Building a consumer scalable anonymity payment protocol for Internet purchases

Hua Wang Jinli Cao
Department of Maths & Computing
University of Southern Queensland
Toowoomba QLD 4350 Australia
(wang, cao)@usq.edu.au
Yahiko Kambayashi
Department of Social Informatics
Kyoto University
Kyoto, Japan
yahiko@db.soc.i.kyoto-u.ac.jp

Abstract

This paper proposes a secure, scalable anonymity and practical payment protocol for Internet purchases. The protocol uses electronic cash for payment transactions. In this new protocol, from the viewpoint of banks, consumers can improve anonymity if they are worried about disclosure of their identities. An agent provides a higher anonymous certificate and improves the security of the consumers. The agent will certify re-encrypted data after verifying the validity of the content from consumers, but with no private information of the consumers required. With this new method, each consumer can get the required anonymity level, depending on the available time, computation and cost.

We also analyse how to prevent a consumer from spending a coin more than once and how to use the proposed protocol for Internet purchases. After comparing with another scheme and discussing the properties of the new payment protocol, the new method will be proved that it is more efficient and can prevent from eavesdropping, tampering and "perfect crime" effectively. It is promising for electronic trades through the Internet.

Keywords: Electronic-cash, Anonymity, Traceability, Hash function.

1 Introduction

Recent advances in the Internet and WWW have enabled rapid development in e-commerce. More and more businesses begin to develop or adopt e-commerce systems to support their selling/business activities. While this brings convenience for both consumers and vendors, many consumers have concerns about security and their private information when purchasing over the Internet, especially with electronic payment or e-cash payment. Consumers often prefer to have some degree of anonymity when shopping over the Internet.

There are a number of proposals for electronic cash systems. All of them lack flexibility in anonymity. David Chaum [5] first proposed an on-line payment system that will guarantee receiving valid coins. This system provides some levels of anonymity against a collaboration of shops and banks. However, users have no flexible anonymity and banks have to keep a very big database for users and coins. Another on-line CyberCoin (http://www.cybercash.com) approach allows clients to make payments by signing fund transfer requests to merchants. The merchants submit the signed requests to the bank for authorization of the payments. The CyberCoin protocol, however, is not fully anonymous since it allows the issuing bank to track every purchase. Furthermore, the scalability of the CyberCoin protocol is questionable since it relies on the availability of a single on-line bank. NetBill [9] extends the above payment mechanism by supporting goods atomicity and certified delivery. The drawbacks of NetBill protocol are the addition of extra messages and the significant increase in the amount of encryption used. The most sophisticated protocol is the SET protocol [13], which was designed to facilitate credit card transactions over the Internet. SET security comes at a considerable computation and communication cost. SET, unlike other simpler on-line protocols, does not offer full anonymity, non-repudiation or certified delivery.

Proceedings of the 12th Int'l Wrkshp on Research Issues in Data Engineering: Engineering e-Commerce/ e-Business Systems (RIDE'02)
1066-1395/02 $17.00 © 2002 **IEEE**

Systems mentioned above are on-line payment systems. They need sophisticated cryptographic functions for each coin, and require additional computational resources for the bank to validate the purchases. Forcing the bank to be on-line at payment is a very strict requirement. On-line payment systems protect the merchant and the bank against customer fraud, since every payment needs to be approved by the customer's bank. This will increase the computation cost, proportional to the size of the database of spent coins. If a large number of people start using the system, the size of this database could become very large and unmanageable. Keeping a database of every coin ever spent in the system is not a scalable solution. Digicash [6] plans to use multiple banks each minting and managing their own currency with inter-bank clearing to handle the problems of scalability. It seems likely that the host bank machine has an internal scalable structure so that it can be set up not only for a 10,000 user bank, but also for a 1,000,000 user bank. Under the circumstances, the task of maintaining and querying a database of spent coins is probably beyond today's state-of-the-art database systems.

In an off-line protocol, the merchant verifies the payment using cryptographic techniques, and commits the payment to the payment authority later in an off-line batch process. Off-line payment systems were designed to lower the cost of transactions due to the delay in verifying batch processes. Off-line payment systems, however, suffer from the potential of double spending, whereby the electronic currency might be duplicated and spent repeatedly.

The first off-line anonymous electronic cash was introduced by Chaum, Fiat and Naor [8]. The security of their scheme relied on some restricted assumptions such as requiring a function which is similar to random oracle and maps from the second argument onto a special range. There is also no formal proof attempted. Although hardly practical, their system demonstrated how off-line e-cash can be constructed and laid the foundation for more secure and efficient schemes. In 1995, Chan, Frankel and Tsiounis [4] presented a provable secure off-line e-cash scheme that relied only on the security of RSA [17]. This scheme extended the work of Franklin and Yung [12] who aimed to achieve provable security without the use of general computation protocols. The anonymity of consumers is based on the security of RSA and it cannot be changed dynamically after the system is established. NetCents [16] proposed a lightweight, flexible and secure protocol for micropayments of electronic commerce over the Internet. This protocol is designed only to support purchases ranging in value from a fraction of a penny and up.

In 2000, David Pointcheval [15] presented a payment scheme in which the consumer's identity can be found any time by a certification authority. So the privacy of a consumer cannot be protected.

Moreover, as mentioned above, the on-line e-cash payments need more computing resources. Most of the previously designed off-line schemes are only for micropayments. They rely on the heuristic proofs of security and therefore do not formally prevent fraud and counterfeit money. Under these conditions, most on-line and off-line payment schemes do not provide efficient anonymity for consumers. Hence, a new payment scheme for the purchases over the Internet with untraceability, flexible anonymity and with low computation will be very useful and very important.

In this paper, we analyse electronic-payment models first, then propose a new off-line electronic cash scheme, in which the anonymity of consumers is scalable and can be done by consumers themselves. Consumers can get the required anonymity without showing their identities to any third party. Furthermore, the new method can prevent from eavesdropping, tampering, impersonation and "perfect crime" effectively. It is more efficient electronic cash scheme by comparing with David Pointcheval [15]. This is truly anonymous for legal consumers and can trace consumers' identities for double spending.

The paper is organized as follows. In the following section, some basic definitions and the simple examples are reviewed. The payment model and the anonymity provider agent are described in section 3. The design of a new off-line electronic cash scheme and its complexity are detailed in section 4 and the security analysis of the scheme is given in section 5. Comparing with David Pointcheval [15] is shown in section 6. An example and how to use the new e-cash for Internet purchases are given in section 7. Conclusions are included in section 8.

2 Some Basic Definitions

2.1 Hash functions

$H(x)$ is a hash function. For a given value W it is computationally hard to find a x such that $H(x) = W$, i.e. collisions are hard to find, where x might be a vector.

Hash function is a major building block for several cryptographic protocols, including pseudorandom generators [1], digital signatures [3], and message authentication.

2.2 DLA and ElGamal encryption system

Discrete Logarithm Assumption (DLA) is an assumption that the discrete logarithm problem is believed to be difficult.

The discrete logarithm problem is as follows: given an element g in a group G of order t, and another element y of G, find x, where $0 < x < t - 1$, such that y is the result of multiplying g with itself x times. In some groups there exist elements that can generate all the elements of G by exponentiation (i.e. applying the group operation repeatedly) with all the integers from 0 to $t-1$. When this occurs, the element is called a generator and the group is called cyclic. Rivest [18] has analyzed the expected time to solve the discrete logarithm problem both in terms of computing power and cost.

For this reason, it has been used for the basis of several public-key cryptosystems, including the famous ElGamal encryption system. ElGamal encryption system [10] is a public key encryption scheme which provides semantic security. Let us briefly recall it.

step 1. The system needs a group G of order q, and a generator g.
The secret key is an element $X \in Z_q = \{0, 1, \ldots, q-1\}$ and
the public key is $Y = g^X$.
step 2. For any message $m \in G$, the ciphertext of m is
$c = (g^r, Y^r m)$, for a random $r \in Z_q - \{0\}$.
step 3. For any ciphertext $c = (a, b)$, the message m can be retrieved by
$m = b/a^X$.

ElGamal encryption scheme

2.3 Undeniable signature scheme and Schnorr signature scheme

The undeniable signature scheme, devised by Chaum and van Antwerpen [7], is a non-self-authenticating signature schemes, where signatures can only be verified with the signer's consent. However, if a signature is only verifiable with the aid of a signer, a dishonest signer may refuse to authenticate a genuine document. Undeniable signatures solve this problem by adding a new component called the disavowal protocol in addition to the normal components of signature and verification.

An undeniable proof scheme consists of the following algorithms:

1. The key generation algorithm K which outputs random pairs of secret and public keys (sk, pk).

2. The proof algorithm $P(sk, m)$ which inputs a message m, returns an "undeniable signature" S on m.

However this proof "S" does not convince anybody by itself. To be convinced of the validity of the pair (m, S), relative to the public key pk, one has to interact with the owner of the secret key sk.

3. The confirmation process confirms (sk, pk, m, S), which is an interactive protocol between the signer and the verifier, where the prover (the signer) tries to convince the validity of the pair (m, S).

4. The disavowal process is an interactive protocol between the signer and the verifier, where the prover (the signer) tries to prove that the pair (m, S) is not valid (i.e. has not been produced by him).

Schnorr proposed an undeniable signature scheme in 1991 [19]. We simply recall it.

The system needs primes p and q such that q is divided by $(p-1)$, i.e.
$q|(p-1)$, $g \in Z_p$ with order q, i.e. $g^q = 1(modp)$, $g \neq 1$. A consumer
generates by himself a private key s which is a random number in Z_q.
The corresponding public key v is the number $v = g^{-s}(modp)$.

To sign message m with the private key s the consumer performs the
following steps:
1. Computes $x = g^r(modp)$, where $r \in Z_q$ is a random number.
2. Computes $e = H(x, m)$, where H is a hash function.
3. Computes $y = r + se(modp)$ and output the signature (e, y).

To verify the signature (e, y) for message m with the public key v a verifier
computes $\overline{x} = g^y v^e(modp)$ and checks $e = h(\overline{x}, m)$.

Schnorr signature scheme

There are three exponentiations in the Schnorr signature scheme, one is from the signer and other two from the verifier.

3 Basic model and new payment model

We will show the basic payment model and then discuss the new payment model in this section.

3.1 Basic payment model

Electronic cash has sparked wide interest among cryptographers ([18, 26, 14], etc.). In its simplest form, an e-cash system consists of three parts (a bank B, a consumer U and a shop S) and three main procedures as shown in Figure 1 (withdrawal, payment and deposit). In a coin's life-cycle, the consumer U first performs an account establishment protocol to open an account with the bank B.

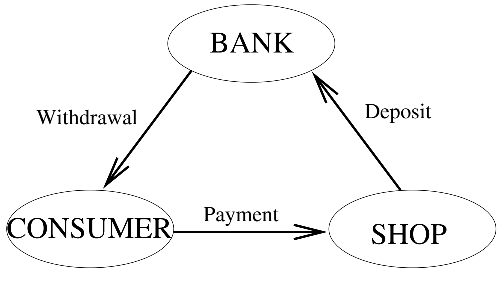

Figure 1. Basic electronic cash system

The consumers and the shops maintain an account with the bank, while

1. U withdraws electronic coins from his account, by performing a withdrawal protocol with the bank B over an authenticated channel.

2. U spends a coin by participating in a payment protocol with a shop S over an anonymous channel, and

3. S performs a deposit protocol with the bank B, to deposit the consumer's coin into his account.

The system is *off-line* if the shop S does not communicate with the bank B during payment. It is *untraceable* if there is no p.p.t. TM (probabilistic polynomial-time Turing Machine) that can identify a coin's origin even if one has all the information of withdrawal, payment and deposit transactions. It is *anonymous* if the bank B, in collaboration with the shop S, cannot trace the coin to the consumer. However, in the absence of tamper-proof hardware, electronic coins can be copied and spent multiple times by the consumer U. This has been traditionally referred to as double-spending. In on-line e-cash, double-spending is prevented by having the bank check if the coin has been deposited before. In off-line e-cash, however, this solution is not possible; instead, as proposed by Chaum, Fiat and Naor [8], the system guarantees that if a coin is double-spent the consumer's identity is revealed with overwhelming probability.

There are also three additional processes such as the bank setup, the shop setup, and the consumer setup (account opening). They describe the system initialization, namely creation and posting of public keys and opening of bank accounts. Although they are certainly parts of a complete system, these are often omitted as their functionalities can be easily inferred from the description of the three main procedures. For clarity we will only describe the bank setup and the consumer setup (because the shop setup is as similar as the consumer setup) for the new scheme in the next section.

Besides the basic participants, a third party named Anonymity Provider (AP) agent will be involved in the scheme. The AP agent will help the consumer to get the required anonymity but will not be involved in the purchase process. The new model can be shown in Figure 2. The AP agent gives a certificate to the consumer who needs a higher level of anonymity.

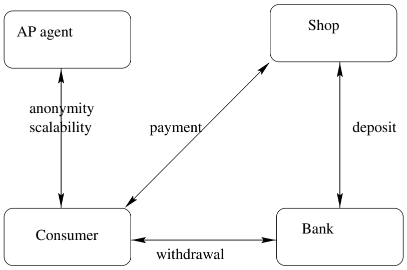

Figure 2. New electronic cash model

3.2 Anonymity Provider Agent

Here we explain what is an AP agent. Assuming a consumer owns a valid coin $c = \varphi(pk_B, pk_u, y)$ with its certificate $Cert_c$, which guarantees correct withdrawal from the bank. Where $\varphi(pk_B, pk_u, y)$ is a function on the public keys of the bank, the user and a variable y, i.e. (pk_B, pk_u, y). Whether a coin is valid or not depends on its certificate. Therefore the bank can revoke the anonymity of the consumer if it finds a consumer who spends a coin twice. After the following processes with the AP agent, the consumer owns a new valid coin, $c' = \varphi(pk_B, pk_u, y + t)$ with its certificate $Cert_{c'}$.

1. The consumer re-encrypts the coin c into $c' = \varphi(pk_B, pk_u, y + t)$.
2. The consumer provides an undeniable signature S, using c as a public key associated with the secret key sk_u of the user, of the equivalence between c and c'. This equivalence is guaranteed by the variable t.
3. The consumer confirms the validity of this signature S to the AP agent.
4. The AP agent certifies the new coin c' and sends $Cert_{c'}$ to the consumer.

Indeed, after steps 2 and 3, the AP is convinced that the conversion has been performed by the owner of the coin c; c' is equivalent to c. The owner of c will not be able to deny S (the relation between c and c'). The AP agent should be an electronic notarized participant in the system. It does not need to know any private information about consumers, only verifies the information of consumers.

3.3 Proof of ownership of a coin

This subsection will show how users prove the ownership of a coin. Let us assume that Y is the public key of the bank, and $I = g^{x_u}$ the identity of a consumer. $H(x, y)$ is a hash function. A coin is the encryption of I: $c = (a = g^r, b = Y^r I^s)$ which is afterwards certified by the bank, where r, s are random numbers. With the certificate of the bank, one knows that the encryption is valid. Therefore, in order to prove his ownership, the consumer has just to convince of his knowledge of (x_u, r, s) such that $b = Y^r I^s$. This can be expressed as follows.

1. Consumers choose random $k \in Z_p$, then compute $t = Y^k g^s (mod\, p)$
and $e = H(m, t)$ where m is a mixed message of c, current time etc,
2. Then compute $u = k - re(mod\, p)$, $v = s - x_u e(mod\, p)$, and
$t_1 = g^{(s-1)x_u e}(mod\, p)$,
3. The signature finally consists of (e, u, v, t_1),
4. In order to verify it, one has just to compute $t' = Y^u g^v b^e$ and check
whether $t' = tt_1$ and $e = H(m, t'/t_1)$.

Proof of validity of a coin $c = Y^r I^s$

We like to note that the message m include the coin c, the certificate $Cert_c$, the current time etc. Due to the current time will be changed when the owner of the shop wants to use it, the coin c can not be used again by the shop.

In the proof process, there are six exponentiations, three are from the consumer and other three from the verifier.

Then, a scrambled coin is simply got by multiplying both parts of the old one by respective bases, g and Y, put at a same random exponent ρ :

$$c' = (a' = g^{\rho} a, b' = Y^{\rho} b) = (g^{r+\rho}, Y^{r+\rho} I^{s}).$$

Then, if the owner of the old coin has certified the message $m' = h^{\rho}$, equivalence of both coins can be proven with the proof of equivalence of three discrete logarithms:

$$log_h m' = log_g(a'/a) = log_Y(b'/b)$$

where h is a public variable.

4 Self-scalable anonymity payment scheme

In this section, we propose an anonymity self-scalable payment scheme. The new payment scheme has two main features, the first is that a consumer can have a higher level of anonymity by himself, the second is that the identity of a consumer can not be traced unless the consumer spends the same coin twice.

Our scheme includes two basic processes in system initialization (bank setup and consumer setup) and three main protocols: a new withdrawal protocol with which U withdraws electronic coins from B while his account is debited, a new payment protocol with which U pays the coin to S, and a new deposit protocol with which S deposits the coin to B and has his account credited. If a consumer wants to get a higher level of anonymity after getting a coin from the bank (withdrawal), s/he can contact the AP agent.

4.1 System Initialization

The bank setup and the consumer setup are described as follows, and the details of the shop setup are omitted (because the shop setup is similar to the consumer setup).

Bank setup: (performed once by B)
Primes p and q are chosen such that $|p-1| = \delta + k$ for a specified constant δ, and $p = \gamma q+1$, for a specified small integer γ. Then a unique subgroup G_q of prime order q of the multiplicative group Z_p and generator g of G_q are defined. Secret key $x_B \in_R Z_q$ for a denomination is created, where $a \in_R A$ means that the element a is selected randomly from the set A with uniform distribution. Hash function H from a family of collision intractable hash function is also defined. B publishes p, q, g, H and its public keys $Y = g^{x_B} (\mathrm{mod}\, p)$.

The secret key x_B is safe under the DLA. The hash function will be used in payment transactions.

Consumer setup : (performed for each consumer U)
The bank B associates the consumer U with $I = g^{x_u}$(mod p) where $x_u \in G_q$ is the secret key of the consumer and is generated by U.

In system initialization, the communication complexity is $O(1)$ for the consumer only sends its account I of length l bits to the bank, and the computation complexity is $O(1)$. It requires only two exponentiations g^{x_B} and g^{x_u}.

After the consumer's account and the shop's account opening, we can describe the new payment scheme.

4.2 New off-line payment scheme

We now describe the new anonymity scalable electronic cash scheme which includes withdrawal, payment and deposit.

Withdrawal: As usual, an anonymous coin is a certified message, which embeds the public key of a consumer. In our scheme, the message is an encryption of this consumer's public key, using the public key Y of the bank.

Instead of using intricate zero-knowledge proofs to convince the bank of the validity of the encryption, the consumer shows some information to the bank including a signature. So the bank certifies the encryption with full confidence.

The consumer $I = g^{x_u}$ constructs a coin $c = (a = g^r, b = Y^r I^s)$ using the public key Y of the bank, where s is a secret key of the coin, which is kept by the consumer and r is a random number in Z_q. She/He also signs c together with the date, using his private key x_u and a Schnorr signature. She/He sends both to the bank together with r, I. Then the bank can check the correct encryption. With the signature of the coin and the date, only the legitimate consumer could have done it. After having modified the consumer's account, the bank sends back a certificate $Cert_c$. The consumer just has to remember $(r, s, Cert_c)$.

Anonymity scalability: The consumer can use the coin now without a higher anonymity since the bank can easily trace any transaction performed through the coin. This is because some information of the consumer such as $I, Cert_c$ has been known by the bank. To solve this problem, an AP agent is established to help the consumer to make a higher level of anonymity: the consumer can derive a new encryption of his identity in an indistinguishable way. However, the consumer will need a new certificate for a new issued ciphertext. The AP agent can provide this new certificate. Before certifying, the consumer requires both the previous coin $(c, Cert_c)$ and the proof of equivalence between the two ciphertexts. Details are described below.

The consumer contacts the AP agent if s/he needs to get a higher level of anonymity. The consumer chooses a random ρ and re-encrypts the coin:

$$c' = (a' = g^{\rho} a, b' = Y^{\rho} b).$$

1. The consumer generates a Schnorr signature S on $m = h^{\rho}$ using the secret key x_u as shown in subsection 2.3.

Because of S, the consumer will not be able to deny his knowledge of ρ later. Furthermore, nobody can impersonate the consumer at this step, since the discrete logarithm x_u of I is required to produce a valid signature. So there is no existential forgery.

2. The consumer also provides a designated -verifier proof of equality of discrete logarithms

$$log_h m = log_g(a'/a) = log_Y(b'/b). \qquad (1)$$

3. The consumer finally sends c, c', S, m to the AP agent.

4. The AP agent checks the certificate $Cert_c$ on c, the validity of the signature S on the message m, then certifies c' and sends back a certificate $Cert_{c'}$ to the consumer.

After these processes the consumer gets a new certified coin $c' = (a' = g^\rho a, b' = Y^\rho b)$ and a new certification $Cert_{c'}$ which is now strongly anonymous from the point of view of the bank. The AP agent has to keep (c, c', m, S) to be able to prove the link between c and c', with the help of the consumer.

In the withdrawal process, the communication complexity is $O(1)$ since the consumer sends c, I and a signature to the bank and the bank returns $Cert_c$ to the consumer, six exponentiations are required in the withdrawal, four are from the consumer and two from the bank. Six exponentiations are required in the scalable anonymity providing process, four are from the consumer and two from the AP agent.

Following the process, the AP agent can also give many smaller new coins for an old one since the amount of new one can be embedded in the certificate $Cert_{c'}$.

Payment: (performed between the consumer and the shop over an anonymous channel)
When a consumer possesses a coin, s/he can simply spend it at shops: proves the knowledge of the secret key (x_u, s) associated with the coin c or c'. This proof is a signature $S = (e, u, v, t_1)$, which has shown in subsection 3.3, of the new certificate $Cert_{c'}$, purchase, date, etc with the secret key (x_u, s) associated to the coin to the receiver (which is later forwarded to the bank). Since the signature S of the message includes the current time which can not be changed and needs the secret key (x_u, s), only the consumer can use the coin. This means the shop can not pay the coin to another shop. This can prevent the shop using the coin sent by the consumer, otherwise, the shop can frame the user.

In payment transactions, the communication complexity is $O(1)$ for the consumer sending c and a signature $S = (e, u, v, t_1)$ to the shop. There are five exponentiations for the signature.

Deposit: (The receiver deposits a coin to a bank)
Since the system is off-line, the shop will send the payment transcript to the bank B later. The transcript consists of the coin c or c' (if the consumer applied a higher level of anonymity), the signature and the date/time of the transaction. The bank will verify the correctness of payment and credit the coin into shop's account.

In the deposit, the communication complexity is $O(1)$ because the shop sends the consumer's response c, and signature $S = (e, u, v, t_1)$ to the bank. The computation complexity is $O(1)$, since it only verifies whether c or c' was used before or not.

Untracebility: The receiver (shop) deposits the coin into its bank's account with a transcript of the payment. If the consumer uses the same coin c twice, then the consumer will be traced: two different receivers will send the same coin c to the bank. The bank can easily search its records to ensure that c has not been used before. If the consumer uses c twice, then the bank has two different signatures. Thus, the bank can isolate the consumer and trace the payment to the consumer's account I.

In the new scheme, the communication complexity is $O(1)$, and required exponentiations are eighteen which is less than that in [15]. So it is quite efficient.

5 Security Analysis

We analyze the security of the system in this section. It includes how the system can preserve the requirements of a secure e-cash system and how to prevent "perfect crime" [20]. The "perfect crime" is a new problem in electronic payement, since users of coin have may been forced by crimers such as killed or kidnapped. Crimers want to use the illegal money from users. Our new payment scheme can stop crimers using the money.

5.1 Payment scheme security

An off-line e-cash scheme is secure [12] if the following requirements are satisfied:

1. *Unreusable*: If any consumer uses the same coin twice, the identity of the consumer can be computed.

2. *Unexpandable*: With any number of the customer's valid withdrawal, payment and deposit protocols, no p.p.t. Turing Machine can compute a legal consumer's identity.

3. *Unforgeable*: With any number of the customer's withdrawal, payment and deposit, no p.p.t. Turing Machine can compute a single valid coin.

4. *Untraceable*: With n withdrawal processes, no p.p.t. Turing Machine can compute $(n + 1)$th distinct and valid coin.

The security in the e-cash scheme is based on the hardness of Discrete Logarithms [27] and hash functions. The system preserves the above four requirements.

Unreusable: The user owns two coins which represent the same money (the old and the new coins), but can exchange or spend both of them. We will see that the identity of users can be found when the old or the new coins are used twice or they are used separately. We analyze what will be happened if users use the higher level anonymous coin (the new coin), and omit the case of users use the old coin twice. This is because these two cases are similar.

When a consumer spends the new coin c' with the new certificate $Cert_{c'}$,s/he hands over the coin together with a signature $S = (e, u, v, t_1)$ to a shop. If the consumer uses a coin twice, then there are two signatures $S_1 = (e_1, u_1, v_1, t_{11})$ and $S_2 = (e_2, u_2, v_2, t_{12})$, where

$$u_1 = k_1 - (r + \rho)e_1(modp), \; v_1 = s - x_u e_1(modp).$$

$$u_2 = k_2 - (r + \rho)e_2(modp), \; v_2 = s - x_u e_2(modp).$$

Then $(v_2 - v_1)/(e_1 - e_2) = x_u$, this is the secret key of the consumer I. This means a coin in the new scheme cannot be reused. If the consumer uses the old and the new coin separately, there are two signatures, $S_1 = (e_1, u_1, v_1, t_{11})$ for the new coin and $S_2 = (e_2, u_2, v_2, t_{12})$ for the old one too, where

$$u_1 = k_1 - (r + \rho)e_1(modp), \; v_1 = s - x_u e_1(modp).$$

$$u_2 = k_2 - re_2(modp), \; v_2 = s - x_u e_2(modp).$$

Then $(v_2 - v_1)/(e_1 - e_2) = x_u$, this is the secret key of the consumer I. Therefore the consumer can not spend them separately. In a word, it is unreusable.

Untraceable: When a consumer constructs a coin, s/he uses the secret keys x_u and s, both are not shown to any other partiess in the purchase process. So no one can trace the consumer from a coin.

Unforgeable: We first discuss whether the bank and the AP agent can forge a valid coin or not. Two requirements are necessary to produce a valid coin, the first is making a encryption $c = (a = g^r, b = Y^r I^s)$ of I, the second is using the secret key x_u of the consumer to sign a Schnorr signature of c together with the current time. The bank can do the first one but can not do the second one since it does not know the secret key x_u. This means the bank can not forge a valid coin. Similarly, the AP agent has no possibility to forge a valid coin. The AP agent knows c, c', S, m, but does not know how to sign the Schnorr signature S of the $m = h^\rho$. This is because the secret key (r, x_u) of the consumer has to be used in the signature S. So the AP agent can not forge a valid coin either. It should be noted that even though both the bank and the AP agent know a valid coin, they can not use it. This is because the signature $S = (e, u, v, t_1) = \Im((r, x_u), m)$ on the message m in the payment process can only be produced by the user. The message m includes the current time, purchase and the coin etc. Therefore the bank, AP agent and shop can not use the coin even they get it. So only the user can use the coin.

As already seen, the secret key x_u of a consumer is never revealed, only used in some signatures. Any consumer is therefore protected against any impersonation, even from a collusion of the bank, the AP agent, and the shop. Only the consumer can construct a valid coin since there is a undeniable signature embedded in the coin. To prevent the bank from framing the consumer as a multiple spender in the scheme, we use digital signature I^s for s which is known only by the consumer. Then the system is unforgeable.

Unexpandable: For a legal consumer and a valid coin, the secret key x_u and the random number s are never shown to others at anytime. Furthermore, usually, the random number s will be changed for different coins. With n withdrawal proceedings, the random number s will be changed n times. Then, no one can compute $(n+1)th$ distinct and valid coins even they see n withdrawal proceedings.

We have seen the system is secure under the definition in [12] and no other parties can frame the user even they do cooperations. Next we will discuss how to prevent the "perfect crime" by the system.

The aim of the crimer in the "perfect crime" is to get money from the bank and use it later. We show the crimer can not use the money even they get it. The user will be found when a crimer forces a user to get the money of the user. The user's identity will be found by the bank, and then the crimer can not withdrawal coins from the bank. The bank can also stop the crimer to use the money of the user even if it has been withdrawaled by the user. This is because the bank can trace coins from the identity of the user and then send a warning message to the AP agent and shops. Either the AP agent or the shops will not accept the coins which can not be used anymore.

6 Comparisons

In this section, we compare the new scheme with the proposed approach in [15]. The computation complexity of the protocols is better than that in [15]. The main processes of David Pointcheval [15] are below.

Registration: The registration of a user is certified by a Certification Authority.

Withdrawal: Users construct coin using the public key of a Revocation Centre. So the Revocation Centre can trace users at any time even users have not spent coin twice.

Self-Scrambling Anonymizer: Users contact another third party (likes AP) to certify his message, and the latter provides a new certified coin to users after verifying the message.

Spending: Users send a coin and a signature of the purchase, date etc, with the secret key associated to the coin to the payee.

Revocation: The identity of users can be traced by the Revocation Centre at anytime. The Revocation Centre has to decrypt the coin. Therefore, the identity of users can be known even the coin does not be spent twice.

The phase of the Self-scrambling anonymizer in [15] requires 10 exponentiations from the user point of view and 11 from the Anonymity Provider's point of view. In the protocol, the phase of the scalable anonymity required four exponentiations from the user point of view and two from the Anonymity Provider's point of view. Moreover, only the double spending user will be found by a simple linear computation, do not need description the coin. These show the new protocol is more efficient.

7 An example and implementations

In this section, we will give a simple example and analyse two different purchase procedures. We will show how to use the new e-cash for Internet purchases and how to get some smaller coins from the AP agent. As a result, we will see the efficiency of the payment protocol.

7.1 An example

This example will show the main steps in the e-cash scheme. We omit the details of two undeniable signatures in withdrawal and scalable anonymity process, because they are only used for verifying the user. For simplicity, module 47 which has been used in the computation below is omitted in the expression.

Bank setup

Suppose $(p, q, \gamma, k) = (47, 23, 2, 4)$, then $G_q = \{0, 1, 2, ..., 22\}$ is a subgroup of order 23. $g = 3$ is a generator of G_q. The bank's secret key $x_B = 4$ and hash function $H(x, y) = 3^x * 5^y$. The bank publishes $H(x, y)$ and $\{p, q, g\} = \{47, 23, 3\}$. The public key of the bank is $Y = g^{x_B} = 34$.

User setup

We assume the secret of a user is $x_u = 7$ and the user sends $I = g^{x_u} = 32$ to the bank. After checking some things like social security card or drive license, the bank authorizes the user (consumer) with I.

After the bank setup and the user setup, the user can do purchase.

Withdrawal

The user chooses $(r, s) = (2, 3)$ and computes $c = (g^r, Y^r I^s) = (9, 2)$, then signs a Schnorr signature S for the message $m = (c, t)$, where t is the current time. The user sends $c = (9, 2)$ and S to the bank, the latter sends back a certificate $Cert_c$.

The user contacts the AP agent if s/he needs a high level of anonymity, or uses the coin in a shop directly (See Payment). The user and the AP agent follow the processes below. We suppose $h = 37$ is a public number.

Anonymity scalability

The user re-encrypts the coin c, chooses $\rho = 4$ and computes $c' = (a' = g^\rho a, b' = Y^\rho b) = (24, 14)$ and signs a Schnorr signature S on $m = h^\rho = 36$. Finally, the user sends (c, c', S, m) to the AP agent. The latter verifies the Schnorr signature S and the equation (1), and sends a certificate $Cert_{c'}$ to the user if they are correct.

Since the new coin $c' = (24, 14)$ and its certificate $Cert_{c'}$ has no relationship with the bank, the user has a high anonymity.

Payment

The user signs a signature $S = (e, u, v, t_1)$ of a message m which includes $c', Cert_{c'}$ and purchase time etc to prove the ownership of the new coin. For convenience, we assume $m = 11$. The user chooses $k = 5$ then computes $t = Y^k g^s = 19, e = H(m, t) = 40, u = 18, v = 5, t_1 = 28$.

The shop computes $t' = 15$ who is convinced that the user is the owner of the coin if the equation of $t' = tt_1$ and the signature S are successful. She/He does not know who is the user.

Deposit

The bank will put the money into the shop's account when the checking of the coin $C' = (24, 14)$ and the signature $S = (e, u, v, t_1) = (40, 18, 5, 28)$ are correct. The shop can also see that the money in his account is added.

7.2 Purchase procedures

Purchase procedure 1

In purchase procedure 1 a consumer decides how much money should be paid to the shop, withdraws the money from the bank, and pays it to the shop.

1. *Consumer to shop*: The consumer wants to buy some goods in a shop, so contacts the shop for the price.

2. *Consumer to bank*: The consumer gets the money from the bank, the amount being embedded in the signature.

3. *Anonymity scalability*: If the consumer wants to maintain higher level of anonymity, s/he can ask the AP agent to certify a new coin which can be then used in the shop.

4. *Consumer to shop*: The consumer proves to the shop that s/he is the owner of the money, and pays it to the shop. Then the shop sends the goods to the consumer.

5. *Shop to bank*: The shop deposits the e-cash in the bank. The bank checks the validation and that there is no double spending of the coin. The bank transfers the money to the shop's account.

Purchase procedure 2

In purchase procedure 2 is that: the consumer does not have to ask the bank to send money since the consumer already has enough e-cash in his "wallet". All s/he needs to do is to get some smaller e-cash from the AP agent to pay the shop.

There are 4 steps in the purchase procedure 2. They are: (1) *consumer to shop*; (2) *consumer to AP agent*; (3) *consumer to shop* again and (4) *shop to bank*. Step 2, *consumer to AP agent* is different from the step 3 in procedure 1 and another three steps are similar to that in procedure 1. Therefore we will focus only on step 2 *consumer to AP agent*. It should be noted that electronic-cash is a digital message and a certification. We say that the AP agent can provide certificates of coins then provide a service in changing small coin.

Consumer to AP agent: The consumer advises the AP agent of the amount of money to pay the shop from his wallet. She/He can ask the AP agent to make some smaller coins. By doing this, the consumer can also get a higher level of anonymity. After checking the old money sent by the consumer, the AP agent creates some new coins of an equivalent value to the original coin. One of these new coins can be used in the shop.

We have already seen that the consumer can keep money in his wallet or get money from the bank. In both purchase procedures 1 and 2 most computations are done by the consumers, so the system is very convenient for Internet purchases.

8 Conclusions

In this paper, a new electronic cash scheme is designed to provide different degree of anonymity for consumers. Consumers can decide the levels of anonymity. They can have a low level of anonymity if they want to spend coins directly after withdrawing them from the bank. Consumers can acheive a higher level of anonymity through the AP agent without revealing their private information and are more secure in relation to the bank because the new certificate of a coin comes from the AP agent who is not involved in the payment process. This system does not need a trusted party to manage consumers' identities. In this new model, we have shown how to derive an efficient and untraceable cash scheme based on the variation of coins. It is an off-line scheme with low communication and computation. With its scalable anonymity, the new payment protocol can effectively prevent eavesdropping, tampering, impersonation and "perfect crime". Finally, we have compared the new payment protocol with another one to show its efficiency.

References

[1] Bellare M., Goldreich O., and Krawczyk H. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In *Advances in Cryptology - Crypto 99*, volume 1666 of *Lectures Notes in Computer Science*. Springer-Verlag, 1999.

[2] Boyko V., Peinado M., and Venkatesan R. Speeding up discrete log and factoring based schemes via precomputations. In *Advances in Cryptology - Eurocrypt'98*, volume 1807 of *Lectures Notes in Computer Science*. Springer-Verlag, 1998.

[3] Canetti R., Goldreich O., and Halevi S. The random oracle methodology. In *Proceedings of the 30th ACM STOC '98*, pages 209–218. IEEE, 1998.

[4] Chan A., Frankel Y., and Tsiounis Y. An efficient off-line electronic cash scheme as secure as RSA. Research report nu-ccs-96-03, Northeastern University, Boston, Massachussets, 1995.

[5] Chaum D. Blind signature for untraceable payments. In *Advances in Cryptology - Crypto 82*, pages 199–203. Plenum Press N.Y., 1983.

[6] Chaum D., editor. *An introduction to e–cash*. DigiCash, http://www.digicash.com, 1995.

[7] Chaum D. and Van antwerpen H. Undeniable signatures. In *Advances in Cryptology–Crypto89*, volume 435 of *Lectures Notes in Computer Science*, pages 212–216. Springer-Verlag, 1990.

[8] Chaum D., Fiat A., and Naor M. Untraceable electronic cash. In *Advances in Cryptology - Crypto 88*, volume 403 of *Lectures Notes in Computer Science*, pages 319–327. Springer-Verlag, 1990.

[9] Cox B., Tygar J.D., Sirbu M. Netbill security and transaction protocol. In *The first USENIX Workshop on Electronic Commerce*, New York, 1995.

[10] EIGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, 1985.

[11] Frankel Y., Yiannis T., and Yung M. Indirect discourse proofs: achieving fair off-line electronic cash. In *Advances in cryptology–Asiacrypt'96*, volume 1163 of *Lectures Notes in Computer Science*, pages 286–300. Springer-Verlag, 1996.

[12] Franklin M., Yung M. Secure and efficient off-line digital money. In *Proceedings of the Twentieth International Colloquium on Automata, Languages and Programming*, volume 700 of *Lectures Notes in Computer Science*, pages 265–276. Springer-Verlag, 1993.

[13] MastercardVisa, editor. *SET 1.0 - Secure electronic transaction specification*. http://www.mastercard.com/set.html, 1997.

[14] Okamoto T. An efficient divisible electronic cash scheme. In *Advances in Cryptology–Crypto'95*, volume 963 of *Lectures Notes in Computer Science*, pages 438–451. Springer-Verlag, 1995.

[15] Pointcheval D. Self-scrambling anonymizers. In *Proceedings of Financial Cryptography*, Anguilla, British West Indies, 2000. Springer-Verlag.

[16] Poutanen T., Hinton H. and Stumm M. Netcents: A lightweight protocol for secure micropayments. In *The 3rd USENIX Workshop on Electronic Commerce*, Boston, Massachusetts, August, 1998.

[17] Rivest R. L., Shamir A., and Adleman L. M. A method for obtaining digital signatures and public-Key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[18] Rivest R. T. The MD5 message digest algorithm. *Internet RFC 1321*, April 1992.

[19] Schnorr C. P. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[20] Solms S.von and Naccache D. On blind signatures and perfect crimes. *Computers and Security*, 11:581–583, 1992.

[21] Wang H., Cao J., Zhang Y. A consumer anonymity scalable payment scheme with role based access control. In *2nd International Conference on Web Information Systems Engineering*, pages 384–389, Kyoto, Japan, Dec. 3-6, 2001.

[22] Wang H., Cao J., Zhang Y. Ticket-based service access scheme for mobile users. In *Twenty-Fifth Australasian Computer Science Conference (ACSC2002)*, Monash University, Melbourne, Victoria, Australia, Jan. 28-Feb. 2.

[23] Wang H., Zhang Y. A protocol for untraceable electronic cash. In Hongjun Lu and Aoying Zhou, editor, *Proceedings of the First International Conference on Web-Age Information Management*, volume 1846 of *Lectures Notes in Computer Science*, pages 189–197, Shanghai, China, 2000. Springer-Verlag.

[24] Wang H., Zhang Y. Untraceable off-line electronic cash flow in e-commerce. In *Proceedings of the 24th Australian Computer Science Conference ACSC2001*, pages 191–198, GoldCoast, Australia, 2001. IEEE computer society.

[25] Yacobi Y. Efficient electronic money. In *Advances in Cryptology–Asiacrypt'94*, volume 917 of *Lectures Notes in Computer Science*, pages 153–163. Springer-Verlag, 1995.

[26] Yiannis T. Fair off-line cash made easy. In *Advances in Cryptology–Asiacrypt'98*, volume 1346 of *Lectures Notes in Computer Science*, pages 240–252. Springer-Verlag, 1998.

[27] Yiannis T., Yung M. On the security of ElGamal-based encryption. In *International Workshop on Practice and Theory in Public Key Cryptography (PKC '98)*, volume 1346 of *Lectures Notes in Computer Science*, Yokohama, Japan, 1998. Springer-Verlag.