Untraceable Fair Network Payment Protocols with Off-line TTP

Chih-Hung Wang

Department of Computer Science and Information Engineering
National Chiayi University
300 University Road, Chiayi, Taiwan 600, R.O.C.
wangch@mail.ncyu.edu.tw

Abstract. A fair network payment protocol plays an important role in electronic commerce. The *fairness* concept in payments can be illustrated as that two parties (e.g. customers and merchants) exchange the electronic items (e.g. electronic money and goods) with each other in a fair manner that no one can gain advantage over the other even if there are malicious actions during exchanging process. In the previous works of fair payments, the buyer is usually required to sign a purchase message which can be traced by everyone. The information about where the buyer spent the money and what he purchased would easily be revealed by this way. This paper employs two techniques of *off-line untraceable cash* and *designated confirmer signatures* to construct a new fair payment protocol, in which the untraceability (or privacy) property can be achieved. A *Restrictive Confirmation Signature Scheme* (RCSS) will be introduced and used in our protocol to prevent the interested persons except the off-line TTP (Trusted Third Party) from tracing the buyer's spending behavior.

Keywords: Cryptography, Electronic cash, Payment System, Undeniable Signature, Designated Confirmer Signatures, Electronic Commerce.

1 Introduction

How the two parties, buyer and merchant, exchange the currency and electronic goods through the network in a fair manner is the crux of the problem on electronic transactions. Since most of the electronic businesses are conducted on an open and insecure network, how to prevent the abnormal behavior, such as malicious termination of the payment process, becomes a critical security consideration on designing a fair payment protocol. A buyer who makes a payment in the network is usually worried that the merchant may refuse to deliver the soft goods though he has sent the money. On the other hand, a merchant will worry that he cannot receive the deserved money after the delivery of goods. Since these two parties do not trust each other, no one wants to send his secret data until receiving the other's.

Two approaches to the achievement of fair exchange have been proposed. The first one is that two parties exchange data simultaneously [EGL85,OO94].

A simplified example to provide simultaneity is that they disclose the secret data bit by bit. This kind of scheme has a drawback that it requires many steps of interactions for exchanging data. In addition, one of these two parties will have an advantage of obtaining one more bit if he maliciously aborts in the middle of the protocol. The second approach is that a trusted third party (TTP) is involved in the exchange process. A straightforward method is that an on-line TTP who acts as a mediator receives the data from both parties in each transaction and then forwards them to the accurate receivers [DGLW96,ZG96]. However, TTP would become a bottleneck on communications since he takes part in all transactions, including the normal cases in which two parties honestly deliver their data. To improve the performance, a novel model called the off-line TTP has been proposed. In this model, TTP is required to participate in the exchange protocols only when the abnormal terminations or faults occur [ASW00,ZG97,BDM98,BF98,Che98]. That means TTP is always able to solve the disputes between two parties but he need not take part in all transactions.

Previously, fair payments seemed to be achieved by use of fair exchange on signatures. For example, two parties can exchange the secret message (soft goods) and the signatures on purchase information. In [ASW98,BDM98,ASW00], a general concept of the fair exchange on signature with off-line TTP is explicated as that one party A sends the encrypted signature to B and convinces B that it is valid and can be decrypted by TTP, without revealing the content of signature. If B completes the verification, he will send his signature (or secret data) to A. In a normal case, A should send his correct signature to B after he received B's signature. However, if A maliciously aborts the protocol and refuses to send B his signature, B can deliver A's encrypted signature to the off-line TTP for decryption. The main technique used in these papers is called *verifiable encryption protocol* or *escrow system* [Sta96,Mao97]. However, a generic and efficient construction on verifiable encryption is difficult to implement. Bao et al. [BDM98] in their paper proposed a special implementation with the modified GQ signature algorithm, in which they claimed that the verifiable encryption protocol of the scheme was quite efficient. Unfortunately, Boyd et al. showed that the fairness could be destroyed because the receiver (or any observer) could directly calculate the sender's signature from the encrypted signature without the help of TTP [BF98].

Recently, Boyd et al. [BF98] and Chen [Che98] proposed the efficient fair exchange protocols with off-line TTP by using *verifiable confirmation signatures* (Boyd et al. called them designated converter signatures to emphasize their conversion property). A designated third party, e.g. TTP, can verify the original signatures with interactive protocol or convert the signatures into the self-authenticated signatures which can be verified by everyone. Their proposed schemes are generic constructions for fair exchange and can efficiently run over the Internet.

Our Contributions. Pervious works of fair exchange are not really suitable for many applications on network payments because they are only used to exchange the confidential data or signatures. Especially, many payment applications need to protect the buyer's purchase privacy, which has never been considered in the previous papers. In our view, a complete solution for fair payment should contain payment actions, such as electronic cash or network credit card method, instead of simply signing the purchase information. Our proposed protocol is the first work to provide a protection on buyer's privacy and it can be regarded as a process of fairly exchanging electronic coins and secret information. The main contributions in this paper are listed as follows:

1. Propose a generic model for *real* fair network payments.
2. Apply a subtle tool of *Restrictive Confirmation Signature Scheme* (RCSS) to achieve the property of untraceability.
3. Design a new technique of *pseudo e-coin* to achieve fairness of exchanging the electronic cash.
4. Demonstrate how to construct a practical and efficient fair network payment protocol based on the Brands' e-cash scheme [Bra93b].

The rest of the paper is organized as follows. We describe the basic model of untraceable fair payment protocol in Section 2. In Section 3, we introduce an useful scheme called Restrictive Confirmation Signature Scheme (RCSS), a basic component for establishing our new protocol. In Section 4, we combines the RCSS and the Brands' electronic cash scheme to realize our protocol. In Section 5, we show the security analysis and properties discussion. Finally, the concluding remarks and future researches are given in Section 6.

2 The Basic Model

We abstractly describe our works in this section. Assume that four parties: the buyer ($\mathcal{U}$), the merchant ($\mathcal{M}$), the bank ($\mathcal{B}$) and the trusted third party (TTP) are involved in the protocol. In a general e-cash scheme, *fairness* can not be achieved because the buyer is required to send *true* electronic coins (e-coins) to the merchant. Instead, this paper designs a technique of *pseudo e-coin* which can be converted to a true one by TTP. The buyer applies the Restrictive Confirmation Signature Scheme (RCSS) (described in Section 3) to sign an order agreement that contains the buyer's and the merchant's names, price of goods, purchase date/information and some other parameters. The RCSS can properly protect the buyer's purchase information by restricting the confirmer's confirmation capability on the signature.

Definition 1. (Restrictive Confirmation Signature Scheme (RCSS)). *Let $Sign_{DCS}(S, C, m)$, which is signed by S and can be confirmed by C, be a designated confirmer signature [Cha94] (or called a confirmation signature by [Che98]) on the message m. Assume that a group of verifiers $\mathcal{G} = \{V_i\}_{i=1,\dots,n}$ are pre-determined by S. We say that $Sign_{RCSS}(S, C, \mathcal{G}, m)$ is a restrictive confirmation signature on m if C can convince only some specified verifiers $V_i \in \mathcal{G}$ that $Sign_{RCSS}(S, C, \mathcal{G}, m)$ is valid and truly signed by S.*

Three procedures similar to a general e-cash (withdrawal, payment and deposit) are briefly depicted in the following. When a dispute occurs, the TTP is required to participate in an additional procedure *Disputes* to force the completion of the payment process.

Withdrawal. The buyer $\mathcal{U}$ withdraws the money from the bank $\mathcal{B}$. A blind signature applied here can guarantee the unlinkability for the bank. The withdrawal procedure in our protocol is the same as the one in the general e-cash scheme. After this procedure, $\mathcal{U}$ obtains an electronic coin which can be directly paid to the merchant.

Payment. The buyer $\mathcal{U}$ and the merchant $\mathcal{M}$ exchange the electronic money and goods in this procedure. We assume $\mathcal{U}$ and $\mathcal{M}$ negotiate an order agreement that contains merchandise items and price. The buyer $\mathcal{U}$ then sends enough pseudo e-coins and a signature of RCSS on the order agreement to $\mathcal{M}$. To prevent the merchant from maliciously delivering the flawed goods, the buyer doesn't send true e-coins to the merchant until he checks and accepts the goods.

1. The buyer $\mathcal{U}$ selects the goods from merchant $\mathcal{M}$'s web and signs an order agreement:
$$\theta = Sign_{RCSS}(\mathcal{U}, \mathcal{M}, TTP, OA),$$
where $OA = \{ID_{\mathcal{U}}, ID_{\mathcal{M}}, purchase\ date/information, goods\ description,$ $coin\ parameters\}$.
2. The buyer $\mathcal{U}$ sends the pseudo e-coins and θ for the goods to the merchant $\mathcal{M}$.
3. The merchant $\mathcal{M}$ verifies whether the pseudo e-coins and θ are valid. If both checks pass, $\mathcal{M}$ sends the goods to $\mathcal{U}$. The merchant $\mathcal{M}$ can gain a conviction in this step that he can prove the validity of θ to TTP and ask TTP convert the pseudo e-coins into true e-coins if some faults occur in the rest of payment process.
4. $\mathcal{U}$ checks the goods delivered by $\mathcal{M}$. If the goods is valid, $\mathcal{U}$ sends his true e-coins to $\mathcal{M}$.

Disputes. Two possible disputes may occur during payment. $\mathcal{M}$ may refuse to send $\mathcal{U}$ the goods or cheat $\mathcal{U}$ by sending flawed goods. In this case, $\mathcal{U}$ will not send the true e-coins to $\mathcal{M}$ if he does not receive or accept the goods. On the other hand, $\mathcal{U}$ may refuse to send the true e-coins to $\mathcal{M}$ after he receives the valid goods. If so, $\mathcal{M}$ will begin the following procedure to ask TTP convert the pseudo e-coins into true ones.

1. The merchant $\mathcal{M}$ sends pseudo e-coins, OA and θ to TTP and proves that θ is a valid signature and truly signed by $\mathcal{U}$. Note that no one except $\mathcal{M}$ and TTP can be convinced that θ is valid, since RCSS is applied to the construction of θ.
2. $\mathcal{M}$ privately sends goods to TTP. TTP checks whether the specification of the goods is consistent with the field of goods description written on OA. If yes, TTP sends $\mathcal{M}$ a transformation certificate ($TCer$) which can be used for the conversion of the pseudo e-coins.

An abnormal action is addressed here that $\mathcal{M}$ may abort the step 3 in the payment procedure and directly ask TTP to send him $TCer$ after he receives the pseudo e-coins. However, $\mathcal{M}$ must send TTP the valid goods since TTP has the responsibility to carefully check the goods specification.

Deposit. Generally, the merchant $\mathcal{M}$ can forward the payment transcript, including the true e-coins, to the bank. However, if the payment process is maliciously aborted by $\mathcal{U}$, $\mathcal{M}$ can sends the partial payment transcript with pseudo e-coins plus the transformation certificate ($TCer$) delivered by TTP to the bank for deposit.

3 The Restrictive Confirmation Signature Scheme

In the general designated confirmer signature [Cha94,Oka94,MS98,NMV99], a confirmer can help *every* recipient prove the validity of the signature to others. That means the confirmer has the complete capability of deciding who will benefit from being convinced by a signature. However, this property doesn't meet the requirements of our protocol. In this section, we will illustrate how to construct a Restrictive Confirmation Signature Scheme (RCSS). The basic structure of RCSS is similar to [WC03] but both schemes have different purposes. The concept of RCSS is that we disallow that the confirmer arbitrarily chooses the verifiers; the signer predetermines one or more verifiers whom the confirmer can convince later. We provide a nice approach to add the simulatability into an undeniable signature [CA89,Cha90,CHP92,GKR97]. Hence the signer can later create the proofs in an non-interactive way to delegate confirmer the capability of confirmation of the signature.

In the following, we first give some informal definitions and techniques used in this scheme.

Definition 2. (Trap-Door Commitment (also see [BCC88,JSI96])). *Let c be a function with input (y, u, v). The notation y denotes the public key of the user whose corresponding secret key is x, u is a value committed to and v is a random number. We say c is a trap-door commitment if and only if it satisfies the following requirements:*

1. *No polynomial algorithm, when given y, can find two different pairs of (u_1, v_1) and (u_2, v_2) such that $c(y, u_1, v_1) = c(y, u_2, v_2)$.*
2. *No polynomial algorithm, when given y and $c(y, u, v)$, can find u.*
3. *There exists a polynomial algorithm that, when given the secret x, (u_1, v_1) and a randomly selected number u_2, can find v_2 such that $c(y, u_1, v_1) = c(y, u_2, v_2)$ (That means the user who knows the secret x, given (u_1, v_1), can easily forge the committed value by changing u_1 into u_2).*

The following example was suggested by [BCC88,JSI96].

Trap-door commitment Example

Let p and q be two large primes and $q|p-1$. The notation g denotes a generator of the subgroup, G_q, of Z_p^* of prime order q. The recipient's secret key is $x \in {Z_q}^*$ and the corresponding public key is $y = g^x \mod p$. The sender randomly selects $v \in {Z_q}^*$ and commits the value $u \in Z_q$ into c as the following:

$$c = g^u y^v \mod p.$$

The sender sends (u, v) to the recipient for decommitting.

Trap-door Commitment for Multiple Recipients

Jakobsson et al. [JSI96] proposed an efficient trap-door commitment scheme for multiple recipients P_i , $i = 1, 2, \ldots, n$. They modified the commitment to be $c = g^u(\prod_{i=1}^{n} y_i)^v \mod p$, where y_i denotes P_i's public key. Each P_i would be convinced by the proof that u cannot be forged by others as long as he knows his secret key has not been compromised. Any other user would not gain this conviction since all $P_i, i = 1, \ldots, n$ can collude to cheat him.

Definition 3. (Message-dependent Proof of Equality of the Discrete Logarithm [Pet97]). *A message-dependent proof of equality of the discrete logarithm of y_1 to the base g_1 and y_2 to the base g_2 is a two-tuple $(w, z) = Proof_{LogEQ}(m, g_1, y_1, g_2, y_2)$, where $w = F(m||g_1||y_1||g_2||y_2||{g_1}^z {y_1}^w||{g_2}^z {y_2}^w)$ and F is a collision resistant hash function.*
This proof shows that the prover knows the discrete logarithm $x : log_{g_1}(y_1) \equiv log_{g_2}(y_2)$. To construct this proof, the prover randomly selects $k \in {Z_q}^*$ and calculates $w = F(m||g_1||y_1||g_2||y_2||{g_1}^k||g_2^k)$ and $z = k - xw \mod q$.

Definition 4. (Designated Verifier Message-dependent Proof of Equality of the Discrete Logarithm). *Let V denote a designated verifier who has a secret key/public key pair $(x_V, y_V = g^{x_V} \mod p)$. A designated verifier message-dependent proof of equality of the discrete logarithm of y_1 to the base g_1 and y_2 to the base g_2 is a four-tuple $(w, z, u, v) = Proof_{DV\,LogEQ}(m, c, g_1, y_1, g_2, y_2, y_V)$, where $w = F(m||c||g_1||y_1||g_2||y_2||{g_1}^z {y_1}^{(w+u)}||{g_2}^z {y_2}^{(w+u)})$ and $c = g^u {y_V}^v \mod p$ is a trap-door commitment.*
The prover, using this proof, only can convince the designated verifier V that he knows the discrete logarithm $x : log_{g_1}(y_1) \equiv log_{g_2}(y_2)$. To construct this proof, the prover randomly selects $u, v, k \in {Z_q}^*$ and calculates $c = g^u {y_V}^v \mod p$, $w = F(m||c||g_1||y_1||g_2||y_2||{g_1}^k||g_2^k)$ and $z = k - x(w + u) \mod q$.

Definition 5. (Interactive Bi-proof of Equality (see [FOO92,MS98])). *Fujioka et. al. in 1992 proposed an interactive bi-proof system that either proved $log_\alpha(Y) = log_\beta(Z)$ or proved $log_\alpha(Y) \neq log_\beta(Z)$. This proof system can be used to construct RCSS in which the confirmer can prove the validity of the signature to the pre-determined verifiers. We use $BP(\alpha, Y, \beta, Z)$ to represent this proof system. We omit the detail protocol here, the reader can refer to [FOO92].*

Construction of RCSS

The previous works of designated confirmer signatures used the general self-authenticated signature (e.g. RSA, Schnorr [Sch91] and extended Fait-Shamir scheme) to construct their schemes. However, it is difficult for these schemes to restrict the confirmer's confirmation capability. Here, we use the *message-dependent proof of equality* (in Definition 3 and Definition 4) and *non-interactive undeniable signature* [JSI96] to construct the RCSS. We also use $a = g^t \ mod \ p$ and $b = {y_C}^t \ mod \ p$, where y_C denotes the confirmer's public key, to add the simulatability to the signature. In addition, we slightly modify the *hinging method* described in the scheme of [Cha94] and [MS98]. The following procedure demonstrates how to pre-determine a single verifier for a signer; however, it is easy to construct an extended scheme to multiple verifiers.

- **System Setup.** The parameters p, q and g are the same ones described previously, and F_1, F_2 are two collision resistant hash functions. The secret key/public key pairs of the signer S, the confirmer C, the recipient R and the verifier V are $(x_S, y_S = g^{x_S} \ mod \ p)$, $(x_C, y_C = g^{x_C} \ mod \ p)$, $(x_R, y_R = g^{x_R} \ mod \ p)$ and $(x_V, y_V = g^{x_V} \ mod \ p)$, respectively.
- **Signing Protocol.** Assume the signer has signed a undeniable signature (a, b, δ) on message m related to the confirmer's public key, i.e., $a = g^t \ mod \ p$, $b = {y_C}^t \ mod \ p$ and $\delta = (F_1(m||a) + b)^{x_S} \ mod \ p$ (note that t is randomly selected by S). For delegating C the ability of confirming this signature, the signer randomly selects k, u, v_1, v_2 and constructs a proof of

  $$(w, z, u, v_1, v_2) = Proof_{DVLogEQ}(c, g, y_S, F_1(m||a) + b, \delta, y_V),$$

  where $c = (c_1||c_2)$, $c_1 = g^u {y_V}^{v_1} \ mod \ p$, $c_2 = g^u {y_C}^{v_2} \ mod \ p$, $w = F_2(c||g||y_S|| F_1(m||a) + b||\delta||g^k||(F_1(m||a) + b)^k)$ and $z = k - x_S(w + u) \ mod \ q$. Note that we eliminate the first parameter m in $Proof_{DVLogEQ}$ because the message has been included in other parameters: $F_1(m||a) + b$ and δ. Thus, the RCSS on m denotes $Sign_{RCSS}(S, C, V, m) = (a, b, u, v_1, v_2, w, z, \delta)$.
- **Proof by the Signer.** In the original definition of designated confirmer signature scheme, the signer can convince the recipient R that a confirmer C can help R prove the validity of the signature to V. However, according to our basic model in Section 2, the confirmer C also plays the role of the recipient R. That means C will be convinced that he is able to prove the validity of the signature to V in this procedure. C checks the proof by computing $c = ((g^u {y_V}^{v_1} \ mod \ p)||(g^u {y_C}^{v_2} \ mod \ p))$ and verifying

  $$w \stackrel{?}{=} F_2(c||g||y_S||F_1(m||a) + b||\delta||g^z {y_S}^{(w+u)}||(F_1(m||a) + b)^z \delta^{(w+u)}).$$

  To prove the relation of a and b, the signer needs to run the interactive protocol of bi-proof $BP(g, a, y_C, b)$ (see Definition 5) to show $log_g(a) \equiv log_{y_C}(b)$.
- **Confirmation Protocol.** The confirmer C can prove the validity of the signature to V by running the interactive protocol bi-proof $BP(g, y_C, a, b)$ with V to show $log_g(y_C) \equiv log_a(b)$. The verifier V needs to check whether the signature $(a, b, u, v_1, v_2, w, z, \delta)$ is created properly, and he can be convinced that the signature is valid if he accepts the proof of $BP(g, y_C, a, b)$.
- **Conversion Protocol.** The confirmer can convert the designated confirmer signature to a general non-interactive undeniable signature. Since the signer

has constructed the designated verifier proof in a non-interactive way, V can check the validity of the signature by himself. The verifier V no longer needs to ask C to help him verify the signature. Here, C randomly selects $\sigma \in Z_q^*$ and computes $E = a^\sigma \ mod \ p$ and $T = \sigma + x_C F(a, E) \ mod \ q$, where F is also a hash function. The confirmer sends (E, T) to the verifier V, thus, V can verify $a^T \stackrel{?}{=} Eb^{F(a,E)}$ [Cha94].

Security of RCSS

Here, some security properties will be considered for RCSS.

Unforgeability. The forgeability problems that the intruder I tries to forge (a^*, b^*, δ^*) without access to secret key x_S, can be illustrated with two scenarios. The first one is that I selects a message m^*, a^* and computes $b^* = F_1(m||a) + b - F_1(m^*||a^*)$. However, the b^* which I can easily calculate would not have the same discrete logarithm as a^* has because F_1 is a collision resistant hash function whose output is approximately random. The second one is that I randomly selects $t^* \in {Z_q}^*$ and compute $a^* = g^{t^*}$ and $b^* = {y_C}^{t^*}$. In this attack scenario, I can not find a proper m^* to satisfy the equation $F_1(m^*||a^*) + b^* = F_1(m||a) + b$ since inverting an one-way hash function F_1, given its output, is computationally infeasible.

Indistinguishability. Given a random number a^*, a simulated signature on the message m^* can be represented as $(a^*, b^*, u, v1, v2, w, z, \delta)$ where $b^* = F_1(m||a) + b - F_1(m^*||a^*)$. The verifier cannot distinguish between the correct signature and simulated signature because he knows nothing about the discrete logarithm of a^* to the base g and b^* to the base y_C. Hence, without confirmer's help, the verifier would not be convinced that both discrete logarithms of a^* and b^* are equal. The indistinguishability of RCSS can also be proved by Decision-Diffie-Hellman assumption [MS98].

The following lemma shows that no one except the confirmer C and the designated verifier V can be convinced that the RCSS is correctly constructed and truly signed by S. Note that C and V can be convinced by the proof of the signature because they know their secret keys have not been compromised; however, others cannot obtain this conviction since they know that C and V are able to collude to create a simulated transcript to cheat them.

Lemma 1. (Simulating Transcripts of RCSS). *The confirmer C and designated verifier V can collude to create a simulated transcript of RCSS without accessing the signer's secret key x_S. Assume that V randomly selects α_1 and computes $c_1 = g^{\alpha_1} \ mod \ p$, and C randomly selects α_2 and computes $c_2 = g^{\alpha_2} \ mod \ p$. Thus they can compute the following simulated transcript by cooperatively choosing the random numbers $\beta, \tau, z \in {Z_q}^*$:*

$$c = (c_1||c_2),$$
$$a = g^\tau \ mod \ p,$$
$$b = {y_C}^\tau \ mod \ p,$$
$$w = F_2(c||g||y_S||F_1(m^*||a) + b||\delta^*||g^z {y_S}^\beta||(F_1(m^*||a) + b)^z \delta^{*\beta}),$$

$u = (\beta - w) \ mod \ q.$

V and C individually computes v_1 and v_2 as below:

$$v_1 = (\alpha_1 - u)(x_V)^{-1} \ mod \ q,$$
$$v_2 = (\alpha_2 - u)(x_C)^{-1} \ mod \ q.$$

4 The Realization of Our Fair Network Payment Model

Brands in 1993 proposed a nice approach to untraceable electronic cash [Bra93b]. In this section, we will present an untraceable fair payment protocol based on a modification of Brands scheme. We develop a pseudo e-coin technique combined into the payment procedure. Some mathematic definitions are omitted here, and the reader can refer [Bra93b] for further details.

The concept of pseudo e-coin technique is to create a designated confirmer signature (DCS) by which the merchant can be convinced that there exists a trusted third party (TTP) who can convert DCS into a self-authenticated signature. Therefore, if the merchant later does not receive the true e-coins from the buyer, he would ask TTP for a transformation certificate $TCer$.

We explicate an off-line fair payment in the following procedures. For simplifying the notation, we redefine all symbols in this section except some common parameters such as p, q and g (Note that the symbols used in this section have different definitions from that in Section 3).

Setup. Let p and q be two large primes as defined in Section 3. The bank $\mathcal{B}$ publishes a generator-tuple (g, g_1, g_2) in G_q and two collision-resistant hash functions $\mathcal{H} : G_q \times G_q \times G_q \times G_q \times G_q \times G_q \rightarrow Z_q{}^*$ and $\mathcal{H}_0 : G_q \times G_q \times ID \times DATE/TIME \rightarrow Z_q{}^*$. $\mathcal{B}$ also generates a random number $x_\mathcal{B} \in Z_q^*$ as his secret key corresponding to a public key $y_\mathcal{B} = g^{x_\mathcal{B}} \ mod \ p$.

Account Opening. The buyer $\mathcal{U}$ randomly selects $u_1 \in Z_q{}^*$ and transmits $I = g_1{}^{u_1} \ mod \ p$ to $\mathcal{B}$ if $Ig_2 \neq 1$. The identifier I used to uniquely identify $\mathcal{U}$ can be regarded as the account number of $\mathcal{U}$. Then $\mathcal{B}$ publishes $g_1{}^{x_\mathcal{B}} \ mod \ p$ and $g_2{}^{x_\mathcal{B}} \ mod \ p$ so that $\mathcal{U}$ can compute $z = (Ig_2)^{x_\mathcal{B}} = (g_1{}^{x_\mathcal{B}})^{u_1} g_2{}^{x_\mathcal{B}} \ mod \ p$ for himself [1].

Withdrawal. The buyer $\mathcal{U}$ performs the following protocol to withdraw a single e-coin from the bank:

1. $\mathcal{B}$ randomly selects $w \in Z_q^*$ and sends $e_1 = g^w \ mod \ p$ and $e_2 = (Ig_2)^w \ mod \ p$ to $\mathcal{U}$.

[1] Chan et al. [CFMT96] have proposed a problem of mis-representation of identities for Brands' scheme (Brands commented that it is only an inadvertent omission and the similar result has been presented in [Bra93a]). This problem can be efficiently solved by applying a minimal-knowledge proof to prove the correct construction of I during the account opening stage.

2. $\mathcal{U}$ randomly selects s, x_1 and x_2 in Z_q^* and computes $A = (Ig_2)^s \ mod\ p$, $B = {g_1}^{x_1}{g_2}^{x_2} \ mod\ p$ and $z' = z^s \ mod\ p$. $\mathcal{U}$ also randomly selects u, v and t_c in Z_q^* and computes ${e_1}' = {e_1}^u g^v \ mod\ p$, ${e_2}' = {e_2}^{su} A^v \ mod\ p$ and $(a_c, b_c) = (g^{t_c} \ mod\ p, {y_{TTP}}^{t_c} \ mod\ p)$. Then $\mathcal{U}$ sends $c = c'/u \ mod\ q$ to $\mathcal{B}$, where $c' = \mathcal{H}(A, B, z', {e_1}', {e_2}', b_c) + a_c \ mod\ q$. Note that (a_c, b_c) is a pair of confirmation parameters.
3. $\mathcal{B}$ sends $r = cx_{\mathcal{B}} + w \ mod\ q$ to $\mathcal{U}$.
4. $\mathcal{U}$ verifies whether $g^r = {y_{\mathcal{B}}}^c e_1 \ mod\ p$ and $(Ig_2)^r = z^c e_2 \ mod\ p$. If the verification holds, $\mathcal{U}$ accepts and computes $r' = ru + v \ mod\ q$. Note that $< A, B, (z', {e_1}', {e_2}', r', a_c, b_c) >$ represents a single pseudo e-coin.

Payment. The buyer $\mathcal{U}$ and the merchant $\mathcal{M}$ exchange the e-coins and the soft goods in this procedure. The following protocol will be done (Note that we add the subscripts to some symbols to represent the multiple e-coins).

1. The buyer $\mathcal{U}$ selects goods and signs an order agreements
$$\theta = Sign_{RCSS}(\mathcal{U}, \mathcal{M}, TTP, OA),$$
where $OA = \{ID_{\mathcal{U}}, ID_{\mathcal{M}}, purchase\ date/information, goods\ description, (A_i, B_i)_{i=1,2,\ldots,n}\}$ and n denotes the number of e-coins for the goods which $\mathcal{U}$ wants to buy.
2. The buyer $\mathcal{U}$ sends the unused e-coins $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}) >$, for $i = 1, 2, \ldots, n$, to $\mathcal{M}$.
3. The merchant $\mathcal{M}$ verifies the pseudo e-coins and θ. If all of them are valid and $A_i \neq 1$, for $i = 1, 2, \cdots, n$, then he sends $d_i = \mathcal{H}_0(A_i, B_i, ID_{\mathcal{M}}, date/time)$ to $\mathcal{U}$.
4. The buyer $\mathcal{U}$ sends $k_{1i} = d_i(u_{1i}s_i) + x_{1i} \ mod\ q$ and $k_{2i} = d_i s_i + x_{2i} \ mod\ q$, for $i = 1, 2, \cdots, n$, to the merchant $\mathcal{M}$. In addition, the buyer $\mathcal{U}$ must run the interactive protocol of bi-proof $BP(g, a_{ci}, y_{TTP}, b_{ci})$ with $\mathcal{M}$ to show all $log_g(a_{ci}) \equiv log_{y_{TTP}}(b_{ci})$.
5. The merchant $\mathcal{M}$ will accept these pseudo e-coins and payment transcripts $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}), (d_i, k_{1i}, k_{2i}) >$, for $i = 1, 2, \ldots, n$, if the following verifications hold:
$$\begin{aligned} &g^{r_i'} = {y_{\mathcal{B}}}^{\mathcal{H}(A_i, B_i, z_i', e_{1i}', e_{2i}', b_{ci}) + a_{ci}} e_{1i}', \\ &{A_i}^{r_i'} = {z_i'}^{\mathcal{H}(A_i, B_i, z_i', e_{1i}', e_{2i}', b_{ci}) + a_{ci}} e_{2i}', \text{ and} \\ &{g_1}^{k_{1i}} {g_2}^{k_{2i}} = {A_i}^{d_i} B_i. \end{aligned}$$
If the above verifications pass, the merchant $\mathcal{M}$ sends the soft goods to the buyer $\mathcal{U}$.
6. The buyer $\mathcal{U}$ checks the soft gooods delivered by $\mathcal{M}$. If it is flawless, he releases t_{ci}, for $i = 1, 2, \cdots, n$, to the merchant $\mathcal{M}$. Since each one can check $a_{ci} = g^{t_{ci}} \ mod\ p$ and $b_{ci} = {y_{TTP}}^{t_{ci}} \ mod\ p$ by himself, the coin $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}, t_{ci}), (d_i, k_{1i}, k_{2i}) >$ denotes a *true* e-coin that can be directly cashed from the bank.

Disputes. If $\mathcal{U}$ refuses to send t_{ci} to the merchant $\mathcal{M}$ (see the Step 6 in the Payment procedure), $\mathcal{M}$ will begin the dispute process in which the TTP can convert the pseudo e-coins into the true e-coins.

1. The merchant $\mathcal{M}$ sends the order agreement OA, the signature θ, soft goods and pseudo e-coins $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}), (d_i, k_{1i}, k_{2i}) >$, for $i = 1, 2, \cdots, n$, to TTP.
2. The TTP checks the validity of the soft goods, pseudo e-coins and signature θ. If the pseudo e-coins are constructed properly, the soft goods transmitted from $\mathcal{M}$ is consistent with the description in OA, and θ is valid, TTP sends $\mathcal{M}$ a transformation certificate $TCer = (E_{ci}, T_{ci})$, for $i = 1, 2, \cdots, n$, to $\mathcal{M}$, where $E_{ci} = a_{ci}^{\sigma_i}$ $mod\ p$ (σ_i is a random number selected by TTP) and $T_{ci} = \sigma_i + x_{TTP} F(a_{ci}, E_{ci})$ $mod\ q$. The transformation certificate can be used to verify the relation of a_{ci} and b_{ci} by the following equation:
$$a_{ci}^{T_{ci}} \stackrel{?}{=} E_{ci} {b_{ci}}^{F(a_{ci}, E_{ci})} \ mod\ p$$
3. TTP sends the soft goods to the buyer $\mathcal{U}$.

Deposit. In a normal case, $\mathcal{M}$ forwards the payment transcript and the true e-coins $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}, t_{ci}), (d_i, k_{1i}, k_{2i}) >$, for $i = 1, 2, \cdots, n$, to the bank for deposit. Nevertheless, if the buyer $\mathcal{U}$ maliciously aborts the payment process, $\mathcal{M}$ can start the dispute process to acquire the $TCer$ from TTP. In this situation, the pseudo e-coins $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}), (d_i, k_{1i}, k_{2i}) >$ plus $TCer = (E_{ci}, T_{ci})$, for $i = 1, 2, \cdots, n$, can be the valid tokens for deposit. We also can regard $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}), (d_i, k_{1i}, k_{2i}), (E_{ci}, T_{ci}) >$ as a true e-coin with different form.

5 Security Issues

The security of our new protocol relies on Brands' e-cash scheme and RCSS. The following properties are provided to prove the *fairness* and *untraceability* which are both pivotal features in our protocol.

Proposition 1. (Unforgeability). *No one except $\mathcal{U}$ can create his own pseudo e-coins $< A_i, B_i, (z_i', e_{1i}', e_{2i}', r_i', a_{ci}, b_{ci}), (d_i, k_{1i}, k_{2i}) >$, for $i = 1, 2, \ldots, n$.*

This proposition holds because the Brand's e-cash scheme is secure. The possible scenario of forging the e-coins is that the attacker randomly selects $\bar{u_{1i}}$, $\bar{s_i}$, $\bar{x_{1i}}$ and $\bar{x_{2i}}$ in Z_q^* and computes $\bar{A_i} = ({g_1}^{\bar{u_{1i}}} g_2)^{\bar{s_i}}$ $mod\ p$ and $\bar{B_i} = {g_1}^{\bar{x_{1i}}} {g_2}^{\bar{x_{2i}}}$ $mod\ p$. In this case, the attacker can randomly select $\bar{z_i'}$, $\bar{r_i'}$ and λ_i to compute $\bar{e_{1i}'} = g^{\bar{r_i'}} {y_\mathcal{B}}^{-\lambda_i}$ and $\bar{e_{2i}'} = \bar{A_i}^{\bar{r_i'}} \bar{z_i'}^{-\lambda_i}$. The purpose of the attacker is to find the proper a_{ci}^* and b_{ci}^* such that $\lambda_i = \mathcal{H}(\bar{A_i}, \bar{B_i}, \bar{z_i'}, \bar{e_{1i}'}, \bar{e_{2i}'}, b_{ci}^*) + a_{ci}^*$. However, though the attacker can easily calculate $a_{ci}^* = \lambda_i - \mathcal{H}(\bar{A_i}, \bar{B_i}, \bar{z_i'}, \bar{e_{1i}'}, \bar{e_{2i}'}, b_{ci}^*)$ by randomly selecting a value of b_{ci}^*, it is computationally infeasible for the attacker to find a_{ci}^* and b_{ci}^* which have the same discrete logarithm because $\mathcal{H}$ is a collision resistant hash function whose output is approximately random.

Proposition 2. (Indistinguishability). *No one can distinguish between a valid pseudo e-coin and a simulated one without the help of the buyer or TTP.*

According to Proposition 1, a simulated pseudo e-coin can be represented as $< \bar{A}_i, \bar{B}_i, (\bar{z_i'}, \bar{e_{1i}'}, \bar{e_{2i}'}, \bar{r_i'}, a^*_{ci}, b^*_{ci}), (d_i, \bar{k_{1i}} = d_i(\bar{u_{1i}}\bar{s_i}) + \bar{x_{1i}}, \bar{k_{2i}} = d_i\bar{s_i} + \bar{x_{2i}}) >$. Any interested party, such as a bank, cannot distinguish between a properly constructed pseudo e-coin and a simulated pseudo e-coin without the help of the buyer or TTP, because the bank knows nothing about the discrete logarithm of a^*_{ci} to the base g and b^*_{ci} to the base y_{TTP}. That means the bank cannot be convinced that the discrete logarithms of both a^*_{ci} and b^*_{ci} are equal. This property indicates the fairness that even if the buyer $\mathcal{U}$ sent the pseudo e-coins to the merchant $\mathcal{M}$ before he receives the soft goods, the merchant $\mathcal{M}$ cannot gain the advantage over $\mathcal{U}$.

Proposition 3. (Convertibility). *If $\mathcal{M}$ accepts the pseudo e-coins, it is guaranteed that TTP can later convert the pseudo e-coins into the true e-coins which can be directly deposited in the bank.*

This proposition can be proven by the confirmation signatures [Cha94,MS98]. The merchant $\mathcal{M}$ cannot accept an invalid pseudo e-coin except with negligible probability.

Lemma 2. (Fairness). *If the propositions of unforgeability, indistinguishability, and convertibility hold for our newly proposed payment protocol, it can be guaranteed that, at the end of the transaction, the buyer $\mathcal{U}$ can obtain the soft goods if and only if the merchant $\mathcal{M}$ can gain the equivalent true e-coins.*

Clearly, if two parties of $\mathcal{U}$ and $\mathcal{M}$ are honest, the fairness can be achieved without interacting with TTP. The rest of the condition is that one of $\mathcal{U}$ and $\mathcal{M}$ is dishonest. The unforgeability can guarantee that $\mathcal{U}$ cannot fool $\mathcal{M}$ by delivering the invalid pseudo e-coins, and the convertibility can prevent $\mathcal{U}$ from refusing to send true e-coins or sending the forged e-coins to $\mathcal{M}$. On the other hand, if $\mathcal{M}$ is dishonest, he may refuse to send valid goods to $\mathcal{U}$ after he receives the valid pseudo e-coins. However, because of the indistinguishability, $\mathcal{M}$ cannot receive the useful e-coins for deposit if he cheats during the payment procedure.

Lemma 3. (Untraceability). *No one except $\mathcal{M}$ and TTP can confirm the signature θ. That means only $\mathcal{M}$ and TTP can be convinced that the order agreement OA is valid.*

This lemma holds because the signature θ is created by RCSS. Thus $\mathcal{M}$ can only convince TTP that θ is really signed by $\mathcal{U}$. The security of RCSS has been discussed in Section 3.

Lemma 4. (Unlinkability). *The bank or other parties can not link a coin $< A_i, B_i, (z_i', {e_{1i}}', {e_{2i}}', {r_i}', a_{ci}, b_{ci}) >$ to the original owner.*

This lemma can be proven by using blind signature property of withdrawal procedure in [Bra93b].

Coin Size. Compared to the Brands' scheme, the individual coin of our protocol has extra three items: a_c, b_c and t_c. The total size of these items is $2|p| + |q|$. Especially, in the dispute condition, TTP is required to release $TCer$ with the size of $|p| + |q|$.

6 Conclusions

Electronic cash is considered to have a significant advantage over network credit card because the former can properly protect the buyer's payment privacy. In the proposed paper, we have presented a general model in which two parties can fairly exchange the electronic cash and soft goods. Our new scheme is also the first one that can provide the untraceability property on fair payments.

The future research is addressed here that we are planning to design the fair payment protocols with other payment tools, such as the electronic check and the divisible electronic cash. The privacy property, for which we have constructed a generic model, is a critical issue on the design of our future work.

Acknowledgement

This work was supported in part by National Science Council of Republic of China under contract NSC91-2213-E-415-005.

References

[ASW98] N. Asokan, V. Shoup, and M. Waidner. Optimistic Fair Exchange of Digital Signatures. In Advances in Cryptology - proceedings of Eurocrypt'98, Lecture Notes in Computer Science (LNCS) 1403, pages 591-606, Springer-Verlag, 1998.

[ASW00] N. Asokan, V. Shoup, and M. Waidner. Optimistic Fair Exchange of Digital Signatures. IEEE Journal on Selected Areas in Communications, vol. 18, pages 591-610, Apr. 2000.

[BDM98] F. Bao, R. H. Deng, W. Mao. Efficient and Practical Fair Exchange Protocols with Off-line TTP. Proceedings of the 1998 IEEE Symposium on Security and Privacy. IEEE Computer Press, pages 77-85, Oakland, CA, May 1998.

[BF98] C. Boyd and E. Foo. Off-line Fair Payment Protocols Using Convertible Signature. In Advances in Cryptology - proceedings of Asiacrypt'98, pages 271-285, Springer-Verlag, 1998.

[Bra93a] S. Brands. An Efficient Off-line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. ftp://ftp.cwi.nl/pub/CWIreports/AA/CS-R9323.pdf.

[Bra93b] S. Brands. Untraceable Off-line Cash in Wallets with Observers. In Advances in Cryptology - proceedings of Crypto'93, Lecture Notes in Computer Science (LNCS) 773, pages 302-318, Springer-Verlag, 1993.

[BCC88] G. Brassard, D. Chaum, C. Crepeau. Minimum Disclosure Proofs of Knowledge. Journal of Computer and System Sciences, Vol. 37, No. 2, pages 156-189, 1988.

[CFMT96] A. Chan, Y. Frankel, P. MacKenzie and Y. Tsiounis. Mis-representation of Identities in E-cash Schemes and how to Prevent it. In Advances in Cryptology - proceedings of Asiacrypt'96, Lecture Notes in Computer Science (LNCS) 1163, pages 276-285, Springer-Verlag, 1996.

[Cha90] D. Chaum. Zero-knowledge Undeniable Signature. In Advances in Cryptology - proceedings of Eurcorypt'90, Lecture Notes in Computer Science (LNCS) 473, pages 458-464, Springer-Verlag, 1990.

[Cha94] D. Chaum. Designated Confirmer Signatures. In Eurocrypt'94, pages 86-91, 1994.

[CHP92] D. Chaum, E. van Heijst and B. Pfitzmann. Cryptographically Strong Undeniable Signers, Unconditionally Secure for the Signer. In Advances in Cryptology - proceedings of Crypto'91, Lecture Notes in Computer Science (LNCS) 576, pages 470-484, Springer-Verlag, 1992.

[CA89] David Chaum and Hans Van Antwerpen: Undeniable Signature. In Advances in Cryptology - proceedings of Crypto'89, Lecture Notes in Computer Science (LNCS) 435, pages 212-217, Springer-Verlag, 1989.

[Che98] L. Chen. Efficient Fair Exchange with Verifiable Confirmation of Signatures. In Advances in Cryptology - proceedings of Asiacrypt'98, pages 286-299, Springer-Verlag, 1998.

[DGLW96] R. H. Deng, L. Gong, A. A. Lazar and W. Wang. Practical Protocol for Certified Electronic Mail. Journal of Network and Systems Management, vol. 4, no. 3, pages 279-297, 1996.

[EGL85] S. Even, O. Goldreich and A. Lempel. A Randomized Protocol for Signing Contracts. CACM, vol. 28, no. 6, pages 637-647, 1985.

[FOO92] A. Fujioka, T. Okamoto, K. Ohta. Interactive Bi-Proof Systems and Undeniable Signature Schemes. In Advances in Cryptology - proceedings of Eurocrypt'91, Lecture Notes in Computer Science, pages 243-256, Springer-Verlag, 1992.

[GKR97] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-Based Undeniable Signatures. In Advances in Cryptology - proceedings of Crypto'97, Lecture Notes in Computer Science (LNCS) 1294, pages 132-149, Springer-Verlag, 1997.

[GKR99] R. Gennaro, H. Krawczyk and T. Rabin. Undeniable Certificates. Electronic Letters, vol. 35, no. 20, pages 1723-1724, Sep. 1999.

[JSI96] M. Jakobsson, K. Sako and R. Impagliazzo. Designated Verifier Proofs and Their Application. In Advances in Cryptology - proceedings of EuroCrypt'96, Lecture Notes in Computer Science (LNCS) 1070, pages 143-154, Springer-Verlag, 1996.

[Mao97] W. Mao. Publicly Verifiable Partial Key Escrow. In ACISP'97, pages 240-248, Springer-Verlag, 1997.

[MS98] M. Michels and M. Stadler. Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In Advances in Cryptology - Eurocrypt'98, Lecture Notes in Computer Science (LNCS) 1403, pages 406-421, Springer-Verlag, 1998.

[NMV99] K. Nguyen, Y. Mu, and V. Varadharajan. Undeniable Confirmer Signature. Information Security - Proceedings of Second International Workshop, ISW'99, Lecture Notes in Computer Science (LNCS) 1729, pages 235-246, Springer-Verlag, 1999.

[Oka94] T. Okamoto. Designated Confirmer Signatures and Public-key Encryption Are Equivalent. In Advances in Cryptology - Crypto'94, Lecture Notes in Computer Science (LNCS) 839, pages 61-74, Springer-Verlag, 1994.

[OO94] T. Okamoto and K. Ohta. How to Simultaneously Exchange Secrets by General Assumption. Proceedings of 2nd ACM Conference on Computer and Communications Security, pages 184-192, 1994.

[Pet97] H. Petersen. How to Convert any Digital Signature Scheme into a Group Signature Scheme, to appear in Security Protocol'97, LNCS, Spring Verlag, 1997.

[PS96] D. Pointcheval, J. Stern. Security Proofs for Signature. In Advances in Cryptology - proceedings of Eurocrypt'96, Lecture Notes in Computer Science (LNCS) 1070, pages 387-398, Springer-Verlag, 1996.

[Sch91] C. P. Schnorr. Efficient Signature Generation for Smart Cards. Journal of Cryptology, vol. 4, no. 3, pages 161-174, 1991.

[Sta96] M. Stadler. Publicly Verifiable Secret Sharing. In Advances in Cryptology - proceedings of Eurocrtpt'96, Lecture Notes in Computer Science (LNCS) 1070, pages 190-199, Springer-Verlag, 1996.

[WC03] C.-H. Wang and Y.-C Chen. Proxy Confirmation Signatures. Informatica (accepted), 2003.

[ZG96] J. Zhou and D. Gollmann. A Fair Non-repudiation Protocol. Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Press, pages 55-61, Oakland, CA, 1996.

[ZG97] J. Zhou and D. Gollmann. An Efficient Non-repudiation Protocol. Proceedings of the 1997 IEEE Computer Security Foundations Workshop (CSFW 10). IEEE CS Press, pages 126-132, 1997.