Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences xxx (2004) xxx–xxx
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Ownership-attached unblinding of blind signatures for untraceable electronic cash ☆

Chun-I Fan *

Department of Computer Science and Engineering, National Sun Yat-sen University, No. 70, Lien-Hai Road, Kaohsiung 804, Taiwan

Received 29 April 2004; received in revised form 12 October 2004; accepted 12 October 2004

Abstract

In an untraceable electronic cash protocol based on blind signatures, an identified customer can withdraw a blinded electronic cash from the bank and the unblinding operation is adopted by the customer to transform the blinded electronic cash into a valid one. Before performing the operation, the blinded electronic cash is protected well since attackers cannot convert it into a valid electronic cash without the blinding factor corresponding to the operation. However, after unblinding, the electronic cash will suffer from the theft attacks since it is not protected by any security mechanism. This paper introduces a new unblinding operation called *ownership-attached unblinding* which attaches the identities of a designated payee and a specified transaction to the blinded electronic cash and then produces an ownership-attached electronic cash other than a bare one such that the cash can withstand the theft attacks during the entire transaction because it is valid for the designated payee and the specified transaction only. Further-

☆ A partial result of this research was presented at the 14th International Conference on Information Networking, Hsinchu, Taiwan, January 26–28, 2000.

† This research was partially supported by the National Science Council of the ROC (Taiwan) under grant NSC 92-2213-E-110-035.

* Tel.: +886 7 5252000x4346; fax: +886 7 5254301.
E-mail address: cifan@cse.nsysu.edu.tw (Chun-I Fan).

doi:10.1016/j.ins.2004.10.010

more, the proposed scheme does not largely increase the computation cost required for each customer so that it also is a customer efficient protection solution for untraceable electronic cash and especially suitable for mobile clients and smart-card users.

Keywords: Electronic cash; Blind signatures; RSA; Cryptography; Security and privacy

1. Introduction

Due to the fast progress of computer technologies, the efficiency of data processing and the speed of information generation has been greatly improved. Moreover, the techniques of networks largely shorten the communicating time among distributed entities. Many advanced network services have been proposed in the literature to take the advantages of the techniques. Among these services, untraceable electronic cash (e-cash) is a popular one since it realizes the digitalization of traditional cash. Untraceable electronic cash makes it possible for customers to pay the e-cash to the merchants through communication networks under privacy protection [1–3,5,6,8,9,12,16,18]. As the unforgeability and untraceability of the e-cash can be guaranteed and the scenarios of the e-cash transactions are similar to those of the traditional-cash transactions, this kind of advanced digital money will be widely used.

Owing to the unforgeability and unlinkability of blind signatures, they are usually adopted to construct untraceable electronic cash protocols [4,5,8,12,19,20]. Basically, an untraceable electronic cash protocol contains initialization, withdrawing, unblinding, and paying stages. At the initialization stage, the bank publishes necessary information such as its public keys. To withdraw an e-cash from the bank, the customer requests a blinded e-cash from the bank at the withdrawing stage. At the unblinding stage, the customer transforms the blinded e-cash into a valid one through a blinding factor, where the transformation is usually referred to as the *unblinding* operation. Attackers cannot derive the valid e-cash from the blinded one without the blinding factor. Finally, the customer pays the e-cash to some payee for some transaction at the paying stage. The key point is that the bank cannot link the paying stage to the withdrawing stage, i.e., the bank cannot link an e-cash to the blinded form of the e-cash without the blinding factor, which is kept secret by the customer. This is the *untraceability* (or *unlinkability*) property [1–3,5,6,8,9,12,16,18].

We make a deep research on the unblinding operation in an untraceable electronic cash protocol based on blind signatures. We design a new type of unblinding operations called *ownership-attached unblinding* operations [10]. By attaching the ownership to an e-cash at the unblinding stage, the ownership-attached unblinding produces an electronic cash attached by the identities of a designated payee and a specified transaction. If it is duplicated by the

attackers or hackers, the duplicate will be invalid for any other payee and transaction. It turns out that the new type of unblinding transforms a blinded e-cash into a valid e-cash much more secure than a bare one produced by the typical unblinding operation, and the new protection mechanism can completely and efficiently protect the e-cash against the theft attacks during the entire payment process.

This paper focuses on the ownership-attached unblinding of blind signatures for untraceable electronic cash. To simplify the presentation, we adopt a basic electronic cash protocol to explain our idea where the protocol requires double-spending checking for each e-cash when paying and it possesses two basic properties, unforgeability and untraceability, of the e-cash. There are some other properties of electronic cash which have been discussed in the literature such as off-line double-spending checking for e-cash [6], divisible e-cash [16], e-cash tracing for the misuse of untraceability [14], and so on. Certainly, they also are interesting research topics to consider ownership-attached unblinding in these protocols. However, they are beyond the scope of the paper.

The rest of this paper is organized as follows. In Section 2, we review a blind signature scheme used in this paper. A basic untraceable electronic cash protocol based on blind signatures is shown in Section 3. In Section 4, we introduce two protection mechanisms for untraceable electronic cash. The proposed method is described in Section 5. The security and performance of the proposed scheme is examined in Section 6 and Section 7, respectively. Finally, a conclusion remark is given in Section 8.

2. Fan-Lei blind signature scheme

In order to guarantee the customer efficiency property for the environments where the computation capabilities of the customers are limited such as mobile clients and smart-card users, Fan-Lei user efficient blind signature scheme [8] is adopted to realize the proposed idea. Fan-Lei scheme is based on quadratic residues [21]. 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 $y^2 \equiv x \pmod{n}$ where Z_n^* is the set of all positive integers less than and relatively prime to n. Given x and n, it is intractable to compute y in Z_n^* if n contains large prime factors and the factorization of n is unknown [21].

2.1. The scheme

There are two kinds of participants, a signer and a group of users, in the blind signature scheme. A user requests signatures from the signer, and the signer computes and issues blind signatures to the users. The protocol has four phases: $\langle 0 \rangle$ initializing, $\langle 1 \rangle$ blinding, $\langle 2 \rangle$ signing, and $\langle 3 \rangle$ unblinding. The signer pub-

lishes the necessary information in the initializing phase. To obtain the signer's signature on a message, the user performs a blinding process with a blinding factor to transform the message into a blinded message, and then submits the blinded message to the signer in the blinding phase. The blinding process makes it information theoretically impossible for the signer to derive the message from the blinded message without the correct blinding factor. In the signing phase, the signer computes the signature on the blinded message, and then sends the signing result back to the user. Finally, the user performs the unblinding operation with the blinding factor to convert the signing result into the exact signature on the message in the unblinding phase. The details are described as follows.

Initially, the signer randomly selects two distinct large primes p and q such that $p \equiv q \equiv -1 \pmod 4$. The signer computes $n = pq$ and then publishes n. In addition, let H be a public one-way hash function.

⟨1⟩ *Blinding*: To request the signer's signature on message m, the user randomly chooses two integers u and v such that $\alpha = (H(m)(u^2 + v^2) \bmod n)$ is in Z_n^* and then submits the integer α to the signer.
After receiving α, the signer randomly selects x such that $(\alpha(x^2 + 1) \bmod n)$ is a QR in Z_n^*, and then sends the integer x to the user.
The user chooses an integer b at random in Z_n^*, and then computes $\delta = (b^2 \bmod n)$ and $\beta = (\delta(ux + v) \bmod n)$. The user transmits 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

$$t^4 \equiv \alpha(x^2 + 1)\lambda^2 \pmod n$$

through the algorithm of [21]. Hence t is one of the 4th roots of $(\alpha(x^2 + 1)\lambda^2 \bmod n)$ in Z_n^*. The signer sends t and λ to the user.

⟨3⟩ *Unblinding*: After receiving t and λ, the user computes

$$\begin{cases} c = \delta\lambda(u - vx) \bmod n \text{ and} \\ s = bt \bmod n. \end{cases}$$

The integer s is the signer's signature on (c, m) where c is called the *randomization* parameter. To verify (c, m, s), one can examine if

$$s^4 \equiv H(m)(c^2 + 1) \pmod n. \tag{1}$$

2.2. The underlying signature foundation

The security of Rabin's signature scheme [21] had been proven to be computationally equivalent to the factoring problem. Hence, if factoring n is computationally intractable where n is the product of two large random distinct

primes with roughly the same size, then Rabin's scheme is provably secure against a passive adversary. However, Rabin's scheme succumbs to the chosen-message attacks [7,15].

Fan-Lei blind signature scheme is based on Rabin's signatures with injecting randomizing factors x's into the messages before the signer performs the signing operations on them. The scheme is robust against a passive adversary due to using the Rabin's method, and the randomizing mechanism enhances the randomization of Rabin's signatures such that it is computationally infeasible for an adversary to predict the contents of the messages the signer exactly signs in the chosen-message attacks such as [7,13].

In the blind signature scheme, the signer perturbs the message received from the user before signing it by using the randomly chosen integer x. This is said to be the *randomization* property [12]. A randomized blind signature scheme can withstand the chosen-message attacks [24]. Fan-Lei scheme and the schemes of [4,8,12,19,20] possess the randomization property, while Chaum's blind signature scheme of [5] does not satisfy this property. In 1999, Coron, Naccache, and Stern also presented a signature forgery strategy of the RSA digital signature scheme [7], and the attack is valid on some special cases of Chaum's blind signature scheme [5]. However, if these two schemes can be randomized, then the attack will be invalid on them.

2.3. The unlinkability property

In Fan-Lei blind signature scheme [8], the signer can keep a set of records $\{(\alpha_j, \beta_j, x_j, t_j)|$ for each instance j of the protocol$\}$, where

$$\begin{cases} \alpha_j \equiv H(m_j)(u_j^2 + v_j^2) \pmod{n} \text{ and} \\ \beta_j \equiv b_j^2(u_j x_j + v_j) \pmod{n}. \end{cases}$$

The 4-tuple $(\alpha_j, \beta_j, x_j, t_j)$ is usually referred to as the *view* of the signer to the instance j of the protocol. By [11], given a triple (c, m, s) produced by Fan-Lei blind signature scheme, the signer can compute b'_j, u'_j, and v'_j for each $(\alpha_j, \beta_j, x_j, t_j)$ in polynomial time from the three congruences

$$\begin{cases} b'_j \equiv s t_j^{-1} \pmod{n}, \\ u'_j x_j + v'_j \equiv \beta_j (b'_j)^{-2} \pmod{n}, \text{ and} \\ (u'_j - v'_j x_j) \equiv c(u'_j x_j + v'_j) \pmod{n} \end{cases}$$

such that

$$\alpha_j \equiv H(m)((u'_j)^2 + (v'_j)^2) \pmod{n}.$$

Thus, given (c, m, s), the signer can derive (b'_j, u'_j, v'_j) for each stored record $(\alpha_j, \beta_j, x_j, t_j)$ and the checking formula

$$\alpha_j \equiv H(m)((u'_j)^2 + (v'_j)^2) \pmod{n}$$

is always satisfied. Hence, it is information theoretically impossible for the signer to derive the link between (c,m,s) and its corresponding view. This is the unlinkability (or blindness) property.

The author of [23] claimed that Fan-Lei blind signature scheme is not really blind. Nevertheless, in [11], Fan and Lei had shown that his claim is not true and Fan-Lei scheme satisfies the blindness property.

3. An untraceable electronic cash protocol based on Fan-Lei blind signatures

An untraceable electronic cash has to be unforgeable and untraceable (or unlinkable) [1–3,5,6,8,9,12,16,18]. To achieve these two purposes, an electronic cash protocol is usually based on the techniques of blind signatures [5,8,9]. We introduce a basic untraceable electronic cash protocol based on Fan-Lei blind signature scheme [8].

3.1. The protocol

Initially, the bank randomly selects two distinct large primes p and q with $p \equiv q \equiv -1 \pmod 8$. The bank derives p' and q' such that $p'p \equiv 1 \pmod q$ and $q'q \equiv 1 \pmod p$, and then keeps (p,q,p',q') secret. It computes $n = pq$ and publishes n. Let H be a public one-way hash function and each e-cash issued by the bank be worth w dollars.

3.1.1. Withdrawing

If a customer decides to withdraw an e-cash from her/his account in the bank, she/he performs the following protocol with the bank.

⟨1⟩ *Blinding*: The customer randomly chooses three integers m, u, and v such that the integer $\alpha = (H(m)(u^2 + v^2) \bmod n)$ is in Z_n^*. She/He submits the integer α to the bank.

⟨2⟩ *Randomizing*: After verifying the identity of the customer through a secure identification protocol [15,17], the bank randomly selects x such that $(\alpha(x^2 + 1) \bmod n)$ is a QR in Z_n^*. The bank sends the integer x to the customer.

⟨3⟩ *Responsing*: After receiving x, the customer randomly selects an integer b in Z_n^*, and then computes $\delta = (b^2 \bmod n)$ and $\beta = (\delta(ux + v) \bmod n)$. She/He transmits β to the bank.

⟨4⟩ *Signing*: After receiving β, the bank computes $\lambda = (\beta^{-1} \bmod n)$ and derives an integer t in Z_n^* such that

$$t^4 \equiv \alpha(x^2 + 1)\lambda^2 \pmod{n}. \tag{2}$$

The bank sends the tuple (t, λ) to the customer and deducts w dollars from the customer's account.

3.1.2. Unblinding

After receiving (t, λ), the customer can verify if $t^4 \equiv \alpha(x^2 + 1)\lambda^2 \pmod{n}$ and then computes

$$\begin{cases} c = \delta\lambda(u - vx) \bmod n \text{ and} \\ s = bt \bmod n. \end{cases} \tag{3}$$

The triple (c, m, s) is an e-cash in the protocol and it is said to be a *bare* e-cash because the e-cash is not protected by any security mechanism for withstanding the theft attacks. The bare e-cash (c, m, s) is valid for any payee and transaction once it is thieved by the attackers.

3.1.3. Paying

If the customer decides to pay the e-cash (c, m, s) to the merchant with identity e and for the transaction with identity r, they perform the protocol below.

⟨1⟩ *Transferring*: The customer sends $\{(c, m, s), r\}$ to merchant e.
⟨2⟩ *Verifying*: After receiving $\{(c, m, s), r\}$, the merchant examines the correctness of the e-cash (c, m, s) by verifying whether $s^4 \equiv H(m)(c^2 + 1) \pmod{n}$ is true or not. If it is false, this protocol terminates and the payment fails.
⟨3⟩ *Forwarding*: The merchant forwards the verified e-cash (c, m, s) and e to the bank.
⟨4⟩ *Double-spending checking*: After receiving $\{(c, m, s), e\}$ and examining the correctness of the e-cash by verifying if $s^4 \equiv H(m)(c^2 + 1) \pmod{n}$, the bank searches its database which stores all spent e-cash to check whether the e-cash is fresh (i.e., not double-spent) or not. If the e-cash is not fresh, the bank informs the merchant not to accept this payment and this protocol terminates. On the contrary, if the e-cash is fresh, the bank informs the merchant to accept this payment and records (c, m, s) into its database for future double-spending checking, and then adds w dollars to the account of merchant e.

3.2. Discussions

There are two key features of the above electronic cash protocol.

⟨1⟩ *Unforgeability*: The unforgeability of the e-cash relies on the security of Fan-Lei blind signatures.

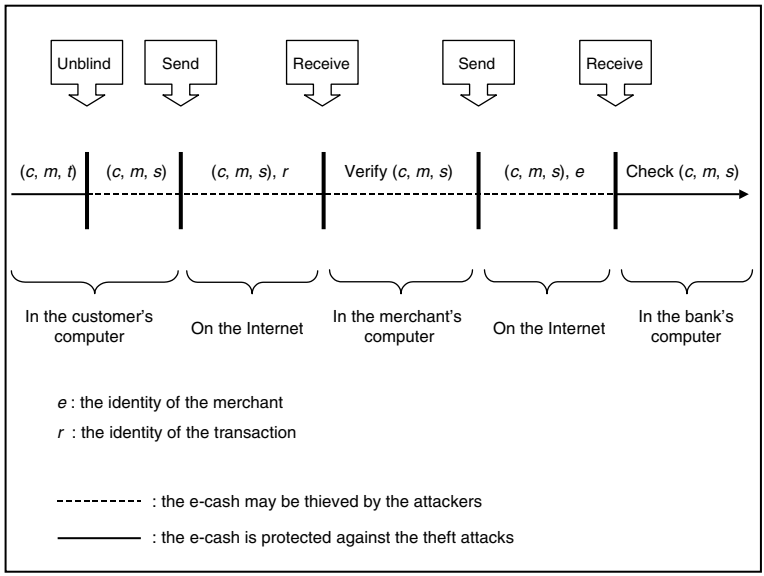

Fig. 1. The theft attacks on a bare e-cash.

⟨2⟩ *Untraceability*: Since the blinding factors b, u, and v are randomly chosen and kept secret by the customer, by [8,11], it is information theoretically impossible for the bank to derive the link between the e-cash (c, m, s) and the instance of the withdrawing protocol which produced the blinded version t of the signature s. This is the untraceability (or unlinkability) property.

Before unblinding, the e-cash is secure against the theft attacks since the attackers cannot derive s from t without the blinding factor b. Note that the e-cash (c, m, s) may be thieved after unblinding because it has not been protected. However, once the e-cash is received and stored by the bank, the theft attacks are invalid due to the double-spending checking. It is illustrated in Fig. 1.

4. Two protection schemes for untraceable electronic cash

In this section, two protection mechanisms for untraceable electronic cash are introduced.

4.1. An encryption-based protection scheme

In order to protect the e-cash during the transaction, a straight-forward solution is to encrypt the e-cash such that only the legitimate payee can decrypt the

encrypted e-cash. To reduce the overhead of key distribution, we adopt a public-key cryptosystem to implement the encryption and decryption functions. Consider the protocol in Section 3. Let E_{merchant} be the public-key encryption function of the merchant and E_{bank} be the public-key encryption function of the bank. At the transferring stage of the protocol in Section 3.1.3, the customer computes $E_{\text{merchant}}(r,s)$ and sends $\{E_{\text{merchant}}(r,s),c,m\}$ to the merchant. The merchant decrypts $E_{\text{merchant}}(r,s)$ to obtain (r,s) and verifies (c,m,s) at the verifying stage. The merchant then computes $E_{\text{bank}}(e,s)$ and sends $\{E_{\text{bank}}(e,s),c,m\}$ to the bank at the forwarding stage. Finally, the bank decrypts $E_{\text{bank}}(e,s)$ and verifies whether the e-cash is correct and fresh or not. If true, the bank deposits the e-cash into the account of the merchant with identity e.

Such a protection mechanism can protect the e-cash against the theft attacks after it is encrypted and before it is decrypted. However, the e-cash may still be thieved by the attackers before it is encrypted in the customer's and merchant's computers, respectively. It is shown in Fig. 2.

Furthermore, the encryption-based scheme has another weakness as compared with the signature-based scheme in Section 4.2 and the proposed scheme in Section 5. It is described below. In the encryption-based scheme, after receiving $\{E_{\text{merchant}}(r,s),c,m\}$, merchant e obtains the e-cash (c,m,s) and then the merchant or some agent of the merchant sends $\{E_{\text{bank}}(e',s),c,m\}$ to the bank to deposit the e-cash into the account with identity e' where $e \neq e'$. Later,

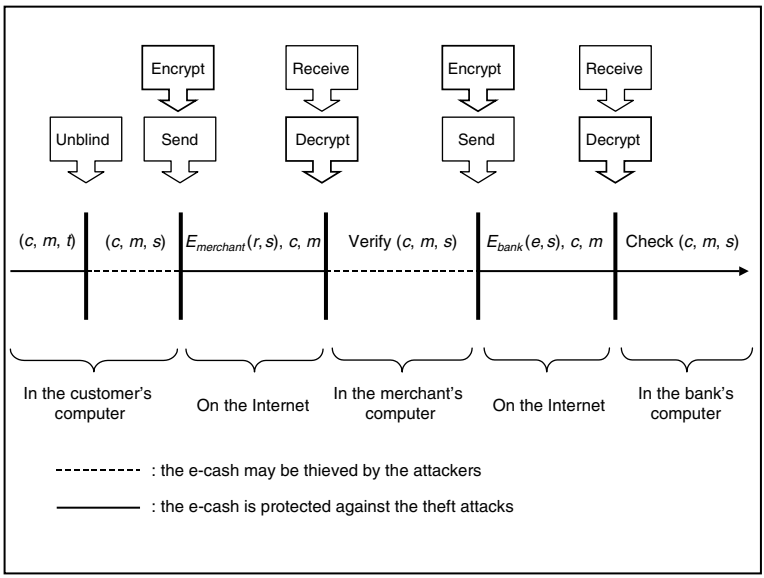

Fig. 2. The theft attacks on an encrypted e-cash.

the merchant sends $\{E_{\text{bank}}(e,s),c,m\}$ to the bank to deposit the e-cash again. Certainly, the merchant will receive a failure notification from the bank since the e-cash (c,m,s) is double-spent. Thus, the merchant can illegally claim that the payment is unsuccessful by presenting the failure notification to the customer. However, the merchant had indeed deposited the e-cash the customer paid into the bank.

4.2. A signature-based protection scheme

Another solution is to sign the e-cash along with the identities of the merchant and the transaction by using the digital signature scheme embedded in the e-cash. Let m contain the verification function V_{cash} of the digital signature scheme selected by the customer where the signing function S_{cash} corresponding to V_{cash} is kept secret by the customer. For the unlinkability property, the customer must select different $\{S_{\text{cash}}, V_{\text{cash}}\}$ for different e-cash. At the transferring stage of the protocol in Section 3.1.3, the customer computes $S_{\text{cash}}(H(r),e,s)$ and then sends $\{S_{\text{cash}}(H(r),e,s),c,m,r,s\}$ to merchant e. The merchant examines if $S_{\text{cash}}(H(r),e,s)$ is a valid signature via V_{cash} extracted from m, and then verifies if the e-cash (c,m,s) is correct at the verifying stage. The merchant forwards $\{S_{\text{cash}}(H(r),e,s),c,e,m,H(r),s\}$ to the bank at the forwarding stage. Finally, the bank checks if the signature $S_{\text{cash}}(H(r),e,s)$ is valid

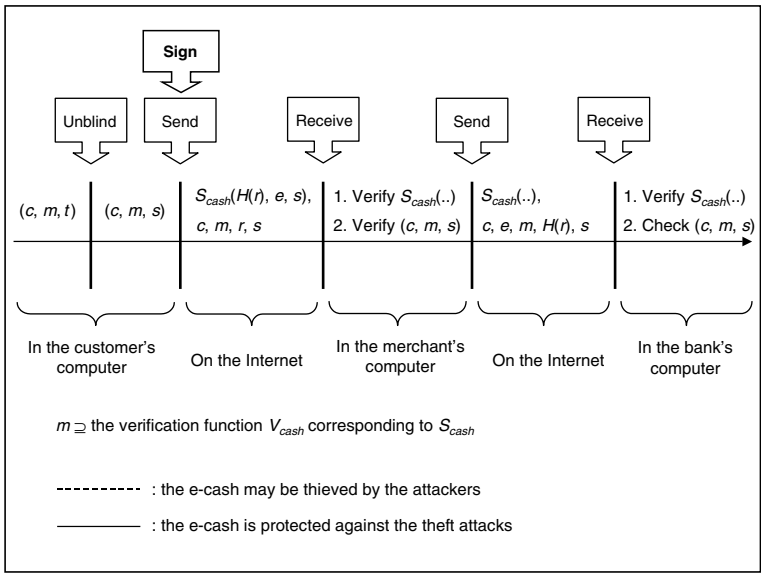

Fig. 3. The theft attacks on a signed e-cash.

by V_{cash} and verifies whether the e-cash is correct and fresh or not. If true, the bank deposits the e-cash into the account of the merchant with identity e.

The protection mechanism can withstand the theft attacks since $\{S_{\text{cash}}(H(r),e,s),c,m,r,s\}$ (or $\{S_{\text{cash}}(H(r),e,s),c,e,m,H(r),s\}$) is valid for merchant e and transaction r only. It is shown in Fig. 3.

If the customer produces $\{S_{\text{cash}}(H(r),e,s),c,e,m,r,s\}$ and $\{S_{\text{cash}}(H(r'),e',s),c,e',m,r',s\}$ with $(e,r) \neq (e',r')$, then only the first used one is valid and the later used one is regarded as a doubly spent e-cash by the bank due to the same (c,m).

If we adopt a practical digital signature scheme, such as the RSA cryptosystem [22], to implement $\{S_{\text{cash}}, V_{\text{cash}}\}$, then the customer must perform the computations of large prime generations, inverse, and modular exponentiation computations for protecting each of her/his e-cash. These computations are time-consuming as compared with modular multiplications, hashing computations, or random-number generations [25], so the signature-based protection scheme is not customer efficient.

5. Ownership-attached unblinding for untraceable electronic cash

Based on Fan-Lei blind signature scheme [8], we propose an electronic cash protocol with ownership-attached unblinding. The key points of our idea are:

⟨1⟩ delaying the unblinding operation until the identities of the merchant and the transaction are ascertained;
⟨2⟩ attaching the identities of the merchant and the transaction to the blinded e-cash after they are ascertained; and
⟨3⟩ performing the unblinding operation after the attachment is finished.

The idea also is illustrated in Fig. 4. Our method can efficiently produce a robust electronic cash against the theft attacks during the entire transaction owing to the ownership-attached unblinding.

The proposed protocol is described below.

5.1. The proposed protocol

Initially, the bank randomly selects two distinct large primes p and q such that $p \equiv q \equiv -1 \pmod{32}$, and then publishes n where $n = pq$. The bank computes p' and q' such that $p'p \equiv 1 \pmod{q}$ and $q'q \equiv 1 \pmod{p}$ and keeps (p,q,p',q') secret. Let each e-cash issued by the bank be worth w dollars, and let F and H be two public one-way hash functions. In addition, the bank issues a secure tamper-resistant hardware device (or module) to each authorized merchant. The device contains $\{n,p,q,p',q',F\}$ and the identity of the

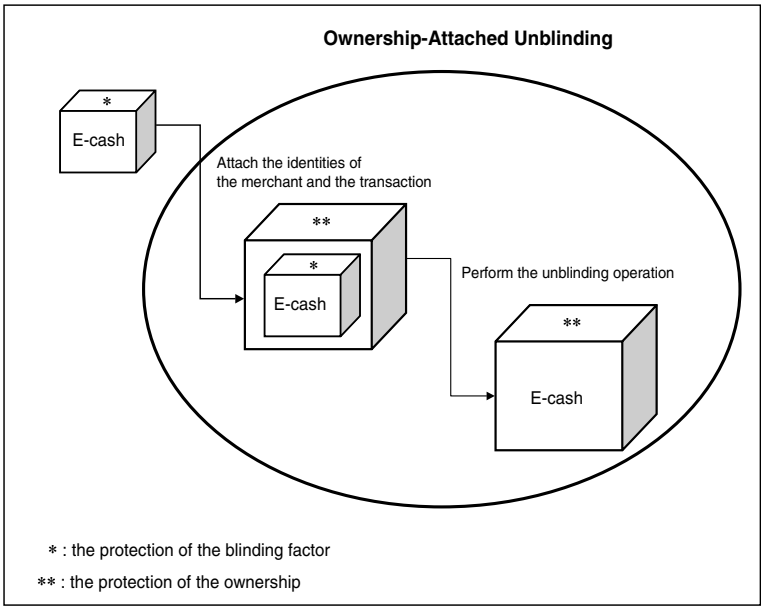

Fig. 4. The idea of ownership-attached unblinding.

merchant where it is impossible to thieve or modify the information and programs embedded in the device. For an input (k, r, t'), the device produces an output (t'', y) satisfying

$$(t'')^{16} \equiv \left(F(e\|k\|F(r\|y))(t')^4\right)^{-1} (\mathrm{mod}\, n) \tag{4}$$

where e being the merchant's identity embedded in the device, $\|$ being the string concatenation operator, and y being randomly chosen by the device such that $F(e\|k\|F(r\|y))$ is a QR in Z_n^*.

5.1.1. Withdrawing

If a customer is about to withdraw an e-cash from her/his account in the bank, she/he performs the following protocol with the bank.

⟨1⟩ *Blinding*: The customer randomly chooses three integers m, u, and v such that the integer $\alpha = (H(m)(u^2 + v^2) \bmod n)$ is in Z_n^*. The customer submits the integer α to the bank.

⟨2⟩ *Randomizing*: After verifying the identity of the customer through a secure identification protocol, the bank randomly selects x such that $(\alpha(x^2 + 1) \bmod n)$ is a QR in Z_n^*. The bank sends the integer x to the customer.

⟨3⟩ *Responsing*: After receiving x, the customer randomly selects an integer b in Z_n^*, and then computes $\delta = (b^8 \bmod n)$ and $\beta = (\delta(ux + v) \bmod n)$. The customer transmits β to the bank.

⟨4⟩ *Signing*: After receiving β, the bank computes $\lambda = (\beta^{-1} \bmod n)$ and derives an integer t in Z_n^* such that

$$t^8 \equiv \left(\alpha(x^2+1)\lambda^2\right)^{-1} \pmod{n} \tag{5}$$

since it has p and q [21,25]. The bank sends the tuple (t, λ) to the customer and deducts w dollars from the customer's account. It is impossible for the attackers to unblind t to form a valid e-cash without the blinding factor b which is kept secret by the customer. Besides, the customer can verify if $(t^8\alpha(x^2+1)\lambda^2) \equiv 1 \pmod{n}$.

5.1.2. Paying

Since the blinding factor b can protect the e-cash against the theft attacks, the customer does not need to perform the unblinding operation until the customer has decided the merchant which she/he wants to pay.

⟨1⟩ *Ownership-attached unblinding*: If the customer decides to pay the e-cash to the merchant with identity e for the transaction with identity r, she/he randomly selects an integer b' in Z_n^* and computes

$$t' = (b')^4 t^2 \bmod n. \tag{6}$$

She/He then derives $c = (\delta\lambda(u - vx) \bmod n)$ and $k = F(c\|m)$. The customer sends a payment request with (k, r, t') to merchant e. The merchant inputs (k, r, t') to the device and then the device produces an output (t'', y) such that (4) is true. The merchant sends (t'', y) to the customer. The customer can verify if $(t'')^{16}(t')^4 F(e\|k\|F(r\|y)) \equiv 1 \pmod{n}$. She/He performs the unblinding operation by computing

$$s = bb't'' \bmod n. \tag{7}$$

This kind of unblinding is called the ownership-attached unblinding. The triple (c, m, s) is an e-cash valid only for merchant e and transaction r, and it is said to be an *ownership-attached electronic cash*. Note that if $c_1 \equiv c_2 \pmod{n}$, $m_1 = m_2$, and $s_1 \equiv s_2 \pmod{n}$, then $(c_1, m_1) = (c_2, m_2)$ and $(c_1, m_1, s_1) = (c_2, m_2, s_2)$ in the scheme.

⟨2⟩ *Transferring*: The customer sends $\{(c, m, s), r, y\}$ to merchant e.

⟨3⟩ *Verifying*: After receiving $\{(c, m, s), r, y\}$, the merchant verifies it by checking if

$$s^{16} F(e\|F(c\|m)\|F(r\|y)) \equiv H(m)(c^2+1) \pmod{n}. \tag{8}$$

The above verification process is called the *ownership-attached verification*.

⟨4⟩ *Forwarding*: The merchant forms $d = F(r\|y)$ and sends $\{(c,m,s),d,e\}$ to the bank for the double-spending checking.

⟨5⟩ *Double-spending checking*: After receiving $\{(c,m,s),d,e\}$, the bank examines whether the formula

$$s^{16}F(e\|F(c\|m)\|d) \equiv H(m)(c^2+1) \pmod{n} \tag{9}$$

is true or not. If true, the e-cash is correct. Then the bank searches its database to check whether (c,m) is distinct with each of the pairs stored in its database or not. If true, the e-cash is fresh (not double-spent). Once the e-cash is correct and fresh, the bank informs the merchant to accept this payment, and then it stores the e-cash in its database and adds w dollars to the account of the merchant with identity e.

In the proposed protocol, after performing the ownership-attached unblinding, the merchant's identity e and the transaction identity r have been embedded into the e-cash, so that it is invalid for any payee other than e and invalid for any transaction other than r even though the e-cash has been thieved by the attackers. Hence, the protection mechanism can completely protect the e-cash against the theft attacks during the transaction. It is also shown in Fig. 5.

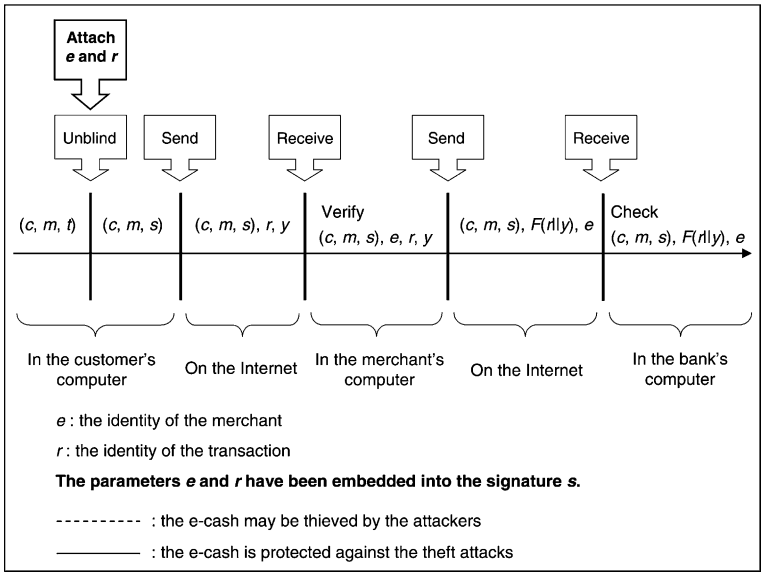

Fig. 5. The theft attacks on an ownership-attached e-cash.

6. Discussions

In this section, we discuss the correctness and security of the protocol proposed in Section 5.

6.1. Correctness

Lemma 1 guarantees the correctness of the proposed protocol in Section 5.

Lemma 1. *If* $\{(c,m,s),e,r,y\}$ *is produced by the protocol of Section* 5, (8) *will be true.*

Proof. By (7), (4), (6), and (5), s^{16}

$$\begin{aligned}
&\equiv (bb't'')^{16} \equiv (bb')^{16}(t')^{-4}F(e\|k\|F(r\|y))^{-1}\\
&\equiv (bb')^{16}((b')^4t^2)^{-4}F(e\|k\|F(r\|y))^{-1}\\
&\equiv b^{16}\alpha(x^2+1)\lambda^2F(e\|k\|F(r\|y))^{-1}\\
&\equiv b^{16}H(m)(u^2+v^2)(x^2+1)b^{-16}(ux+v)^{-2}F(e\|k\|F(r\|y))^{-1}\\
&\equiv H(m)(1+(ux+v)^{-2}(u-vx)^2)F(e\|k\|F(r\|y))^{-1}\\
&\equiv H(m)(1+c^2)F(e\|k\|F(r\|y))^{-1}\\
&\equiv F(e\|F(c\|m)\|F(r\|y))^{-1}H(m)(c^2+1) \pmod{n}.
\end{aligned}$$

Thus, we have that $s^{16}F(e\|F(c\|m)\|F(r\|y)) \equiv H(m)(c^2+1) \pmod{n}$. □

6.2. Unforgeability

The unforgeability of the e-cash in the proposed scheme relies on the unforgeability of the signatures in Fan-Lei scheme [8] and Rabin's scheme [21] with randomization factors.

If the attackers attempt to select a set $\{(c,m,s),e,r,y\}$ such that (8) is true, they must decide the values of $\{c,m,e,r,y\}$ in advance. If they do not do so, the forgery will fail since F and H are one-way. Let the attackers choose the values of $\{c,m,e,r,y\}$ in advance. It is still computationally infeasible for the attackers to derive s by (8) without the factorization of n because the integer s is one of the 16th roots of $(F(e\|F(c\|m)\|F(r\|y))^{-1}H(m)(c^2+1) \bmod n)$ in Z_n^* [21].

Assume that the attackers have the device and attempt to forge a set $\{(c,m,s),e,r,y\}$ satisfying (8) with the help of the device. The attackers must prepare the triple $(k = F(c\|m),r,t')$ as the input of the device where $(t')^{-4} \equiv H(m)(c^2+1) \pmod{n}$, and then obtain the output (t'',y) and let

$s = t''$. However, they cannot compute t' except that they can forge the signer's signature $((t')^{-1} \bmod n)$ on (c,m) in Fan-Lei blind signature scheme [8] of Section 2.1.

Besides, if the customer forms $\{(c,m,s),e,r,y\}$ and $\{(c,m,s'),e',r',y'\}$, respectively, with the help of merchant e and merchant e', respectively, the first used one will be valid and the later used one will be invalid since it cannot pass the double-spending checking.

The scheme is designed to protect the e-cash against illegally parties or attackers. Therefore, we assume that the bank does not illegally remove or modify $\{e,r\}$ embedded in the e-cash during the transaction even though the bank has the factorization of n.

6.3. Untraceability

Compared with the electronic cash protocol of Section 3, the extra information attached to the e-cash is $\{e,F(r\|y)\}$ in the protocol of Section 5 from the bank's point of view. Clearly, the identity e of the merchant is known to the bank after the e-cash is deposited into the merchant's account even if e has not been attached to the e-cash. The transaction identity r is selected or randomly selected by the merchant and/or the customer to identify the transaction, and it is meaningful to the merchant and/or the customer only. Therefore, the attachment does not affect the untraceability or unlinkability of the e-cash.

Let the device record $\{e,k,r,y,t'\}$ and send the set to the bank. Due to the unlinkability of Fan-Lei scheme and $t' = ((b')^4 t^2 \bmod n)$ where b' is unknown to the bank, the bank cannot link $\{(c,m,s),e,k,r,y,t'\}$ to t or the instance of the withdrawing protocol which produced t. The untraceability property is still guaranteed.

6.4. Robustness

In the protocol of Section 5, the blinding factor b is kept secret by the customer for preserving the unlinkability of the e-cash. Before performing the unblinding operation, the blinding factor can also protect the e-cash against the theft attacks. After unblinding, the protocol produces an ownership-attached electronic cash for merchant e and transaction r. If the e-cash is duplicated, the duplicate cannot be paid to any payee other than e for any transaction other than r.

7. Performance analysis

In this section, we will evaluate the performance of the proposed scheme and make comparisons with the others.

7.1. The computation cost

Typically, under the modulus n, the computation cost of a modular exponentiation computation is about $O(|n|)$ times that of a modular multiplication where $|n|$ denotes the bit length of n [25]. The modulus n is usually taken at least 1024 bits in practical implementation [15,25]. Besides, an inverse computation in Z_n^* takes about the same time as that of a modular exponentiation computation in Z_n^*. The 4th-root, 8th-root, or 16th-root computation in Z_n^* requires the computation cost not less than that of a modular exponentiation computation in Z_n^* [21,25]. A random-number generation or a hashing computation does not take longer time than that of a modular multiplication computation in Z_n^* [25]. Hence, the computation time consumed by the modular multiplications, hashing operations, and random-number generations in all of the presented protocols can be neglected as compared with the modular exponentiation, inverse, 4th-root, 8th-root, or 16th-root computations under a modulus with 1024 or more bits in these protocols.

In order to compare the proposed protection scheme with the others on performance, we adopt a popular and widely used cryptosystem, the RSA cryptosystem [22], with a modulus whose length is roughly equal to $|n|$ bits to implement the encryption and decryption functions in the scheme of Section 4.1 and the signing and signature verification functions in the scheme of Section 4.2. In the RSA cryptosystem, we need to perform one modular exponentiation computation for each of the encryption, decryption, signing, and signature verification operations [22]. Let the computation cost of a modular exponentiation computation in Z_n^* be T.

7.1.1. The computation cost of the encryption-based protection scheme

Considering the protocol in Section 4.1, the additional computations are two encryption and two decryption operations as compared with the bare electronic cash protocol in Section 3. In the RSA cryptosystem, it requires to perform four modular exponentiation computations for the two encryption and two decryption operations. Therefore, the computation cost of the encryption-based protection scheme is about $4T$.

7.1.2. The computation cost of the signature-based protection scheme

In the protocol of Section 4.2, the parameters of the RSA cryptosystem, including two distinct large primes and the public and private keys, must be generated by the customer for each e-cash as compared with the protocol in Section 3. Twice of prime generation consume the computation time much more than performing two modular exponentiation computations [15,25] and it requires an inverse computation to generate the public and private keys in the RSA cryptosystem. In addition to prime generation and inverse computation, one signing and two signature verification operations are performed in the

protocol for each e-cash as compared with the protocol in Section 3. Hence, the computation cost of the signature-based protection scheme is much higher than $6T$.

7.1.3. The computation cost of the proposed protection scheme

Consider the protocol in Section 5. We have designed an efficient algorithm, i.e., Algorithm 1, in the Appendix A to compute one of the 8th or 16th roots of the inverse of a QR in Z_n^* and we have also shown that the computation cost of Algorithm 1 is about T. To compute t such that (5) is true, we can perform once of Algorithm 1 with input $(p,q,p',q',n,\theta,\sigma)$ where $\theta = 8$ and $\sigma = \alpha(x^2+1)\lambda^2 \bmod n$. To compute t such that (2) holds, we can perform once of a modified version of Algorithm 1 with modified Step 2: "compute $\varphi_1 = \sigma^{\frac{p+1}{2\theta}} \bmod p$" and modified Step 3: "compute $\varphi_2 = \sigma^{\frac{q+1}{2\theta}} \bmod q$" where input $= (p,q,p',q',n,4,\alpha(x^2+1)\lambda^2 \bmod n)$. Hence, computing t in the protocol of Section 5 takes the same time as computing t in the protocol of Section 3. Thus, the additional computation in the protocol of Section 5 is the computation of t'' as compared with the bare electronic cash protocol in Section 3.

If y is chosen such that $F(e\|k\|F(r\|y))$ is a QR in Z_n^*, the device can perform once of Algorithm 1 with input $(p,q,p',q',n,\theta,\sigma)$ where $\theta = 16$ and $\sigma = F(e\|k\|F(r\|y))(t')^4 \bmod n$, and then obtain the output φ and let $t'' = \varphi$ where (4) is satisfied. Since $p \equiv q \equiv -1 \pmod 4$, given a randomly chosen string y, the probability of $F(e\|k\|F(r\|y))$ being a QR in Z_n^* is about 1/4 [15,21,25]. Hence, the device must repeatedly randomly choose y and perform Algorithm 1 at least one time and at most four times to obtain correct φ such that $\varphi^{16}\sigma \equiv 1 \pmod n$. It follows that the computation cost of the proposed protection scheme is between T and $4T$.

Furthermore, only some additional modular multiplications, hashing computations, and random-number generations are performed by the customer in the proposed scheme, so the computation cost for the customer is much lower than T and the customer efficiency property is preserved. However, the computation cost for the customer is much greater than $4T$ in the signature-based protection scheme such that it is quite inefficient for the customer.

7.2. The communication cost

Since the communication traffic of the protocol in Section 4.1 is roughly equal to that of the bare electronic cash protocol in Section 3, the communication cost of the encryption-based protection scheme is about 0.

The extra communication traffic of the protocol in Section 4.2 is the transmission of $H(r)$ and twice of the transmission of $\{S_{\text{cash}}(H(r),e,s)$, V_{cash} embedded in $m\}$ where V_{cash} contains the signature verification key and the underlying modulus. Compared with the bare electronic cash protocol in Sec-

Table 1
The comparisons of the three protection schemes

	Encryption based	Signature based	Ownership-attached unblinding
Complete protection:	No	Yes	Yes
The computation cost:	$4T$	$>>6T$	$T \sim 4T$
The communication cost:	0	$>4\|n\|$	$2.5\|n\| \sim 3\|n\|$

tion 3, the additional communication cost of the protocol in Section 4.2 is $|H(r)| + 2|S_{\text{cash}}(H(r), e, s)| + 2|V_{\text{cash}}| > |H(r)| + 2|n| + 2|n| > 4|n|$ bits.

Compared with the protocol in Section 3, the additional messages transmitted in the protocol of Section 5 are d, k, r, t', t'', and two y's where r being the identity of the transaction, y being a randomly chosen string, $|t'| = |t''| = |n|$, and $|d| = |k| = 160$ bits when we adopt the popular hash function SHA-1 [15,25] to implement F in the protocol. In practical implementation, y can be produced by performing a one-way hash function. Let y be generated by SHA-1. The additional communication traffic of the protocol in Section 5 is $(|d| + |k| + |r| + |t'| + |t''| + 2|y|) = (160 + 160 + |r| + |n| + |n| + 320) = (2|n| + |r| + 640)$ bits as compared with the protocol in Section 3. Since we choose the modulus n with 1024 or more bits and r is usually short, the communication cost of the proposed protection scheme is between $2.5|n|$ bits and $3|n|$ bits, i.e., $2.5|n| \leq (2|n| + |r| + 640) \leq 3|n|$.

The comparisons of the two protection schemes shown in Section 4 and the proposed scheme described in Section 5 are summarized in Table 1.

Although the communication cost of the encryption-based protection scheme is lower than that of the proposed scheme, the encryption-based scheme cannot completely protect the e-cash in the transaction including that the merchant may cheat the customer of her/his e-cash in the encryption-based scheme, just as the case shown in Section 4.1. The signature-based scheme can provide a complete protection solution, but the computation cost of the scheme is quite high, even for the customer. Not only does the proposed scheme completely protect the e-cash during the entire transaction but it is customer efficient as well.

8. Conclusions

We have proposed a new unblinding operation, ownership-attached unblinding, to produce a robust electronic cash against the theft attacks during the entire transaction process. Compared with other protection mechanisms, although the security hardware devices are required, the proposed scheme provides a customer efficient and completely secure protection solution for untraceable electronic cash such that it is especially suitable for the situations

where the computation capabilities of the customers are limited and highly secure transaction environments are desired, such as mobile commerce or smart-card transactions with macro payment. This is because that we successfully integrate the protection mechanism into the underlying blind signature scheme through the common modulus.

Acknowledgement

I would like to thank the Editor-in-Chief and the anonymous reviewers for their valuable comments on this paper.

Appendix A. Algorithm 1.

Input = $(p,q,p',q',n,\theta,\sigma)$, where $\theta \in \{8,16\}$, p and q being two distinct large primes with $p \equiv q \equiv -1 \pmod{2\theta}$, $p'p \equiv 1 \pmod{q}$, $q'q \equiv 1 \pmod{p}$, $n = pq$, and σ being a QR in Z_n^*.
Output = φ, where φ is one of the θ-th roots of σ^{-1} in Z_n^*, i.e., $\varphi^\theta \equiv \sigma^{-1}$ or $\varphi^\theta\sigma \equiv 1 \pmod{n}$.
Begin
Step 1: input $(p,q,p',q',n,\theta,\sigma)$.
Step 2: compute $\varphi_1 = \sigma^{(p-1)-\frac{p+1}{2\theta}} \bmod p$.
Step 3: compute $\varphi_2 = \sigma^{(q-1)-\frac{q+1}{2\theta}} \bmod q$.
Step 4: compute $\varphi = (\varphi_1 q'q + \varphi_2 p'p) \bmod n$.
Step 5: output φ.
End.

Correctness of Algorithm 1

Since p and q are primes, $\sigma^{(p-1)-\frac{p+1}{2\theta}} \equiv \sigma^{-\frac{p+1}{2\theta}} (\bmod p)$ and $\sigma^{(q-1)-\frac{q+1}{2\theta}} \equiv \sigma^{-\frac{q+1}{2\theta}} (\bmod q)$, i.e., $\varphi_1 \equiv \sigma^{-\frac{p+1}{2\theta}} (\bmod p)$ and $\varphi_2 \equiv \sigma^{-\frac{q+1}{2\theta}} (\bmod q)$ [15,25]. The integer σ is a QR in Z_n^*, so ($\sigma \bmod p$) and ($\sigma \bmod q$), respectively, also are QR's in Z_p^* and Z_q^*, respectively, and we have that $\sigma^{\frac{p-1}{2}} \equiv 1 (\bmod p)$ and $\sigma^{\frac{q-1}{2}} \equiv 1 (\bmod q)$ [15,25]. Thus, $\varphi_1^\theta \equiv \sigma^{-\frac{p+1}{2}} \equiv (\sigma^{-1})(\sigma^{\frac{p-1}{2}})^{-1} \equiv (\sigma^{-1})(1)^{-1} \equiv \sigma^{-1} (\bmod p)$ and $\varphi_2^\theta \equiv \sigma^{-\frac{q+1}{2}} \equiv (\sigma^{-1})(\sigma^{\frac{q-1}{2}})^{-1} \equiv (\sigma^{-1})(1)^{-1} \equiv \sigma^{-1} (\bmod q)$. As ($\varphi \bmod p$) = φ_1 and ($\varphi \bmod q$) = φ_2, it follows that $\varphi^\theta \equiv \sigma^{-1} \pmod{n}$ by the Chinese Remainder Theorem [15,25].

The Computation Cost of Algorithm 1

Due to $p \equiv q \equiv -1 \pmod{2\theta}$, $\frac{p+1}{2\theta}$ and $\frac{q+1}{2\theta}$, respectively, can be obtained by performing 4-bit or 5-bit right-shift operations on $(p+1)$ and $(q+1)$, respectively. Besides, both $(p-1)-\frac{p+1}{2\theta}$ and $(q-1)-\frac{q+1}{2\theta}$ are positive integers. To compute φ_1 and φ_2, one modular exponentiation computation in Z_p^* and one modular exponentiation computation in Z_q^* are required. Since $|p| + |q| = |n|$,

the two modular exponentiation computations take nearly the same computation time as that of one modular exponentiation computation in Z_n^*. Considering Step 4, it needs only several modular multiplications for the derivation of φ. Hence, the computation cost of Algorithm 1 is about equivalent to that of a modular exponentiation computation in Z_n^*, i.e., T.

References

[1] M. Abe, E. Fujisaki, How to date blind signatures, in: K. Kwangjo, M. Tsutomu (Eds.), Advances in Cryptology-ASIACRYPT'96 (LNCS 1163), Springer-Verlag, Berlin, 1996, pp. 244–251.
[2] S. Brands, Untraceable off-line cash in wallets with observers, in: D.R. Stinson (Ed.), Advances in Cryptology-CRYPTO'93 (LNCS 773), Springer-Verlag, Berlin, 1993, pp. 302–318.
[3] J. Camenisch, J. Piveteau, M. Stadler, An efficient fair payment system protecting privacy, in: D. Gollmann (Ed.), ESORICS'94 (LNCS 875), Springer-Verlag, Berlin, 1994, pp. 207–215.
[4] J. Camenisch, J. Piveteau, M. Stadler, Blind signatures based on the discrete logarithm problem, in: A.D. Santis (Ed.), Advances in Cryptology-EUROCRYPT'94 (LNCS 950), Springer-Verlag, Berlin, 1995, pp. 428–432.
[5] D. Chaum, Blind signatures for untraceable payments, in: D. Chaum, R.L. Rivest, A.T. Sherman (Eds.), Advances in Cryptology-CRYPTO'82, Springer-Verlag, Berlin, 1983, pp. 199–203.
[6] D. Chaum, A. Fiat, M. Naor, Untraceable electronic cash, in: S. Goldwasser (Ed.), Advances in Cryptology-CRYPTO'88 (LNCS 403), Springer-Verlag, Berlin, 1990, pp. 319–327.
[7] J. Coron, D. Naccache, J. Stern, On the security of RSA padding, in: M. Wiener (Ed.), Advances in Cryptology-CRYPTO'99 (LNCS 1666), Springer-Verlag, Berlin, 1999, pp. 1–18.
[8] C. Fan, C. Lei, User efficient blind signatures, Electronics Letters 34 (6) (1998) 544–546.
[9] C. Fan, W. Chen, Y. Yeh, Blind signatures with double-hashed messages for fair electronic elections and ownership claimable digital cash, in: J. Filipe (Ed.), Enterprise Information Systems, Kluwer Academic, Dordrecht, 1999, pp. 191–197.
[10] C. Fan, W. Chen, Y. Yeh, A new electronic cash scheme based on blind signatures and asymmetric cryptosystems, Proceedings of the 14th International Conference on Information Networking, 2000, pp. 1B3.1–1B3.5.
[11] C. Fan, C. Lei, Cryptanalysis on improved user efficient blind signatures, Electronics Letters 37 (10) (2001) 630–631.
[12] N. Ferguson, Single term off-Line coins, in: T. Helleseth (Ed.), Advances in Cryptology-EUROCRYP'93 (LNCS 765), Springer-Verlag, Berlin, 1994, pp. 318–328.
[13] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, Technical Report, MIT Lab., Computer Science, Cambridge, Mass, 1995.
[14] A. Juels, Trustee tokens: simple and practical anonymous digital coin tracing, in: M. Franklin (Ed.), Financial Cryptography (LNCS 1648), Springer-Verlag, Berlin, 1999, pp. 29–45.
[15] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of applied cryptography, CRC Press LLC, Boca Raton, 1997.
[16] T. Okamoto, K. Ohta, Universal electronic cash, in: J. Feigenbaum (Ed.), Advances in Cryptology-CRYPTO'91 (LNCS 576), Springer-Verlag, Berlin, 1992, pp. 324–337.
[17] T. Okamoto, Provably secure and practical identification schemes and corresponding signature schemes, in: E.F. Brickell (Ed.), Advances in Cryptology-CRYPTO'92 (LNCS 740), Springer-Verlag, Berlin, 1992, pp. 31–53.

[18] B. Pfitzmann, M. Waidner, Strong loss tolerance of electronic coin systems, ACM Transactions on Computer Systems 15 (2) (1997) 194–213.
[19] D. Pointcheval, J. Stern, Provably secure blind signature schemes, in: K. Kwangjo, M. Tsutomu (Eds.), Advances in Cryptology-ASIACRYPT'96 (LNCS 1163), Springer-Verlag, Berlin, 1996, pp. 252–265.
[20] D. Pointcheval, J. Stern, New blind signatures equivalent to factorization, Proceedings of the 4th ACM Conference on Computer and Communication Security, 1997, pp. 92–99.
[21] M. Rabin, Digitalized signatures and public-key functions as intractable as factorization, Technical Report, MIT/LCS/TR212, MIT Lab., Computer Science, Cambridge, Mass, 1979.
[22] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM 21 (2) (1978) 120–126.
[23] Z. Shao, Improved user efficient blind signatures, Electronics Letters 36 (16) (2000) 1372–1374.
[24] A. Shamir, C. Schnorr, Cryptanalysis of certain variants of Rabin's signature scheme, Information Processing Letters 19 (1984) 113–115.
[25] J. Simmons, Contemporary cryptology: the science of information integrity, IEEE, New York, 1992.