An electronic cash scheme and its management

Hua Wang (*) Yanchun Zhang (**) Jinli Cao (***)

(*)Department of Maths & Computing, University of Southern Queensland
Toowoomba QLD 4350 Australia
Email: wang@usq.edu.au

(**)School of Computer Science and Mathematics
Victoria University of Technology, Melbourne City, MC8001, Australia.
Email: yzhang@csm.vu.edu.au

(***)Department of Computer Science & Computer Engineering
La Trobe University, Melbourne, VIC 3086, Australia
Email: jinli@cs.latrobe.edu.au

Abstract:

A secure electronic cash scheme and its role-based access control (RBAC) management are proposed in this paper. The scheme uses electronic cash for payment transactions. In this protocol, from the viewpoint of banks, consumers can improve anonymity if they are worried about disclosure of their identities. A new role called anonymity provider agent (AP) provides a high anonymous certificate for consumers during a payment processing. The role AP certifies re-encrypted data after verifying the validity of the content from consumers, but with no private information of the consumers required. Each consumer can get a required anonymity level through this new method, depending on the available time, computation and cost. Consumers can also use different banks (cross-payment) and anonymity provider agents (multiple APs) without double spending problems.

RBAC is widely used in system management and products since its advantages such as reducing administration cost and complexity. However, conflicting problems may arise between roles when RBAC is applied for system management. For example, mutually exclusive roles may be granted to a user with RBAC and the user may have or derive a high level of authority. To solve these problems, we analyze the duty separation constraints of the roles and role hierarchies in the scheme, then discuss how granting a role to a user.

Key words: Payment scheme, Electronic cash, RBAC, AP agent, Authorization, Signature.

1. INTRODUCTION

E-commerce is the practice of buying and selling products and services over the Internet, utilizing technologies such as the Web, electronic data interchange, electronic fund transfers. While E-commerce brings convenience for both consumers and vendors, many consumers have concerns about security of 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. The issues on ensuring secure transactions for electronic commerce have attracted many researchers.

There are a number of proposals for electronic cash systems [6, 10, 17]. All of them lack flexibility in anonymity. For instance, David Chaum [6] first proposed an on-line payment system that guarantees receiving valid coins. This system provides some anonymity against a collaboration between shops and banks. However, consumers have no flexible anonymity and banks have to keep a very big database for consumers and coins. Systems mentioned above are on-line payment systems.

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. Digicash [7] 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 even be set up for a 1,000,000 consumer bank. However, 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 [1].

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, may suffer from the potential 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 [9]. The security of their scheme relied on some restricted assumptions such as requiring a function, which is similar to random oracle and has to map 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 lay the foundation for more secure and efficient schemes. In 1995, Chan, Frankel and Tsiounis [5] presented a provable secure off-line e-cash scheme that relied only on the security of RSA [20]. This scheme extended the work of Franklin and Yung [15] 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. In 2000, David Pointcheval [19] 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.

Recently, role-based access control (RBAC) has been widely used in system management and operating system products. RBAC involves individual users being associated with roles as well as roles being associated with permissions (each permission is a pair of objects and operations). As such, a role is used to associate users and permissions. A user in this model is a human being (e.g. a staff member in a bank). A role is a job function or job title within an organization associated with authority and responsibility (e.g. role BANK manages money for consumers). Permission is an approval of a particular operation to be performed on one or more objects. There are many relationships between users and roles, and between roles and permissions as shown in Figure 1. Assigning people to tasks is a normal managerial function. The assignments of users to roles is a natural part of assigning users to tasks. Hence, user-role assignment is a basic issue of RBAC.

As mentioned above, the on-line e-cash payments need more computing resources. The previous designed off-line schemes do not provide efficient anonymity for consumers. In this paper, we present a new electronic-payment model first, and then propose a flexible payment scheme, in which the anonymity of consumers is scalable and can be done by consumers themselves. Consumers can get a required anonymity without showing their identities to any third party. Furthermore, to reduce administration cost and complexity and to improve the security of the scheme, the basic issue of RBAC for the new payment scheme is discussed.

The paper is organized as follows. In the first two sections, some basic definitions and simple examples are reviewed. The payment model and the anonymity provider agent are described in section 3. An off-line electronic cash scheme, its security analysis including how to prevent double-spending and an example of the scheme are shown in section 4. The relation of roles and granting model are discussed in section 5. Conclusions are included in section 6.

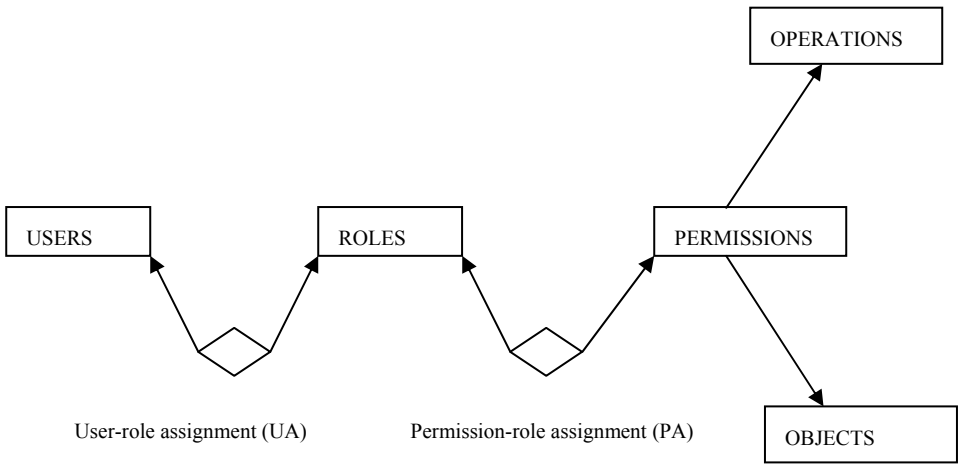

Figure 1: RBAC relationship

2. SOME BASIC DEFINITIONS

Hash functions: *H(x)* is a hash function. For a given value *W* it is computationally hard to find an *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 pseudo-random generators [3], digital signatures [4], and message authentication.

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 *q*, and another element *y* of *G*, find *x*, where $0 < x < q - 1$, such that *y* is the result of multiplying *g* with itself *x* times, i.e. $y = g^x$. 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 $q - 1$. When this occurs, the element is called a generator and the group is called cyclic. Rivest [21] has analyzed the expected time to solve the discrete logarithm problem both in terms of computing power and cost and shown that it is computational hard to get *x* from *y*.

For this reason, it has been used for the basis of several public-key cryptosystems, including the famous ElGamal encryption system. The ElGamal encryption system [11] is a public key encryption scheme which provides semantic security. Let us briefly recall it.

Table 1. ElGamal encryption scheme

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

Table1 indicates that the message m can be obtained only by the person who has the secret key X.

Undeniable signature scheme and Schnorr signature scheme: he undeniable signature scheme, devised by Chaum and Antwerpen [8], is a non-self-authenticating signature schemes, where signatures can only be verified with the signer's consent. Schnorr, for example, proposed an undeniable signature scheme in 1991 [25]. It is simply recalled in Table 2. The signer is not able to deny the signature $Sch = (e, y)$ because the signature can be produced by the owner of the secret key sk only.

Table 2. Schnorr signature scheme

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. $g^q = 1 \pmod p$, g does not equal 1. A consumer generates by himself a private key sk which is a random number in Z_q. The corresponding public key pk is the number $pk=g^{sk} \pmod p$.
To sign message m with the private key sk the consumer performs the following steps: 1. Computes $x=g^r \pmod p$, where r in Z_q is a random number. 2. Computes $e=H(x, m)$, where H is a hash function. 3. Computes $y=r+sk*e \pmod q$ and output the signature $Sch = (e,y)$.
To verify the signature $Sch = (e,y)$ for message m with the public key pk a verifier computes $x^+ = g^y pk^e \pmod p$ and checks $e=h(x^+, m)$.

Role based access control: RBAC model supports the specification of:

a. User/role associations; the constraints specifying user authorizations to perform roles,

b. Role hierarchies; the constraints specifying which role may inherit all of the permissions of another role,

c. Duty separation constraints; these are role/role associations indicating conflict of interest:

c1. Static separated duty (SSD); a constraint specifying that a user cannot be authorized for two different roles,

c2. Dynamic separated duty (DSD); a constraint specifying that a user can be authorized for two different roles but cannot act simultaneously in both,

d. Cardinality; the maximum number of users allowed, i.e. how many users can be authorized for

any particular role (role cardinality), e.g., only one manager.

A comprehensive administrative model for RBAC must account for all four issues mentioned above, among others. However, user-role assignment is a particularly critical administrative activity. Because when administrators grant memberships in roles to users and revoke memberships from users, the process of granting depends on duty separation constraints and the revocation depends on role hierarchies. Therefore, this paper focuses on user-role assignments for the flexible payment scheme.

3. NEW PAYMENT MODEL

In the simplest form, an e-cash system consists of three parts (a bank, a consumer and a shop) and three main procedures as shown in Figure 2 (Withdrawal, Payment and Deposit). In a coin's life-cycle, the consumer first performs an account establishment protocol to open an account with the bank.

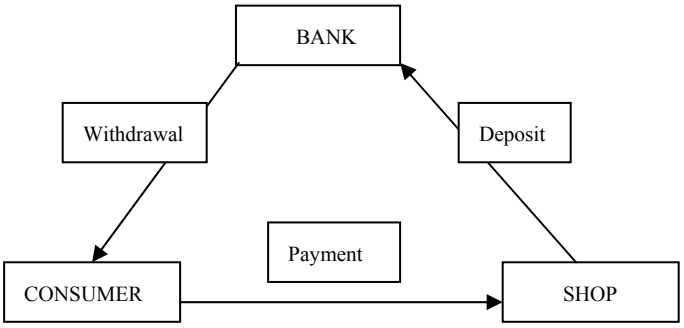

Figure 2: Basic processes of an electronic cash system

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

1. A consumer withdraws electronic coins from his/her account, by performing a withdrawal protocol with the bank over an authenticated channel.
2. The consumer spends a coin by participating in a payment protocol with a shop over an anonymous channel.
3. The shop performs a deposit protocol with the bank, to deposit the consumer's coin into its account.

The system is *off-line* if the shop does not communicate with the bank 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, in collaboration with the shop, 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. 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 [9], 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: the bank setup, the shop setup, and the consumer setup (account opening). They describe the system initialization, namely the 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.

Besides the basic participants, a third party named anonymity provider (AP) agent is 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 model is shown in Figure 3. The AP agent gives a certificate to the consumer when s/he needs a high level of anonymity.

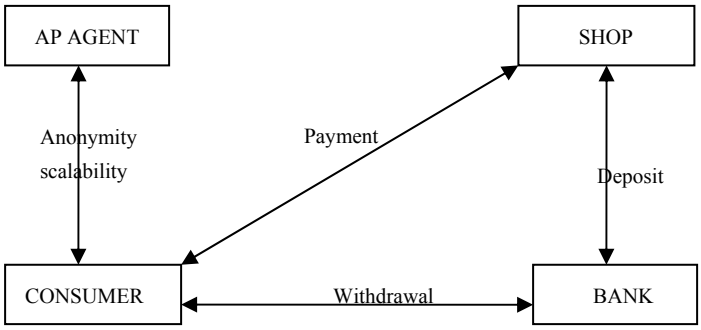

Figure 3. Electronic cash model

The AP agent cannot merge with the bank, otherwise the consumer is not able to get a coin with a high level of anonymity. We will discuss this case in detail in section 5.

3.1 Anonymity Provider Agent

Here we explain what the AP agent is. Assume a consumer owns a valid coin c with its certificate $Cert_c$, which guarantees correct withdrawal from the bank. Whether a coin is valid or not depends on its certificate. After the following processes with the AP agent, the consumer owns a new valid coin c' with its certificate $Cert_{c'}$.

1. The consumer re-encrypts the coin c into c'.
2 . The consumer provides an undeniable signature *Sch*.

3. The consumer confirms the validity of this signature *Sch* 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 *Sch* (the relation between c and c'). The AP agent only verifies the information of consumers. It does not need to know any private information about consumers.

3.2 Proof of ownership of a coin

Let us assume that Y is the public key of the bank, x_C is a secret key of a consumer, and $I=g^{xc}$ the identity of a consumer. $H(x, y)$ is a hash function. A coin is an encrypted message 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_C, r, s) such that $b=Y^r I^s$. This can be expressed as follows.

Table 3. Proof of validity of a coin

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

Then, a scrambled coin is got by multiplying both parts of the old one by respective bases, g and Y, both to the 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:

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

Where h is a public variable.

4. A FLEXIBLE ANONYMITY PAYMENT SCHEME

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

The scheme includes two basic processes in system initialization (bank setup and consumer setup) and three main protocols: a withdrawal protocol with which a consumer withdraws electronic coins from a bank while his account is debited; a payment protocol with which the consumer pays the coin to a shop; and a deposit protocol with which the shop deposits the coin in the bank and has its account credited. If a consumer wants to get a high 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. The details of the shop setup are omitted because the shop setup is similar to the consumer setup.

Bank setup: (performed once by banks) 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 functions is also defined. The bank publishes p, q, g, H and its public key $Y = g^{xB} \pmod p$.

The secret key x_B is safe under the Discrete Logarithm Assumption (DLA) [11]. The hash function will be used in payment transactions.

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

After the consumer's account and the shop's account are opened, the payment scheme can be described.

4.2 New off-line payment scheme

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

Withdrawal: (Consumers withdraw coins from the bank)

Usually, 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.

The consumer $I = g^{xC}$ constructs a coin $c=(a=g^{r}, b=Y^{r} I^{s})$ using the public key Y of the bank, where s is the secret key of the coin, which is kept by the consumer and r is a random number in Z_q. Using the private key x_C,, the consumer signs a Schnorr signature *Sch1* on the message of c together with the date etc. She/He sends *(c, Sch1)* 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 can use the coin now without a high anonymity since the bank can easily trace any transaction performed through the coin. This is because information of the consumer such as *I,* $Cert_c$ are known by the bank. To solve this problem, a member in the AP agent is established to help the consumer to achieve a high 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 cipher text. 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 cipher texts. Details are described below.

Anonymity scalability: (Performed between consumers and the AP agent)

The consumer contacts the AP agent if s/he needs to get a high 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 *Sch2* on $m' = h^{\rho}$ using the secret key x_C. Because of *Sch2*, 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_C 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

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

3. The consumer finally sends *c, c', Sch2, m* to the AP agent.

4. The AP agent checks the certificate $Cert_c$ on *c*, the validity of the signature *Sch2* 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.

Payment: (performed between the consumer)

When a consumer possesses a coin, s/he can simply spend it at shops by proving knowledge of the secret key (x_C , s) associated with the coin *c* or *c'*. This proof is a signature $Sig = (e, u, v, t_1)$ of the new certificate $Cert_{C'}$, purchase, date, etc with the secret key (x_C , s) associating the coin to the receiver (which is later forwarded to the bank).

Deposit: (The receiver deposits a coin to a bank)

The shop will send the payment transcript to the bank. The transcript consists of the coin *c* or *c'* (if the consumer applies 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 the shop's account.

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 is traced:

two different receivers 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*.

Remark: There are no limits with shops and banks. It means that consumers can use the coin At different shops as well as receivers can deposit coin to different banks. This is because shops and banks in processes of the payment and deposit are verifiers only.

4.3 Security Analysis

An off-line e-cash scheme is secure [15] 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 *Untraceable*: With *n* withdrawal processes, no p.p.t. (Probabilistic polynomial time) Turing Machine can compute *(n+1)*th distinct and valid coin.

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

4 *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.

The security in the e-cash scheme is based on the hardness of Discrete Logarithms [28] and hash functions. The system preserves the above four requirements.

Unreusable: When a consumer spends a coin, s/he hands over the coin together with a signature $S=(e, u, v, t_1)$ to a shop. The consumer has the old coin $c=(a=g^r, b=Y^r I^s)$ and the new coin $c' = (a' = g^\rho a, b' = Y^\rho b)$. The secret key of the consumer will be found when the old coin c or the new coin c' is used twice (or the old coin and the new coin are both used). If the consumer uses a coin c twice, then we have two signatures $S_1 =(e_1, u_1, v_1, t_{11})$ and $S_2 =(e_2, u_2, v_2, t_{12})$, where

$$u_1 = k_1 - r e_1 \pmod q, \quad v_1 = s - x_C e_1 \pmod q$$

$$u_2 = k_2 - r e_2 \pmod q, \quad v_2 = s - x_C e_2 \pmod q$$

Then $(v_2 - v_1)/(e_1 - e_2) = x_C$, this is the secret key of the consumer I. The same result will be obtained if the new coin c' is used twice or the old coin and the new coin are both used. This means there is no double-spending.

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

Unforgeable: We first discuss whether the bank and the AP agent can forge a valid coin or not. To produce a valid coin, the first requirement is making a encryption $c=(a=g^r, b=Y^r I^s)$ of I. The second requirement is using the secret key x_C of the consumer to sign a Schnorr signature for c together with the current time. The bank can do the first step but cannot do the second step since it does not know the secret key x_C . This means the bank cannot forge a valid coin. Similarly, the AP agent cannot forge a valid coin either. It should be noted that even though both the bank and the AP agent know a valid coin, they still couldn't use it. This is because there is a signature in payment process, which can only be done by the consumer.

As already seen, the secret key x_C of a consumer is never revealed; only used in some signatures. A 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_C 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 the $(n+1)$th distinct and valid coin even if they see n proceeding withdrawals.

4.4 An example

We give a simple example of the flexible payment scheme that will show the main steps in the processes. We omit the details of two undeniable signatures in the withdrawal and in the scalable anonymity process, because they are only used for verifying the consumer.

Bank setup

Suppose (p, q, γ, 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 $Y=g^{xB} \pmod p = 34$.

Consumer setup

We assume the secret of a consumer is x_C = 7 and the consumer sends $I = g^{xC} \pmod p = 32$ to the bank. After checking some identification like social security card or driver's license, the bank authorizes the consumer with I.

After the bank setup and the consumer setup, the consumer can purchase.

Withdrawal

The consumer chooses *(r, s)* = (2,3) and computes c = (9, 2) , then signs a Schnorr signature *Sch1* for the message *(c, t)*, where *t* is the current time. The consumer sends *c* = (9, 2) and *Sch1* to the bank, the latter sends back a certificate $Cert_C$.

Data *c* and $Cert_C$ are known by the bank. Therefore the consumer can be traced by the bank if s/he uses the coin directly. S/He contacts the AP agent to achieve a high level of anonymity. The consumer and the AP agent follow the processes below. We suppose *h* = 37 is a public number.

Anonymity scalability

The consumer re-encrypts the coin *c*, chooses $\rho = 4$ and computes $c' = (24, 14)$ and signs a Schnorr signature *Sch2* on *m1* = 36. Finally, the consumer sends *(c, c', Sch2, m1)* to the AP agent. The latter verifies the Schnorr signature *Sch2* and the equation (1), and sends a certificate $Cert_{C'}$ to the consumer if they are correct.

Since the new coin $c' = (24, 14)$ and its certificate $Cert_{C'}$ have no relationship with the bank, the consumer gets a level high of anonymity.

Payment

The consumer signs a signature $S = (e, u, v, t_1)$ of a message *m2* which includes *c', Cert* $_{C'}$ and purchase time etc to prove the ownership of the new coin. For convenience, we assume *m2* = 11. The consumer chooses *k* = 5 then computes $t = Y^k g^s \pmod p = 19$, $e = H(m2, t) \pmod p = 40$, $u = 18$, $v = 5$, $t_1 = 28$.

The shop who is convinced that the consumer is the owner of the coin computes $t' = 15$ if the equation of $t' = t\, t_1$ and the signature *S* are successful. She/He does not know who the consumer is.

Deposit

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

There are four roles in the scheme, BANK, SHOP, CONSUMER and AP that are acted by a bank, a shop, a consumer and an AP agent respectively. The coin c and its certificate $Cert_C$ are known by the role BANK while the new coin c' and its certificate $Cert_{C'}$ are known by the role AP. BANK will get c' and $Cert_{C'}$ if merging the AP agent with the bank, and then the consumer cannot get a high level of anonymity. Therefore, BANK and AP are mutually exclusive roles and they cannot assign to an individual. It shows that user-role assignment is an important issue for the flexible payment scheme.

5. USER-ROLE ASSIGNMENT

Sandhu and Bhamidipati developed a model called URA97 in which RBAC is used to manage user-role assignment [24]. It does not discuss user-role assignments for electronic commerce. We will analyze user-role assignment for the new scheme in this section. Before analyzing user-role assignment of the new scheme, we need to consider the relationships of roles.

5.1 Duty separation constraints

There are two types in duty separation constraints. One is SSD and another one is DSD. In Figure 4, for example, since all staff in the AP agent, the bank and the shop are employees, their corresponding roles inherit the role EMPLOYEE. Role AP, SHOP and BANK have DSD relationships with role CONSUMER. This indicates that an individual consumer cannot act the roles of AP, SHOP or BANK simultaneously. The staff in these three participants have to first log out if they want to register as consumers. For example, a consumer, who is a staff member of the AP agent and is able to act the role AP, can ask the AP agent to help him to get a coin with a high level anonymity. But as a consumer, s/he cannot give herself/himself a new certificate $Cert_{C'}$ of a coin

when s/he works for the AP agent. Another staff member of the AP agent should do the job for this person.

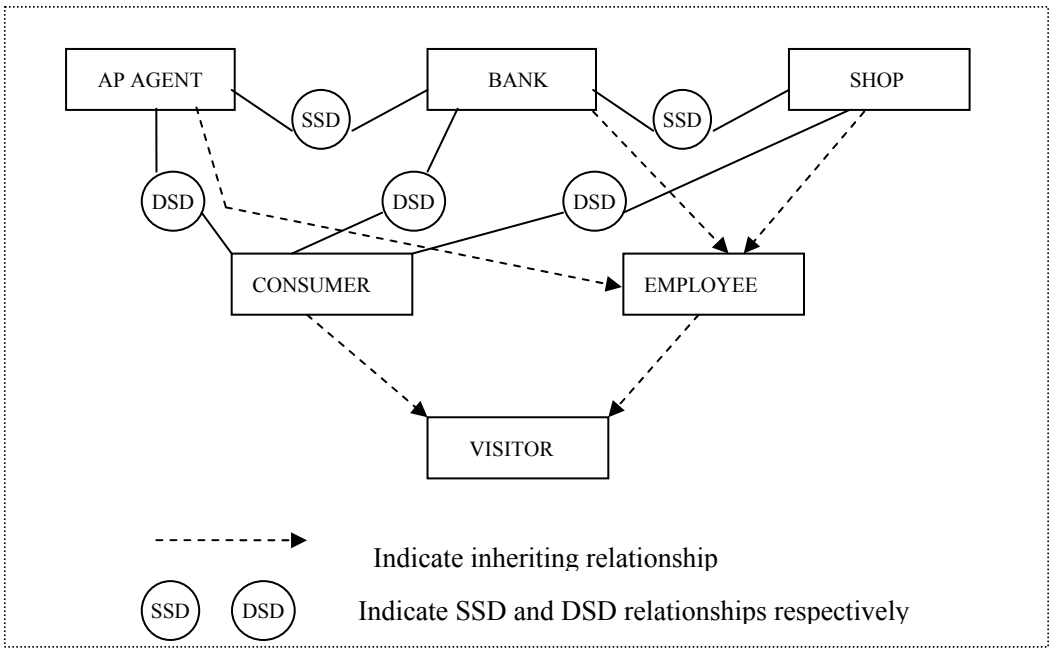

Figure 4: The relationships of the roles in the scheme

AP has an SSD relationship with BANK. This is because the duty of AP is to help a consumer to get a coin with a high level of anonymity. BANK knows the old coin $c = (g^{r} , Y^{r} I^{s})$ and its certificate $Cert_{C}$. AP sends the new certificate $Cert_{C'}$ of the new coin $c' = (g^{(r+\rho)} , Y^{(r+\rho)} I^{s})$ to the consumer. BANK will know the new certificate $Cert_{C'}$ and new coin c' if the same staff member from the AP agent and the bank processed the coin for the consumer. If this occurs, the consumer cannot have a coin with the required anonymity because BANK has known the new coin. SHOP also has an SSD relationship with BANK since BANK verifies the payment as well as depositing the coin to the shop's account. The SSD relationship is also a conflict of interest relationship like the DSD relationship but much stronger. If two roles have a SSD relationship, then they may not even be

authorized to the same individual. Thus, the role AP, BANK, and SHOP may never be authorized to the same individual.

5.2 Granting models

RBAC administration encompasses the issues of assigning users to roles, assigning permissions to roles, and assigning roles to roles to define a role hierarchy. These activities are all required to bring users and permissions together. In many cases, they are best done by different administrators. To analyze granting and revocation models, we add a manager role (M1) etc in an AP agent, a manager role (M2) etc in a bank, a manager role (M3) etc in a shop and some administrative roles Senior Officer(SSO) etc in the system as shown in Figure 5 and Figure 6. A hierarchy of roles and a hierarchy of administrative roles are also shown in these two Figures. Senior roles are shown towards the top of the hierarchies and junior are to the bottom. Senior roles inherit permissions from junior roles. The roles in Figure 5can be granted and revoked by the administrative roles in Figure 6.

Let $x > y$ denote x is senior to y with obvious extension to $x >= y$. The notion of a prerequisite condition is a key part in the processes of user-role assignment [24].

Prerequisite condition is an expression using Boolean operators $\wedge$ and $\vee$ on terms of the form x and $\overline{x}$ and where x is a role and $\wedge$ means ``and", $\vee$ means ``or". A prerequisite condition is evaluated for a user u by interpreting x to be true if $\exists x' >= x, (u, x') \in$ UA and $\overline{x}$ to be true if $\forall$ x' >= x), (u, x') $\notin$ UA, where UA is a set of user-role assignments.

For a given set of roles R let CR denote all possible prerequisite conditions that can be formed using the roles in R. Not every administrator can assign a role to a user. The relation of **can-assign** $\subseteq AR \times CR \times 2^{R}$ provides what roles an administrator can assign with prerequisite conditions, where AR is a set of administrative roles.

User-role assignment (UA) is authorized by can-assign relation. Table 4 shows the can-assign relations with the prerequisite conditions in the new scheme. To identify a role range within the role hierarchy of Figure 5, we use the familiar closed and open interval notation.

$[x, y] = \{ r \in R \mid r >= x \wedge y >= r \}$, $(x, y] = \{ r \in R \mid r > x \wedge y >= r \}$

$[x, y) = \{ r \in R \mid r >= x \wedge y > r \}$, $(x, y) = \{ r \in R \mid r > x \wedge y > r \}$

Let us consider the APSO tuples (the analysis for BankSO and ShopSO are similar). The first tuple authorizes APSO to assign users with the prerequisite role FPS into members in the AP agent (AP). The second one authorizes APSO to assign users with the prerequisite condition FPS $\wedge \overline{OP}$ to be quality controllers (QC). Similarly, the third tuple authorizes APSO to assign users with the prerequisite condition FPS $\wedge \overline{QC}$ to be operators (OP). The second and third tuple show that the APSO can grant a user who is a member of the AP agent into one but not both of QC and OP. This illustrates how mutually exclusive roles can be forced. However, for the NSSO and SSO these are not mutually exclusive. The fourth tuple authorizes APSO to put a user who is a member of both QC and OP into a manager (M1). Of course, a user could have become a member of both QC and OP only by actions of a more powerful administrator than APSO.

Table 4. can-assign

Admin.role	Prereq.Condition	Role Range
APSO	FPS	[AP, AP]
APSO	FPS $\wedge$ $\overline{\text{OP}}$	[QC, QC]
APSO	FPS $\wedge$ $\overline{\text{QC}}$	[OP, OP]
APSO	QC $\wedge$ OP	[M1, M1]
NSSO	FPS	(FPS, DIR)
SSO	E	[FPS, FPS]
SSO	FPS	(FPS, DIR]

6. CONCLUSIONS

In this paper, a secure payment scheme and its access management are proposed. The new scheme provides different degrees 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 from the bank. Consumers can achieve a higher level of anonymity through the AP agent without revealing their private information. From the viewpoint of the bank, it is more secure because the new coin and its certificate come from the AP agent who is not involved in the payment process. The duty separation constraints of the four roles in the scheme and how to grant a role to a user associated with a can-assign relation are discussed.

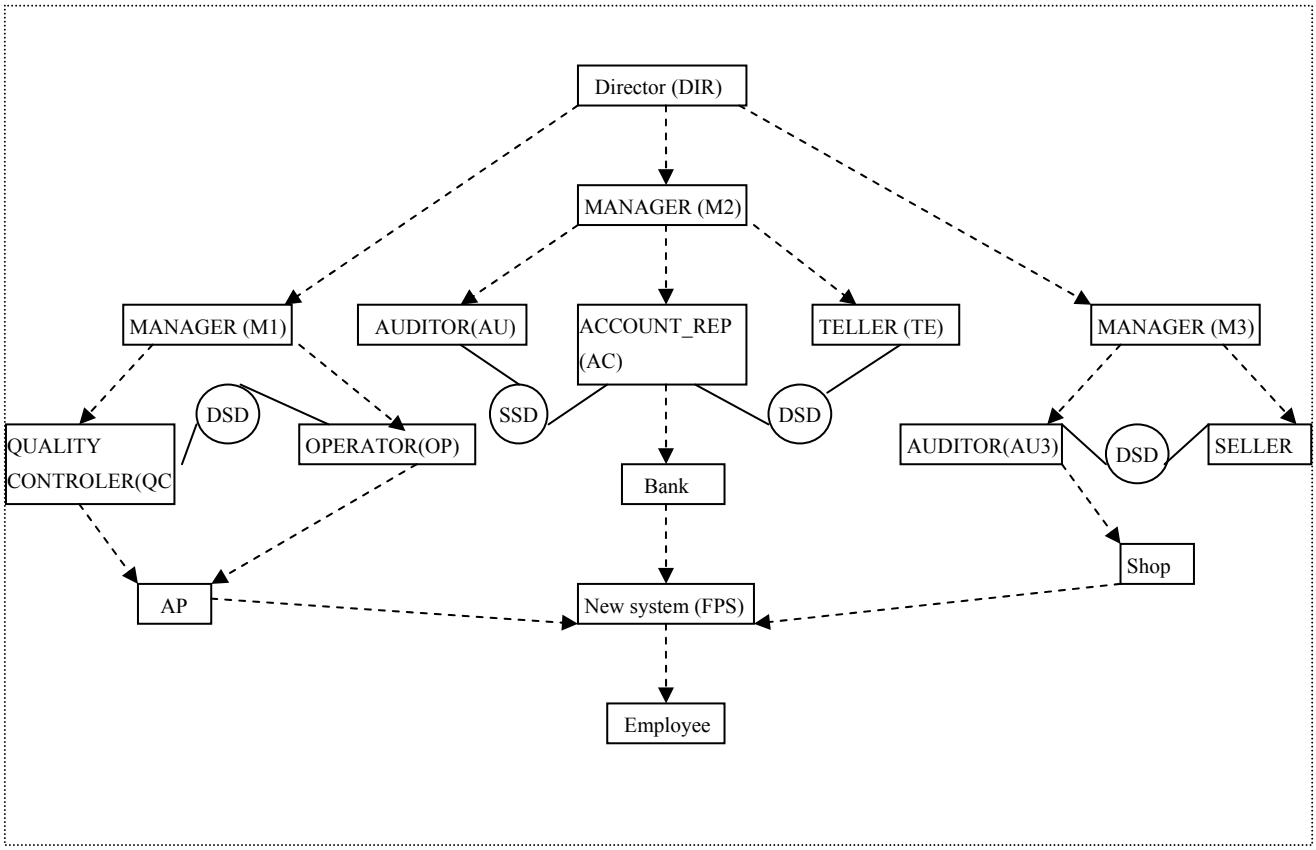

Figure 5. User-role assignment

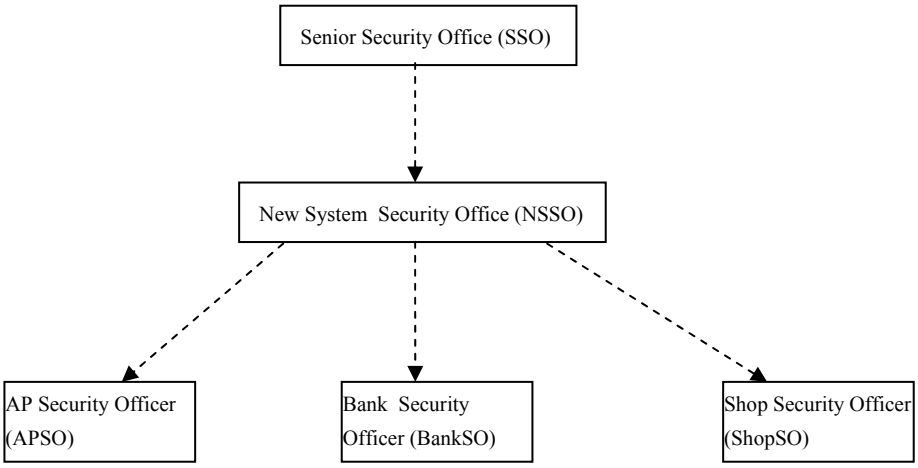

Figure 6: Administrative role assignment

REFERENCES

1. M. Peirce, PayMe: Secure Payment for World Wide Web Services, B.A. (Mod) Project Report, Computer Science Department, Trinity College Dublin, Dublin 2, Ireland. May, 1995.

2. Barkley J. F., Beznosov K. and Uppal J., Supporting Relationships in Access Control Using Role Based Access Control. The Fourth ACM Workshop on Role-Based Access Control; October, 1999: 55-65.

3 Bellare M., Goldreich O., and Krawczyk H., Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. Advances in Cryptology -- Crypto 99; 1999, Springer-Verlag No. 1666.

4. Canetti R., Goldreich O., and Halevi S., The random oracle methodology. Proceedings of the 30th ACM STOC '98; 1998: 209-218.

5. 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.

6 Chaum D., Blind signature for untraceable payments. Advances in Cryptology -- Crypto 82; Plenum Press N.Y. 1983: 199-203.

7. Chaum D., An introduction to e—cash. 1995. http://www.digicash.com.

8. Chaum D. and Van Antwerpen H., Undeniable signatures. Advances in Cryptology--Crypto89; Springer-Verlag; No. 435, 1990: 212-216.

9. Chaum D., Fiat A., and Naor M., Untraceable electronic cash. Advances in Cryptology -- Crypto 88; Springer-Verlag, No. 403; 1990: 319-327.

10. Cox B., Tygar J.D., Sirbu M., NetBill Security and Transaction Protocol. The First USENIX Workshop on Electronic Commerce; New York, 1995.

11. EIGamal T., A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory; Vol. IT-31, No.4; 1985: 469-472.

12. Feinstein H. L., Final report: NIST small business innovative research (SBIR) grant: role based access control: phase 1. Technical report; SETA Corp., Jan. 1995.

13. Ferraiolo D. F. and Kuhn D. R., Role based access control. 15th National Computer Security Conference; 1992: 554—563.

14. Ferraiolo D. F., Barkley J. F. and Kuhn D. R., Role-Based Access Control Model and Reference Implementation within a Corporate Intranet. TISSEC; 1999: 34-64.

15. Franklin M., Yung M., Secure and efficient off-line digital money. Proceedings of the Twentieth International Colloquium on Automata, Languages and Programming; Vol.700. Springer-Verlag; 1993: 265-276.

16. Goldschlag D., Reed M., and Syverson P., Onion routing for anonymous and private Internet connections. Communications of the ACM; Vol.24, No.2; 1999: 39-41.

17. MastercardVisa, SET 1.0 - Secure electronic transaction specification. 1997; http://www.mastercard.com/set.html

18. Okamoto T., An efficient divisible electronic cash scheme. Advances in Cryptology--Crypto'95; Springer-Verlag; Vol. 963; 1995: 438-451.

19. Pointcheval D., Self-Scrambling Anonymizers. Proceedings of Financial Cryptography; 2000, Anguilla, British West Indies.

20. Rivest R. L., Shamir A., and Adleman L. M., A method for obtaining digital signatures and public-Key cryptosystems. Communications of the ACM; Vol. 21, No.2 1978: 120-126.

21. Rivest R. T., The MD5 message digest algorithm. Internet RFC 1321; April, 1992.

22. Sandhu R., Future Directions in Role-Based Access Control Models. MMS, 2001; http://www.list.gmu.edu/confrnc/misconf/pdf_ver/mms01-rbac-future.pdf.

23. Sandhu R., Role activation hierarchies. Third ACM Workshop on Role-Based Access Control; October, 1998.

24. Sandhu R. and Bhamidipati V., The URA97 model for role-based administration of user-role assignment. T. Y. Lin and Xiao Qian, editors, Database Security XI: Status and Prospects; North-Holland, 1997.

25. Schnorr C. P., Efficient signature generation by smart cards. Journal of cryptology; Vol. 4 No.3; 1991:161-174.

26. Wang H. and Zhang Y., Untraceable off-line electronic cash flow in e-commerce. Proceedings of the 24th Australian Computer Science Conference ACSC2001; IEEE computer society; GoldCoast, Australia; 191-198.

27. Yiannis T., Fair off-line cash made easy. Advances in Cryptology--Asiacrypt'98; Springer-Verlag; Vol. 1346, 1998: 240-252.

28. Yiannis T., Yung M., On the security of ElGamal-based encryption. International Workshop on Practice and Theory in Public Key Cryptography (PKC '98); Springer-Verlag, Vol. 1346; Yokohama, Japan.