Workshop on Cryptology and Information Security

Nearly Optimal User Efficient Partially Blind Signatures

Chun-I Fan

Telecommunication Laboratories
Chunghwa Telecom Co., Ltd.
12, Lane 551, Min-Tsu Road Sec. 5
Yang-Mei, Tao-Yuan, Taiwan 326, R.O.C.
TEL: +886-3-424-5081
FAX: +886-3-424-4147
Email: chunifan@ms35.hinet.net

Corresponding Author: Chun-I Fan Ph.D.
Postal Address: P.O. Box 8-210, Shin-Juang, Taiwan 242
E-mail Address: chunifan@ms35.hinet.net
TEL: +886-3-424-5081
FAX: +886-3-424-4147

Abstract

A low-computation partially blind signature scheme was proposed in 1998. However, a weakness on the partial blindness property has been found in the scheme. This manuscript presents a new user efficient partially blind signature protocol to repair the weakness. Furthermore, the proposed scheme has been demonstrated as being nearly optimal in users' computations.

Keywords: Partially blind signatures, electronic cash, Security & privacy, Cryptography

1 Introduction

A blind signature scheme contains two types of roles, a signer and a group of users. A user requests signatures from the signer, and the signer computes and issues blind signatures to the users. There are two sets of messages known to the signer: (1) the signing results actually computed by the signer and (2) the signatures shown by the users for verification later. The key point is that the actual correspondence between these two sets of messages is unknown to the signer. This property is usually referred to as the *unlinkability* (or *blindness*) property. Owing to the unlinkability property, blind signature techniques have been widely used in untraceable electronic cash and anonymous voting systems proposed [3, 9].

Due to the characteristics of electronics, e-cash can be easily duplicated. Hence, to prevent a payer from double-spending his e-cash, the bank has to keep a database which stores all spent e-cash to check whether a specified e-cash has been spent or not by searching this database. This operation is referred to as the *freshness checking* (or the *double-spending checking*) of e-cash. Certainly, the database kept by the bank may grow unlimitedly [1].

The techniques of partially blind signatures make it possible to prevent the bank's database from growing unlimitedly. In an electronic cash system based on a partially blind signature scheme, the bank (or signer, respectively) assures that the e-cash (or signatures, respectively) issued by him contains the information he desires, such as the date information [1]. This property is referred to as the *partial blindness* property. By embedding an expiration date into each e-cash issued by the bank, all expired e-cash recorded in the bank's database can be removed. Armed with partial blindness, we can deal with the unlimited growth problem of the bank's database in an electronic cash system.

In [6], a low-computation partially blind signature scheme for electronic cash was proposed. Since only several modular multiplications are performed by a user in the scheme, it is especially suitable for the situations where computation capabilities of users are limited such as smart cards and mobile units. However, a weakness on the partial blindness property is found in the scheme of [6]. In this manuscript, we propose a solution to repair the weakness, and present a new user efficient partially blind signature scheme. Compared to [6], users' computations are further reduced in the proposed scheme, and, furthermore, it is demonstrated as being nearly optimal.

The rest of the paper is organized as follows. In Section 2 we briefly describe and review some related works. In Section 3 we present a new user efficient partially blind signature scheme. The security and performance of the scheme are examined in Section 4 and Section 5, respectively. Finally, a concluding remark is given in Section 6.

2 Preliminary

The concepts of blind signatures was first introduced by Chaum [3]. Based on the RSA cryptosystem, he proposed a blind signature scheme to achieve the unlinkability property. By means of the techniques of blind signatures, an untraceable electronic cash system was proposed in [3]. In such an electronic cash system, the bank (or the signer) issues e-cash, and a customer (or a user) can withdraw e-cash from his account and deposit e-cash into his account in the bank. The authors of [1] proposed a variation of the Chaum's blind signature scheme of [3] to achieve the partial blindness property. Based on the RSA cryptosystem, Ferguson [7] introduced another blind signature scheme tailored for his untraceable electronic cash system. In [2], the authors proposed a blind signature scheme based on discrete logarithm problems, and it is derived from a variation of the DSA [10]. The authors of [2] also presented a blind signature scheme based on the Nyberg-Rueppel signature scheme [11]. Based on the Okamoto's protocol of [12] and the Schnorr's protocol of [18], a blind signature scheme was proposed in [14]. The authors of [14] presented another blind signature scheme based on the Okamoto's protocol of [12] and the Guillou-Quisquater protocol of [8]. In addition, based on the theories of quadratic residues, two blind signature schemes are proposed in [15]. For more details, the readers can refer to [1, 2, 3, 7, 14, 15].

2.1 A Generic Randomized Partially Blind Signature Scheme

In a partially blind signature scheme, the signer assures that each signature issued by him contains an information he desires, and anyone else cannot modify the information embedded in the signature. Under the same embedded information, the signer cannot link a signature to the corresponding instance of signing protocol which produces the corresponding blind signature. This is the partial blindness property.

In general, two parties, a signer and a group of users, participate in a partially blind signature protocol. A set of predefined-format information must be negotiated and agreed by the signer and all users in advance. The corresponding protocol is described below.

<1> **Blinding:** A user blinds a message by performing an encryption-like process on it. The user submits the blinded message to the signer to request the signer's signature on the blinded message.

<2> **Signing:** The signer signs on the combination of the blinded message and a predefined-format information by using its signing function (only the signer knows), and then sends the result, called the partially blind signature since the predefined-format information is clear to the signer, to the user.

<3> **Unblinding:** Finally, the user unblinds the partial blind signature by performing a decryption-like operation, and then he can obtain the signer's signature on the combination of the message and the predefined-format information.

<4> **Verifying:** The signer's signature can be verified by checking if the corresponding public verification formula with the signature, the message, and the predefined-format information as parameters is true.

In a blind signature scheme, Ferguson [7] suggested that the signer had better inject one or more randomization factors into the message on which it is about to sign such that the attackers cannot predict the exact content of the message the signer signs to withstand chosen-text attacks, such as [5, 19]. This is referred to as *randomized* blind signature.

Let M be the underlying set of strings and A be a finite set of strings with the predefined format which is negotiated by the signer and all users in advance. A generic partially blind signature scheme contains five elements (B, H, S, U, V) where

<1>. $H : M \to M$ is a public one-way hash function. Given y, it is computationally infeasible to derive m such that $H(m) = y$, and it also is computationally infeasible to find two distinct messages m_1 and m_2 such that $H(m_1) = H(m_2)$. In a blind signature scheme, the hash function is used to hash a message into a message digest for signing.

<2>. $S : M \times M \times A \to M^K$ is the signing function which is kept secret by the signer where K is a positive integer, $M^K = M^{K-1} \times M$ when $K \geq 2$, and $M^K = M$ when $K = 1$. Given a message $m \in M$, a predefined-format information $a \in A$, and for any $x \in M$, it is computationally infeasible to form $S(H(m), x, a)$ except the signer, where $S(H(m), x, a)$ is the signer's signature on message m with the predefined-format information a and the randomization factor x chosen by the signer at random.

<3>. $V : M^K \times M \times M \times A \to$\{true, false\} is the public verification formula. If $V(t, m, c, a)$ = true, then t is the signer's signature on m with the randomization parameter c and the predefined-format information a. Besides, $V(S(H(m), c, a), m, c, a)$ is always true for each $m \in M$, $c \in M$, and $a \in A$.

<4>. $B : M \times M \to M$ is the blinding function. It is computationally infeasible for the signer to derive m from $B(m, r)$ without r where $r \in M$ is randomly chosen and kept secret by some user. The string r is called the blinding factor of m, and $B(m, r)$ is said to be the blinded message.

<5>. $U : M^K \times M \to M^K$ is the unblinding function. For each $m \in M$, $r \in M$, $x \in M$, and $a \in A$, $U(S(B(H(m), r), x, a), r) = S(H(m), c, a)$ where $c = B(x, u)$ and $u \in M$ is randomly chosen by some user. It is computationally infeasible to derive $S(H(m), c, a)$ from $S(B(H(m), r), x, a)$ without r and u.

The details of the blind signature protocol are described as follows.

<1> **Blinding:** First, a user chooses a message $m \in M$, randomly selects a blinding factor $r \in M$, and prepares a string a according to the predefined format. He computes and submits

$$\alpha = B(H(m), r) \tag{1}$$

and a to the signer.

<2> **Signing:** After verifying a, the signer randomly chooses $x \in M$, and then computes and sends the blind signature $t = S(\alpha, x, a)$ and the randomization factor x to the user.

<3> **Unblinding:** After receiving (t, x), the user derives

$$s = U(t, r) \tag{2}$$

which is $S(H(m), c, a)$ where

$$c = B(x, u) \tag{3}$$

and u is another blinding factor randomly chosen by the user.

<4> **Verifying:** The tuple (s, c) is the signer's signature on message m with randomization parameter c and the predefined information a. The 4-tuple (s, m, c, a) can be verified by checking whether the verification formula

$$V(s, m, c, a) \tag{4}$$

is true or not.

It is information-theoretically impossible for the signer to derive the link between a 4-tuple (s, m, c, a) and the instance of the signing protocol which produces $S(B(H(m), r), x, a)$. Under the same a, all of the 4-tuples are indistinquishable from the signer's point of view as long as the blinding factors r's and u's are kept secret by the users. This is the unlinkability property under the same predefined information.

In a randomized blind signature scheme, the signer has to randomly choose at least one string (such as x in the generic scheme) and inject it into a message before signing on the message. The string is said to be the randomization factor. In a randomized blind signature scheme, at least three parameters are needed to be shown for verification where one is the signature, another is the message, the other is the randomization parameter. The randomization parameter (such as c in the generic scheme) is derived from the randomization factor and some other parameters where at least one parameter (such as u in the generic scheme) is randomly determined by the user, not the signer, for the unlinkability property. The derivation of the randomization parameter (such as $c = B(x, u)$ in the generic scheme) also is a kind of blinding operation.

We assume that multiplications are the major arithmetical computations in the generic randomized partially blind signature scheme. The computations required for a user in the scheme are summarized as follows.

<**1**>. The user performs twice of random-number generation for (r, u), and computes (1) and (3) for blinding where each of them requires at least one multiplication. Besides, one hash operation is needed for preparing the message digest $H(m)$ in the blinded message α.

<**2**>. Besides, the user has to perform the unblinding operation (2) which requires at least one multiplication.

<**3**>. At least four parameters, the signature, the message, the randomization parameter, and the predefined information, i.e., (s, m, c, a), are needed to form a signature tuple.

<**4**>. To verify the 4-tuple (s, m, c, a), the user has to compute at least one hash operation and three multiplications because there are four elements in the signature tuple.

<**5**>. At least four messages, i.e., the blinded message α, the blind signature t, the randomization factor x, and the predefined information a, must be transmitted over the channel between the user and the signer during an instance of the protocol.

From the above, the lower bound of the computations required for a user in a randomized partially blind signature scheme is shown in Table 1.

Table 1: The lower bound of user's computation in a randomized partially blind signature scheme

	Blinding	Unblinding	Verifying	Signature size	Transmitted
Multiplication computation:	2	1	3	–	–
Hashing operation:	1	0	1	–	–
Random-number generation:	2	0	0	–	–
Number of messages:	–	–	–	4	4

2.2 A Low-Computation Partially Blind Signature Scheme

In 1998, a low-computation partially blind signature scheme was proposed [6]. The protocol consists of five stages: initialization, blinding, signing, unblinding, and verifying, described as follows.

<0> **Initialization.** The signer randomly selects two distinct large primes p_1 and p_2 where $p_1 \equiv p_2 \equiv 3 \pmod 4$. The signer computes $n = p_1 p_2$ and publishes n. In addition, let H be a public one-way hash function.

<1> **Blinding.** A user prepares a string a according to a predefined format negotiated and agreed by all users and the signer. The user chooses a message m and two random integers u, v, and computes $\alpha = (H(m)(u^2 + v^2) \bmod n)$. He then submits the tuple (a, α) to the signer.

After verifying that the string a is of the predefined format, the signer randomly selects x such that $(H(a)(\alpha(x^2+1))^3 \bmod n)$ is a quadratic residue (QR) in Z_n^*, and then sends x to the user, where Z_n^* is the set of all positive integers less than and relatively prime to n.

After receiving x, the user randomly selects an integer b, and then computes $\delta = (b^4 \bmod n)$ and $\beta = (\delta(u - vx) \bmod n)$. The user submits the integer β to the signer.

<2> **Signing.** After receiving β, the signer computes $\lambda = (\beta^{-1} \bmod n)$ and derives an integer t in Z_n^* such that

$$t^8 \equiv H(a)(\alpha(x^2+1))^3(\lambda^2)^3 \pmod n \tag{5}$$

The signer sends the tuple (t, λ) to the user.

<3> **Unblinding.** After receiving (t, λ), the user computes $s = (b^3 t \bmod n)$ and $c = (\delta\lambda(ux + v) \bmod n)$. The tuple (s, c) is the signer's signature on m with the predefined information a.

<4> **Verifying.** To verify (s, m, c, a), one can examine if $s^8 \equiv H(a)(H(m)(c^2+1))^3 \pmod n$.

In the above protocol, the user can choose an a' with $a' \neq a$ and find an integer k such that $(8k+1)$ can be divided by 3. He prepares $\alpha = (H(m)(u^2+v^2)H(a)^{-(8k+1)/3}H(a')^{(8k+1)/3} \bmod n)$ and then obtains s in the unblinding stage, where s^8

$$\begin{aligned}
&\equiv (b^3t)^8 \equiv (b^3)^8 H(a)(\alpha(x^2+1))^3(\lambda^2)^3 \equiv H(a)(\alpha(x^2+1))^3((u-vx)^{-2})^3 \\
&\equiv H(a)(H(m)(u^2+v^2)H(a)^{-(8k+1)/3}H(a')^{(8k+1)/3}(x^2+1))^3((u-vx)^{-2})^3 \\
&\equiv H(a)(H(m)(c^2+1)H(a)^{-(8k+1)/3}H(a')^{(8k+1)/3})^3 \\
&\equiv H(a)^{-8k}H(a')^{8k}H(a')(H(m)(c^2+1))^3 \pmod{n}.
\end{aligned}$$

Thus, the user can form a signature 4-tuple (s', m, c, a') with $s' = (sH(a)^kH(a')^{-k} \bmod n)$ such that $(s')^8 \equiv H(a')(H(m)(c^2+1))^3 \pmod{n}$. The user can obtain a signature on m with the predefined-format information $a' \neq a$. This is a weakness on the partial blindness property.

3 A Nearly Optimal User Efficient Partially Blind Signature Scheme

The proposed partially blind signature scheme is based on the theories of quadratic residues [20]. Under a modulus n, x is a quadratic residue (QR) in Z_n^* if and only if there exists an integer y in Z_n^* such that $y^2 \equiv x \pmod{n}$. Given n and x, it is computationally infeasible to derive the square root y of the integer x in Z_n^* if n contains large prime factors and the factorization of n is unknown [16]. The security of our scheme depends on the difficulty of computing a square root of an integer in Z_n^* without the factorization of n. The proposed scheme consists of five stages: initialization, blinding, signing, unblinding, and verifying, are described as follows.

<0> **Initialization.** The signer randomly selects two distinct large primes p_1 and p_2 where $p_1 \equiv p_2 \equiv 3 \pmod{4}$. It then computes $n = p_1p_2$ and publishes n. Since $p_1 \equiv p_2 \equiv 3 \pmod{4}$, given a QR in Z_n^*, there are four different square roots (or 2nd roots) of the QR in Z_n^*, and one of these roots is a QR in Z_n^*, too [20]. Hence, in addition to the 2nd roots of a QR in Z_n^*, we can derive the 4-th roots, 8-th roots, and 2^i-th roots of the QR in Z_n^* where i is a positive integer. Such a special form of primes p_1 and p_2 does not affect the difficulty of factoring n [21]. Besides, let H be a public one-way hash function.

<1> **Blinding.** A user prepares a string a according to the predefined format negotiated and agreed by all users and the signer. The user submits a to the signer.

After verifying that the string a is of the predefined format, the signer randomly selects x in Z_n^* such that $(xH(a) \bmod n)$ is a QR in Z_n^*, and then sends the integer x to the user.

After receiving x, the user chooses a message m and two random integers (r, u). He then computes

$$\begin{cases} c = u^2x \bmod n \\ \alpha = r^2uH(c||m) \bmod n \end{cases} \tag{6}$$

and submits α to the signer where $||$ is the string concatenation operator.

<2> **Signing.** After receiving α, the signer derives an integer t in Z_n^* such that

$$t^4 \equiv (\alpha^2 x H(a))^{-1} \pmod{n} \tag{7}$$

by some efficient algorithms [13, 16]. Hence, the integer t is one of the 4-th roots of $((\alpha^2 x H(a))^{-1} \bmod n)$ in Z_n^*. Thus, the signer sends t to the user.

<3> **Unblinding.** After receiving t, the user computes

$$s = rt \bmod n \tag{8}$$

The triple (s, c, a) is the signer's signature on m.

<4> **Verifying.** To verify (s, m, c, a), one can examine if

$$(s^2 H(c||m))^2 H(a) c \equiv 1 \pmod{n}. \tag{9}$$

The following theorem ensures that the signature 4-tuple (s, m, c, a) produced by the proposed partially blind signature scheme satisfies (9).

Theorem 1 *If (s, m, c, a) is a signature 4-tuple produced by the protocol of Section 3, then (9) is true.*

Proof. By the Chinese remainder theorem [20], each integer w in Z_n^* can be represented by $< w_1, w_2 >$ where $w_1 = (w \bmod p_1)$ and $w_2 = (w \bmod p_2)$. For convenience, $< w_1, w_2 >$ is denoted by $< w >$ sometimes. For each $< k > = < k_1, k_2 >$ and $< w > = < w_1, w_2 >$ in Z_n^*, $< kw \bmod n > = < k_1 w_1 \bmod p_1, k_2 w_2 \bmod p_2 >$, and $< k^{-1} \bmod n > = < k_1^{-1} \bmod p_1, k_2^{-1} \bmod p_2 >$. Besides, for each $< k_1, k_2 >$ and $< w_1, w_2 >$ in Z_n^*, $< k_1, k_2 > = < w_1, w_2 >$ if and only if $k_1 \equiv w_1 \pmod{p_1}$ and $k_2 \equiv w_2 \pmod{p_2}$.

Since both $(\alpha^2 \bmod n)$ and $(xH(a) \bmod n)$ are QRs in Z_n^*, we have that $(\alpha^2 x H(a))^{-1} \equiv r^{-4}(cH(a)H(c||m)^2)^{-1} \pmod{n}$ is a QR in Z_n^*. Let $y = ((cH(a)H(c||m)^2)^{-1} \bmod n)$. Since $(r^{-4} \bmod n)$ is a QR in Z_n^*, the integer y is also a QR in Z_n^*. Let $< d_1, d_2 >$ be one of the 4-th roots of y in Z_n^*. Thus, the four 4-th roots of y in Z_n^* are $< \pm d_1, \pm d_2 >$, and the four 4-th roots of $(r^{-4} y \bmod n)$ in Z_n^* are $< \pm r_1^{-1} d_1, \pm r_2^{-1} d_2 >$. By (7), $t^4 \equiv r^{-4} y \pmod{n}$, so that $t \in \{< \pm r_1^{-1} d_1, \pm r_2^{-1} d_2 >\}$. Since $s = (rt \bmod n)$, s is an element in $\{< \pm r_1 r_1^{-1} d_1, \pm r_2 r_2^{-1} d_2 >\} = \{< \pm d_1, \pm d_2 >\}$. It follows that s is one of the 4-th roots of y in Z_n^*. Hence we have that $s^4 \equiv y \pmod{n}$ and (9) is satisfied. □

Based on the proposed protocol of Section 3, an electronic cash system can be constructed through the methods introduced in [3, 4], where the signer of the blind signature protocol is regarded as the bank of the electronic cash system. An e-cash issued by the bank is of the form (s, m, c, a) which is produced by our partially blind signature protocol. Let (s, m, c, a) be an e-cash withdrawn by a payer from the bank by performing an electronic cash system based on the proposed protocol. To pay a payee the e-cash, the payer gives him (s, m, c, a). The payee verifies the correctness of the e-cash by checking if (9) is true, and then he immediately calls the bank to verify if the e-cash is fresh. An e-cash is fresh if and only if the e-cash has not been deposited into the bank, i.e., the e-cash has not been spent. If the e-cash is not double-spent, the payee accepts this payment, and deposits the e-cash into the bank. The bank then stores the e-cash in its database. In other words, the bank has to record all the used e-cash in its database to check whether a specified e-cash has been spent or not. Hence, the bank's database may grow unlimitedly. With the help of the partial blindness techniques, the size of the bank's database can be controlled. Let the predefined-format information a contain an expiration date of e-cash in an electronic cash system based on the proposed protocol. If an e-cash (s, m, c, a) is with an expired a, then it cannot be used in any transaction or payment. Hence, each e-cash (s, m, c, a) with an expired a recorded in the bank's database can be removed. Certainly, any fresh e-cash (s, m, c, a) with an unexpired a can be exchanged for another fresh e-cash (s', m', c', a') with a newer a' by performing another run of the proposed protocol.

4 Security

In this section we examine the security of the proposed partially blind signature scheme in Section 3, and discuss some key properties of this protocol.

4.1 Partial Blindness

The partial blindness property guarantees that all signatures issued by the signer contain a valid information a according to a predefined format negotiated and agreed by all users and the signer, and the users cannot remove or change the string a embedded in their signatures.

In the blinding stage of the proposed scheme, a user submits a and α to the signer to request a signature on m. After verifying that the string a is of the predefined format, the signer randomly selects x such that $(xH(a) \bmod n)$ is a QR in Z_n^*, and then derives t such that (7) holds. If the user tries to select α such that $(\alpha^2 H(a)x)^{-1} \equiv x^{-1} \pmod{n}$ and then remove or modify the string $H(a)$ in the signature, then he has to compute the square-root problem in Z_n^* to prepare a special value for α.

The scheme of [6] uses the power of 3 (or any odd number) in the computation formula (5) to derive t, so that the weakness, shown in Section 2, exists. Clearly, the above situation does not exist in the proposed scheme.

4.2 Randomization

In the proposed scheme, the signer perturbs the message received from a user before he signs on it by using a random integer x. This is the randomization property [7]. A randomized blind signature scheme can withstand the chosen-text attacks [5, 19]. Our scheme and the blind signature schemes of [2, 7, 14, 15] possess the randomization property, while the blind signature schemes of [1, 3] do not possess this property.

In the blinding stage of the proposed scheme, the user submits a and α to the signer to request a signature on m. The signer then randomly chooses x such that $(xH(a) \bmod n)$ is a QR in Z_n^*, and derives t such that (7) holds. If the user tries to select a special α such that

$(\alpha^2 H(a)x)^{-1} \equiv H(a)^{-1} \pmod{n}$ and then remove x from the signature, then he must cope with the square-root problem $\alpha = (x^{-\frac{1}{2}} \bmod n)$.

4.3 Unforgeability

According to the verification formula (9), if an attacker is about to select a 4-tuple (s, m, c, a) such that (9) is satisfied, he has to choose and decide the values of m, c, and a in advance because each of them is a parameter of H in (9). If he does not do so, it is intractable to derive (m, c, a) since H is one-way. Let the attacker decide the values of m, c, and a in advance. To solve s from (9), the attacker has to compute one of the 4-th roots of $((cH(a)H(c||m)^2)^{-1}$ mod $n)$, which is intractable when the factorization of n is unknown.

4.4 Unlinkability

For each instance numbered i of the proposed protocol, the signer can record α_i received from the user who communicated with the signer during the instance i of the protocol. The tuple (α_i, x_i) is usually referred to as the *view* of the signer to the instance i of the protocol. Thus, we have the following theorem.

Theorem 2 *Given a signature 4-tuple (s, m, c, a) produced by the protocol in Section 3, the signer can derive r and u for each view (α_i, x_i) such that (6) is satisfied where (α_i, x_i) is regarded as (α, x) in (6).*

Proof. Let $\left[\frac{h}{g}\right]$ denote the Legendre symbol h over g where g is a prime [20]. In the scheme, if an integer w with $\left[\frac{w}{p_1}\right] = \left[\frac{w}{p_2}\right] = 1$, then w is a QR in Z_n^* and it has four square roots $\{w_1, w_2, w_3, w_4\}$ in Z_n^* where $\left[\frac{w_1}{p_1}\right] = \left[\frac{w_2}{p_1}\right] = \left[\frac{w_1}{p_2}\right] = \left[\frac{w_3}{p_2}\right] = 1$ and $\left[\frac{w_3}{p_1}\right] = \left[\frac{w_4}{p_1}\right] = \left[\frac{w_2}{p_2}\right] = \left[\frac{w_4}{p_2}\right] = -1$.

In the instance i of the protocol, the signer chooses x_i such that $(x_i H(a) \bmod n)$ is a QR in Z_n^*. Let $\left[\frac{H(a)}{p_1}\right] = k_1$ and $\left[\frac{H(a)}{p_2}\right] = k_2$, where k_1, $k_2 \in \{1, -1\}$, then $\left[\frac{x_i}{p_1}\right] = \left[\frac{c}{p_1}\right] = k_1$ and $\left[\frac{x_i}{p_2}\right] = \left[\frac{c}{p_2}\right] = k_2$. Since $\left[\frac{cx_i^{-1}}{p_j}\right] = \left[\frac{c}{p_j}\right]\left[\frac{x_i^{-1}}{p_j}\right] = \left[\frac{c}{p_j}\right]\left[\frac{x_i}{p_j}\right]^{-1} = k_j k_j^{-1} = 1$ for $j = 1$ and 2, $(cx_i^{-1}$ mod $n)$ is a QR in Z_n^* and the signer can derive four different square roots $\{u_1, u_2, u_3, u_4\}$ in Z_n^* such that $c \equiv u_j^2 x_i \pmod{n}$ for $j = 1$, 2, 3, and 4. In addition, there must exist an integer $u_z \in \{u_1, u_2, u_3, u_4\}$ such that $\left[\frac{\alpha_i(u_z H(c||m))^{-1}}{p_1}\right] = \left[\frac{\alpha_i(u_z H(c||m))^{-1}}{p_2}\right] = 1$. Thus, the signer can derive four different square roots $\{r_1, r_2, r_3, r_4\}$ in Z_n^* such that $\alpha_i \equiv r_j^2 u_z H(c||m) \pmod{n}$ for $j = 1$, 2, 3, and 4. □

Therefore, given a signature 4-tuple (s, m, c, a) produced by the protocol, the signer can always derive r and u for each view (α_i, x_i) such that (6) is satisfied. It turns out that all of the 4-tuples (s, m, c, a)'s are indistinguishable from the signer's point of view under the same a. This is the unlinkability property under the same predefined information.

Table 2: Computations required for a user in the proposed scheme

	Blinding	Unblinding	Verifying	Signature size	Transmitted
Multiplication computation:	5	1	5	–	–
Hashing operation:	1	0	2	–	–
Random-number generation:	2	0	0	–	–
Number of messages:	–	–	–	4	4

5 Performance

In the proposed blind signature scheme, no modular exponentiation and inverse computations are performed by users. Moreover, only eleven modular multiplications, three hashing operations, and twice of random-number generation are performed by a user to obtain and verify a signature in the protocol. Comparing with [6], the proposed scheme reduces one random-number generation and nine modular multiplications in Z_n^* for the user.

In the proposed scheme, the signer performs a 4-th root computation, an inverse computation, and twice, on the average, of QR testing in Z_n^*. The computation required for the signer in [6] is almost the same as that in the proposed scheme.

There are five and four, respectively, messages transmitted over the channel between the signer and the user in [6] and the proposed protocol, respectively. Hence, the communication traffic is reduced by about 20% in our scheme. Besides, the size of a signature in the proposed scheme is equal to that in [6]. The computations required for a user are summarized in Table 2.

In addition, by Table 1 and Table 2, the proposed scheme is optimal in both the amount of transmitted messages and the size of a signature, and is almost optimal in arithmetical computations for users.

6 Conclusions

In the manuscript, a generic randomized partially blind signature scheme has been defined, and a new user efficient partially blind signature scheme has been proposed where the communication traffic and the computations for users are nearly optimal. It will make the applications based on partially blind signatures much more efficient in hardware-limited environments, such as smart cards and handsets. Besides, the weakness in the previous version of the scheme has also been repaired.

References

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

[2] J. L. Camenisch, J. M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," *Advances in Cryptology-EUROCRYPT'94*, LNCS 950, Springer-Verlag, pp. 428-432, 1995.

[3] D. Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology-CRYPTO'82*, Springer-Verlag, pp. 199-203, 1983.

[4] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," *Advances in Cryptology-CRYPTO'88*, LNCS 403, Springer-Verlag, pp.319-327, 1990.

[5] J. S. Coron, D. Naccache, and J. P. Stern, "On the security of RSA padding," *Advances in Cryptology-CRYPTO'99*, LNCS 1666, Springer-Verlag, pp. 1-18, 1999.

[6] C. I. Fan and C. L. Lei, "Low-computation partially blind signatures for electronic cash," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81-A, no. 5, pp. 818-824, 1998.

[7] N. Ferguson, "Single term off-line coins," *Advances in Cryptology-EUROCRYPT'93*, LNCS 765, Springer-Verlag, pp. 318-328, 1994.

[8] L. C. Guillou and J. J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," *Advances in Cryptology-EUROCRYPT'88*, LNCS 330, Springer-Verlag, pp. 123-128, 1988.

[9] C. L. Lei and C. I. Fan, "A universal single-authority election system," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81-A, no. 10, pp. 2186-2193, 1998.

[10] NIST FIPS PUB XX, Digital Signature Standard (DSS), National Institute of Standards and Technology, U.S. Department of Commerce, DRAFT, 1993.

[11] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the DSA giving message recovery schemes," *The first ACM Conference on Computer and Communications Security*, November 3-5, Fairfax, Virginia.

[12] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," *Advances in Cryptology-CRYPTO'92*, LNCS 740, Springer-Verlag, pp. 31-53, 1992.

[13] R. C. Peralta, "A simple and fast probabilistic algorithm for computing square roots modulo a prime number," *IEEE Transactions on Information Theory*, vol. 32, no. 6, pp. 846-847, 1986.

[14] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," *Advances in Cryptology-ASIACRYPT'96*, LNCS 1163, Springer-Verlag, pp. 252-265, 1996.

[15] D. Pointcheval and J. Stern, "New blind signatures equivalent to factorization," *Proceedings of the 4th ACM Conference on Computer and Communication Security*, pp. 92-99, 1997.

[16] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," *Technical Report*, MIT/LCS/TR212, MIT Lab., Computer Science, Cambridge, Mass. Jan. 1979.

[17] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[18] C. P. Schnorr, "Efficient identification and signatures for smart cards," *Advances in Cryptology-CRYPTO'89*, LNCS 435, Springer-Verlag, pp. 235-251, 1990.

[19] A. Shamir and C. P. Schnorr, "Cryptanalysis of certain variants of Rabin's signature scheme," *Information Processing Letters*, vol. 19, pp. 113-115, 1984.

[20] G. J. Simmons, *Contemporary Cryptology: The Science of Information Integrity*, IEEE Press, N.Y., 1992.

[21] H. C. Williams, "A modification of the RSA public-key encryption procedure," *IEEE Transactions on Information Theory*, vol. 26, no. 6, pp.726-729, 1980.