-\gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot (r + h)(s + x_A) H(ID_A \| P_A), P) \\
& e(H_1(m, R, S) H(ID_A \| P_A), P_{pub} + P_A)^{-1} \\
& e(-\gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot (r + h)(s + x_A) H(ID_A \| P_A), P) \\
& e\left(\varsigma, P\right)^{-1} e\left(S, P_{pub} + P_A\right) e\left(H(\Delta), R\right) \\
& e(-\gamma H(\Delta), P_{pub} + P_A) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot (r + h)(s + x_A) H(ID_A \| P_A), P) \cdot e\left(\varsigma, P\right)^{-1} \\
& e\left(S, P_{pub} + P_A\right) e\left(H(\Delta), \log_{\bar{\varsigma}} \varsigma \cdot R'\right) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot (r + h)(s + x_A) H(ID_A \| P_A), P) \\
& e\left(\log_{\bar{\varsigma}} \varsigma \cdot r H(\Delta), P\right) \cdot e\left(\varsigma, P\right)^{-1} e\left(S, P_{pub} + P_A\right) \\
= \quad & e(\log_{\bar{\varsigma}} \varsigma \cdot ((r + h)(s + x_A) H(ID_A \| P_A) + r H(\Delta)), P) \\
& e\left(\varsigma, P\right)^{-1} e\left(S, P_{pub} + P_A\right) \\
= \quad & e(\varsigma, P) e(\varsigma, P)^{-1} e\left(S, P_{pub} + P_A\right) \\
= \quad & e(S, P_{pub} + P_A)
\end{aligned}$$

Hence, from the above deduction, the blind factors α, β and γ always exist which lead to the same relation defined in the blind signature issuing protocol.

Thus, going back to Step 6 of the game defined in definition 2, the signer A succeeds in determining b with probability $\frac{1}{2}$.

Finally, taking these two cases into account, the probability that A wins the game is $\frac{1}{2}$. Therefore, our proposed $\mathcal{SCPBS}$ scheme is partially blind. This completes the proof. □

Theorem 2. Let $\mathcal{G}en$ be a bilinear parameter generator, and assume that the hash function H and H_1 are random oracles. Let $\mathcal{A}$ be an EF-CMA adversary against the $\mathcal{SCPBS}$ scheme in the random oracle model, that produces an existential forgery with probability $\epsilon = \mathbf{Succ}^{\textsf{EF-CMA}}_{\mathcal{SCPBS},\mathcal{A}}$, within running time τ, making q_{h_1} and q_s queries to the random oracle $\mathcal{O}_{H_1}$ and to the signing oracle $\mathcal{O}_S$. If $\epsilon \geq 10(q_s + 1)(q_s + q_{h_1})/q$, then the CDH problem in $\mathbb{G}_1$ can be resolved with expected time $\tau' \leq 120686 q_{h_1} \tau / \epsilon$.

Proof. Adopting the game simulation approach due to Shoup [19], we define a sequence of game $\textsf{Game}_0$, $\textsf{Game}_1, \cdots$, of modified attacks starting from the actual EF-CMA adversary $\mathcal{A}$. All the games operate on the same underlying probability space: the public and private keys, the coin tosses of $\mathcal{A}$ and the random oracles. Let $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ be a 5-tuple generated by $\mathcal{G}en(k)$, where k is the security parameter. Let $H : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be two cryptographic hash functions, and both behave as the random oracle in the simulation. Let $(X^* = xP, Y^* = yP)$ be a random instance of CDH problem. We will use the adversary $\mathcal{A}$'s capability to compute xyP.

$\textsf{Game}_0$: This is an actual game in the random oracle model. The KGen algorithm generates params=$(q, \mathbb{G}_1, \mathbb{G}_2, e, P, H, H_1, P_{pub})$ and signer A's public and private key $(ID_A, P_A, x_A, d_A = sH(ID_A \| P_A))$. Then, the adversary $\mathcal{A}$ is fed with params=$(q, \mathbb{G}_1, \mathbb{G}_2, e, P, H, H_1, P_{pub})$ and $ID_A, P_A, H(ID_A \| P_A)$ and an agreed common information info $= \Delta$. $\mathcal{A}$ is also allowed to access a random oracle $\mathcal{O}_{H_1}$ and a signing oracle $\mathcal{O}_S$. In the end, the adversary $\mathcal{A}$ outputs its blind signature forgery $(\sigma^\star, m^\star)$, then we check whether it is a valid signature or not. We denote $\textsf{Forge}_0$ to be the event that the forged signature $(\sigma^\star, m^\star)$ is valid and use the notation $\textsf{Forge}_j$ for the same meaning in any game $\textsf{Game}_i$. By definition, we have

$$\epsilon = \mathbf{Succ}^{\textsf{EF-CMA}}_{\mathcal{SCPBS},\mathcal{A}} = \Pr[\textsf{Forge}_0]. \tag{10}$$

$\textsf{Game}_1$: In this game, we implant the challenge $X^* = xP$ to the simulation. We first choose two elements $P_1, P_2 \in \mathbb{G}_1$ such that $P_1 + P_2 = X^* = xP$. Then, we set $P_{pub} = P_1$ and $P_A = P_2$. Since P_{pub} and P_A are uniformly distributed in $\mathbb{G}_1$, the probability distribution is unchanged. Therefore, we have

$$\Pr[\textsf{Forge}_1] = \Pr[\textsf{Forge}_0]. \tag{11}$$

Game$_2$: In this game, we implant the challenge $Y^* = yP$ to the simulation. Because H behaves as a random oracle, when we set $H(ID_A \| P_A) = Y^* = yP$ and $H(\Delta)$ be a random element in $\mathbb{G}_1$, this game is identical to the previous one in the random oracle model. Hence, we have

$$\Pr[\mathsf{Forge}_2] = \Pr[\mathsf{Forge}_1]. \tag{12}$$

Game$_3$: In this game, we will simulate the random oracle $\mathcal{O}_{H_1}$ by maintaining a Λ_{H_1}-list. When a fresh query (m, R, S) is queried, we choose a random number $v \xleftarrow{R} \mathbb{Z}_q^*$ and compute $H_1(m, R, S) = v$ We then store (m, R, S, v) in the Λ_{H_1}-list and return $H_1(m, R, S)$ as the answer to the oracle query. Clearly, in the random oracle model, v is randomly chosen from $\mathbb{Z}_q^*$, then $H_1(m, R, S)$ is uniformly distributed in $\mathbb{G}_1$, and this game is therefore perfectly indistinguishable from the previous one. Hence,

$$\Pr[\mathsf{Forge}_3] = \Pr[\mathsf{Forge}_2]. \tag{13}$$

Game$_4$: In this game, we simulate the random oracle $\mathcal{O}_S$. When a fresh query m is asked, we will proceed as follows: First, we choose three random numbers $u, v, w \xleftarrow{R} \mathbb{Z}_q^*$, and compute R, S and ς, where

$$\begin{aligned} R &= wP \\ S &= uP - vH(ID_A \| P_A) = uP - vyP \\ \varsigma &= u(P_{pub} + P_A) + wH(c) = uxP + wH(\Delta) \end{aligned}$$

Then, we set $H_1(m, R, S) = v$ and store (m, R, S, v) in the Λ_{H_1}-list. In the end, we return (R, S, ς) as a signature σ on message m to $\mathcal{A}$. In the random oracle model, this game is also identical to the previous one. Hence,

$$\Pr[\mathsf{Forge}_4] = \Pr[\mathsf{Forge}_3]. \tag{14}$$

At the end of Game_4, the adversary $\mathcal{A}$ eventually outputs a new valid signature-message pair $(\sigma^\star, m^\star)$. From the forking lemma due to Pointcheval and Stern [18], if $\Pr[\mathsf{Forge}_4] \geq 10(q_s + 1)(q_s + q_{h_1})/q$, then $\epsilon = \mathbf{Succ}^{\mathsf{EF\text{-}CMA}}_{\mathcal{SCPBS},\mathcal{A}} \geq 10(q_s + 1)(q_s + q_{h_1})/q$, and then by replaying $\mathcal{A}$ with the same tape but different choices of H_1, $\mathcal{A}$ outputs two valid signatures $\sigma^\star = (R, S, \varsigma)$ and $\sigma'^\star = (R, S, \varsigma')$ on the same message $m^\star$, where $\varsigma = \alpha(r + \alpha^{-1}H_1(m, R, S) + \beta)(x_A + s)H(ID_A \| P_A) + \alpha r H(\Delta)$ and $\varsigma' = \alpha(r + \alpha^{-1}H_1'(m, R, S) + \beta)(x_A + s)H(ID_A \| P_A) + \alpha r H(\Delta)$. Since $H_1(m, R, S) \neq H_1'(m, R, S)$, we can compute

$$\begin{aligned} xyP &= (x_A + s)H(ID_A \| P_A) \\ &= \tfrac{1}{H_1(m,R,S) - H_1'(m,R,S)} \cdot (\varsigma - \varsigma') \end{aligned}$$

and output it as the CDH problem challenge. The total running time τ' of solving the CDH problem is equal to the running time of the forking lemma, which is bounded by $120686 q_{h_1} \tau / \epsilon$, as desired. This concludes the proof. □

VI. Conclusions

In this paper, we have studied self-certified partially blind signature. Firstly, we formalized the definition and security notions for self-certified partially blind signature, and then proposed a novel and effective $\mathcal{SCPBS}$ scheme based on the bilinear pairing. We have discussed its partial blindness, and analyzed its unforgeability through the technique of provable security. We have proved that the proposed $\mathcal{SCPBS}$ scheme can truly generate a secure self-certified partially blind signature, which should solidly contribute to the development of electronic cash systems.

References

[1] D. Chaum, "Blind signatures for untraceable payments," *in Proc. Advances in Cryptology - Crypto'82*, Santa Barbara, California, USA, pp. 199-203, Aug. 1982.

[2] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, Vol. 28, No. 10, pp. 1030-1044, 1985.

[3] D. Chaum, "Privacy protected payments: unconditional payer and/or payee untraceability," *in Proc. Smartcard 2000*, North Holland, Amsterdam, 1988.

[4] R. Lu, Z. Cao, and Y. Zhou, "Proxy blind multi-signature scheme without a secure channel," *Applied Mathematics and Computation*, Vol. 164, No. 1, pp. 179-187, 2005.

[5] X. Lin and Y. Yang, "A blind signature scheme based on Lucas sequence," *Journal of Beijing University of Posts and Telecommunications*, Vol. 21, No. 1, pp. 88-91, 1998.

[6] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," *in Proc. Australasian conference on Information Security and Pricacy - ACISP 2003*, LNCS 2727, Springer-Verlag, Wollongong, Australia, pp. 312-323, 2003.

[7] M. Abe and E. Fujisaki, "How to date blind signatures," *in Proc. Advances in Cryptology - Asiacrypt'96*, LNCS 1163, Springer-Verlag, Kyongju, Korea, pp. 244-251, 1996.

[8] M. Abe and T. Okamto, "Provably secure partially blind signatures," *in Advances in Cryptology - Crypto'00*, LNCS 1880, Springer-Verlag, Santa Barbara, California, USA, pp. 271-286, 2000.

[9] W. Juang and C. Lei, "Partially blind threshold signatures based on discrete logarithm," *Computer Communications*, Vol. 22, No. 1, pp. 73-86, 2003.

[10] F. Zhang and X. Chen, "Cryptanalysis of Huang-Chang partially blind signature scheme," *Journal of Systems and Software*, Vol. 76, No. 3, pp. 323-325, 2005.

[11] Sherman S.M. Chow, Lucas C.K. Hui, S.M. Yiu, and K.P. Chow, "Two improved partially blind signature schemes from bilinear pairings," *in Proc. Australasian conference on Information Security and Pricacy - ACISP 2005*, LNCS 3574, Springer-Verlag, Brisbane, Australia, pp. 316-328, 2005.

[12] M. Girault, "Self-certified public keys," *in Proc. Advances in Cryptology - Eurocrypt'91*, LNCS 1440, Springer-Verlag, Brighton, UK, pp. 491-497, 1991.

[13] IEEE P1363: Standard specifications for public key cryptography. Jan. 2000.

[14] A. Shamir, "Identity-based cryptosystems and signature schemes," *in Proc. Advances in Cryptology - Crypto'84*, LNCS 196, Springer-Verlag, Santa Barbara, California, USA, pp. 47-53, 1984.

[15] Z. Shao, "Self-certified signature scheme from pairings," *Journal of Systems and Software*, Vol. 80, No. 3, pp. 388-395, 2007.

[16] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *in Proc. Advances in Cryptology - Crypto'01*, LNCS 2139, Springer-Verlag, Santa Barbara, California, USA, pp. 213-229, 2001.

[17] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *in Proc. Advances in Cryptology - Asiacrypt'01*, LNCS 2248, Springer-Verlag, Gold Coast, Australia, pp. 514-532, 2001.

[18] D. Pointcheval and J. Stern, "Security arguments for digit signatures and blind signatures," *Journal of Cryptology*, Vol. 13, No. 3, pp. 361-396, 2000.

[19] V. Shoup, "OAEP reconsidered," *Journal of Cryptology*, Vol. 15, No. 4, pp. 223-249, 2002.

[20] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptively chosen message attacks," *SIAM Journal on Computing*, Vol. 17, No. 2, pp. 281-308, 1988.