Study on New E-cash Systems

By Koji HIROHASHI

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Eiji OKAMOTO

February 15, 1999

Abstract

In this paper, we have examined the actual problems in the e-cash systems [14,15], and then proposed two new *untraceable off-line* e-cash systems. One is the new e-cash system with the properties of the two blind signatures presented in [2,22], which are based on Schnorr signature scheme [21], and is the system made by improving [13]. The other is the new e-cash system with the feature of Nyberg-Rueppel signature [16,17], which provides *message recovery*. Moreover, We have estimated the security in our e-cash systems from the viewpoints of *completeness*, *user's privacy in the payment*, *forgery of coins* and *double-spending detection*. Considering the cost of communication and computation, our systems are more efficient than other e-cash systems [3,11,12].

Acknowledgement

I am most grateful to Professor Eiji Okamoto for his good support and encouragement. I would like to thank Associate Professor Mineo Kaneko and Associate Professor Kunihiro Hiraishi for much advice. I am grateful to Associate Professor Atsuko Miyaji for her helpful comments.

I greatly thank Associate Mitsuru Tada and Associate Xun Yi for their eager suggestions and helpful comments. I deeply thank Dr. Hisao Sakazaki for many comments and helpful teaching.

Special thanks are due to Associate Professor Masahiro Mambo of Tohoku University for his valuable advice.

Finally, I wish to express my gratitude to all the members at Okamoto-Miyaji Laboratory for their helpful comments, nice talks and much advice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic cash systems (e-cash systems) have become one of the most important research in both practical and theoretical viewpoints. The features of e-cash systems are the following points:

1. A coin consists of some electronic data.
2. The coin can be transferred through networks.

E-cash systems mainly contain the following schemes:

- **Withdrawal:** A user withdraws an e-cash from a bank.
- **Payment:** Using the e-cash, the user buys something at a shop.
- **Deposit:** The shop deposits the e-cash to his bank account.

In addition, there are the following payment methods:

- **On-line Payment:** When a user buys something at a shop, the shop links to a bank in order to check the validity of the received e-cash, and then deposits the e-cash. That is, both *payment* and *deposit* are simultaneously executed in an on-line manner.

- **Off-line Payment:** When a user pays an e-cash to a shop, the procedure between the user and the shop can be performed without linking to a bank. The shop deposits the received e-cash afterward.

Some on-line e-cash systems have been proposed by [6,8,20]. However, since the on-line e-cash systems require that the shop confirms the validity of the received e-cash by linking to the bank, their systems are not practical from the viewpoints of turn-around-time, communication cost and database-maintenance cost. Therefore, the off-line e-cash systems are preferable from the practical viewpoint. Hereafter, we consider only *off-line payment*.

Off-line e-cash systems should also satisfy the following properties:

- **Independence:** The security of e-cash must not depend on any physical conditions. Then, the coin can be transferred through networks.
- **Security:** Nobody can copy (reuse) or forge coins.
- **Privacy (Untraceability):** The privacy of a user should be protected in the payment. That is, the relationship between the user and his purchases must be untraceable by anyone else.

These points are considered by many e-cash systems [3,4,7],[11]-[13],[18,19,22]. From [4,5], the e-cash system [4] allows the attacker to forge coins by executions of the scheme in parallel. In other words, this system is weak in *parallel attack*. In [11,12], the withdrawal scheme is not efficient because of enormous communication cost. The e-cash systems [18,19] realize the dividability that a coin can be subdivided into many pieces. However, the e-cash system [18] utilizing *cut and choose technique* makes the coin which consists of many terms (for example, 40 terms). Therefore, this system is very inefficient. On the other hand, the system [19] does not realize the unlinkability among coins divided from the same coin.

In [22], Schoenmakers presented the blind signature scheme utilizing Schnorr signature scheme [21]. This scheme has the following feature:

- The signer makes the signature using the different private key for each verifier.

Moreover, Schoenmakers proposed the e-cash system [22] with this property.

In [1], Abe and Fujisaki introduced the concept of *partially blind signature*, which holds the following property:

- Using the clear part in a message, which is the common information between a signer and each verifier, the signer creates the signature on the message. Therefore, he can assure himself that the message contains accurate information, and then signs the message.

This property has been already realized in the e-cash system [7] utilizing *cut and choose technique*. Unfortunately, this scheme is very inefficient in terms of communications during the generation of a signature when reasonable security is required. On the other hand, the previous e-cash systems [3,4,11,12,18,19,22] do not hold the feature of *partially blind signature*, because when a bank signs a message (a coin) in the withdrawal, he must assure himself that the message (the coin) contains accurate information without seeing it. Afterward, Abe and Camenisch proposed *partially blind signature* scheme [2] based on Schnorr signature scheme, which is related with the discrete logarithm problem.

In [14], Miyazaki and Sakurai presented the new e-cash system utilizing the two signature schemes [2,22]. However, this e-cash system allows anyone to forge coins. The reason is that a user can make the coin, which satisfies the verification equations, even if he does not know the private keys a bank uses in the withdrawal scheme. Therefore, we introduced the e-cash system [13] with the feature of the two signature [2,22], and then solved the problem in the e-cash system [14]. Unfortunately, this system is in danger of allowing a user to forge coin value in the withdrawal.

In [16,17], Nyberg and Rueppel introduced the signature scheme, which holds the following feature:

- **Message Recovery:** A message can be conveyed within a signature and can be recovered at a verifier's site. That is, the message need not be hashed or sent along with the signature, which saves storage space and communication bandwidth.

The previous signature schemes based on the discrete logarithm problem, such as ElGamal [9] and Schnorr signature schemes, cannot realize this property.

Utilizing the feature of this signature, Nguyen, Mu and Varadharajan proposed the e-cash system [15] with *message recovery* unlike the previous e-cash systems [3,4,7],[11]-[13],[18,19,22]. However, this e-cash system allows the forgery of coins as well as the system presented in [14].

In this paper, we will first consider the actual problems in the e-cash systems [14,15]. Secondly, we will propose two new *untraceable off-line* e-cash systems. One is the e-cash system using the two blind signature schemes proposed in [2,22], which are based on Schnorr signature scheme, and is the system made by improving [13]. The other is the e-cash system with the property of Nyberg-Rueppel signature, which provides *message recovery*. In addition, we will estimate the security and the performance in the two proposed e-cash systems.

Chapter 2

Digital Signature Schemes

In this chapter, we introduces the important digital signature schemes.

2.1 Schnorr Signature Scheme [21]

The system parameters consist of two primes p and q, where $q|p-1$, and an element $g \in \mathbb{Z}_p^*$ whose order is q. $\mathcal{H}$ is an appropriate hash function mapping into $\mathbb{Z}_q$. The signer's private and public keys are $x \in \mathbb{Z}_q$ and $h = g^x$, respectively. To sign a message m with the private key x, the signer chooses $k \in \mathbb{Z}_q$ at random, and then computes the signature (r, s) as follows:

$$\begin{aligned} r &= \mathcal{H}(m, g^k); \\ s &= rx + k \pmod{q}. \end{aligned}$$

The validity of the signature (r, s) for the message m can be confirmed if the following equality holds:

$$g^s h^{-r} = g^k.$$

2.1.1 Schoenmakers Blind Signature Scheme [22]

The system parameters p, q, g and the hash function $\mathcal{H}$ are the same as Schnorr signature scheme. $x \in \mathbb{Z}_q$ is the signer's private key, which is different for each verifier. The signer's

public key is $h = g^x$. The process to obtain the signature (r, s) on a message m from the signer can be achieved as follows:

Step1. The signer randomly selects $k \in \mathbb{Z}_q$, and then sends $\delta = h^k$ to the verifier.

Step2. The verifier generates three random numbers $y \in \mathbb{Z}_q^*$ and $a, b \in \mathbb{Z}_q$, and then computes $\alpha = h^y$ and $t = \delta g^a h^b$.

Step3. The verifier calculates $r = \mathcal{H}(\alpha, m, t)$, and then sends $r' = r + a \pmod{q}$ to the signer.

Step4. The signer sends $s' = r'x^{-1} + k \pmod{q}$ to the verifier.

Step5. The verifier obtains $s = (s' + b)y^{-1} \pmod{q}$, and then verifies the signature (r, s) from the verification equation, $\alpha^s g^{-r} = t$.

2.1.2 Partially Blind Signature Scheme [2]

The system parameters p, q, g are the same as Schnorr signature scheme. $\mathcal{H}$ is a strong hash function mapping from $\{0,1\}^*$ to $\{0,1\}^\ell$ ($\ell \approx 128$). The signer's private keys are $x_1, x_2 \in \mathbb{Z}_q^*$, while the corresponding public keys are $h_1 = g^{x_1}, h_2 = g^{x_2}$. The process to get the signature (r, s) on a message (c, m) from the signer can be performed as follows:

Step1. The verifier sends the clear part c to the signer.

Step2. The signer randomly chooses $k \in \mathbb{Z}_q$, and then sends $\delta = g^k$ to the verifier.

Step3. The verifier generates two random numbers $a, b \in \mathbb{Z}_q$, and then computes $t = \delta g^a (h_1^c h_2)^b$.

Step4. The verifier calculates $r = \mathcal{H}(c \| m \| t)$, and then sends $r' = r - a \pmod{q}$ to the signer.

Step5. The signer sends $s' = \dfrac{k - r'}{cx_1 + x_2} \pmod{q}$ to the verifier.

Step6. The verifier obtains $s = s' + b \pmod{q}$, and then confirms the signature (r, s) from the verification equation, $r = \mathcal{H}(c \| m \| g^r (h_1^c h_2)^s)$.

2.2 Nyberg-Rueppel Signature Scheme [16,17]

The system parameters p, q, g are the same as Schnorr signature scheme. The signer's private key is $x \in \mathbb{Z}_q$, while the corresponding public key is $h = g^x$. To sign a message $m \in \mathbb{Z}_p$, the signer selects $k \in \mathbb{Z}_q$ at random, and then computes

$$\begin{aligned} r &= mg^{-k}; \\ ak &= b + cx \pmod{q}, \end{aligned}$$

where (a, b, c) is a permutation $(\pm 1, \pm r', \pm s)$. If we ignore the $\pm$ signs, then the signature equation leads to the following six equations:

$$\begin{aligned} sk &= 1 + r'x \pmod{q}; \\ r'k &= 1 + sx \pmod{q}; \\ k &= s + r'x \pmod{q}; \\ sk &= r' + x \pmod{q}; \\ r'k &= s + x \pmod{q}; \\ k &= r' + sx \pmod{q}. \end{aligned}$$

The pair (r, s) turns out to be the signature of the message m. The message can be recovered by computing a verification equation:

$$m = g^{b/a} h^{c/a} r.$$

Chapter 3

E-cash Systems [14,15]

In this chapter, we display the e-cash systems [14,15], and then consider the problems in these systems.

3.1 E-cash System [14]

3.1.1 Preparation

Let p and q be primes with $p = 2q+1$. We suppose both are public. Moreover, we suppose $g \in \mathbb{G}_q$ is also public when $\mathbb{G}_q$ is a subset of $\mathbb{Z}_p^*$, consisting of order-q elements. $\mathcal{H}$ is an appropriate hash function. The bank $\mathcal{B}$ generates two private keys x_1, $x_2 \in \mathbb{Z}_q^*$, and then computes $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$, which are public keys. The user $\mathcal{U}$ has the private key u sharing with $\mathcal{B}$, which is the user identity, and the corresponding public key $v = h_1^u h_2$. c is the coin information consisting of value, expiration date and so on.

3.1.2 Withdrawal Scheme

When $\mathcal{U}$ wants to withdraw some coins, the following scheme is run:

Step1. $\mathcal{B}$ randomly picks up $k_1, k_2 \in \mathbb{Z}_q$, and then transfers c, $\delta_1 = g^{k_1}$ and $\delta_2 = g^{k_2}$ to $\mathcal{U}$.

Step2. $\mathcal{U}$ generates five random numbers a,b,y,z_1,$z_2 \in \mathbb{Z}_q$, and then computes $\alpha = v^y$, $\beta = {\delta_2}^y$, $t = \delta_1(v{\delta_2}^c)^b(h_1^c h_2)^a$ and $m = h_1^{z_1} h_2^{z_2}$.

Step3. $\mathcal{U}$ calculates $r = \mathcal{H}(\alpha \| \beta \| m \| t \| c)$, and then sends $r' = a - r \pmod q$ to $\mathcal{B}$.

Step4. $\mathcal{B}$ sends $s' = \dfrac{k_1 + r'(cx_1 + x_2)}{ux_1 + x_2 + ck_2} \pmod q$ to $\mathcal{U}$.

Step5. $\mathcal{U}$ obtains $s = (s' + b)y^{-1} \pmod q$, and then confirms the validity of the coin from the verification equation, $t = (\alpha\beta^c)^s(h_1^c h_2)^r$.

3.1.3 Payment Scheme

When $\mathcal{U}$ wants to pay the coin $M = [\alpha,\beta,m,r,s,c]$ to the shop $\mathcal{S}$, the following scheme is performed:

Step1. $\mathcal{U}$ sends the coin M to $\mathcal{S}$.

Step2. $\mathcal{S}$ checks the coin from the verification equation, $r = \mathcal{H}(\alpha \| \beta \| m \| (\alpha\beta^c)^s(h_1^c h_2)^r \| c)$, and then transfers the challenge $d \in \mathbb{Z}_q$ to $\mathcal{U}$.

Step3. $\mathcal{U}$ computes the response (r_1,r_2), where $r_1 = z_1 + udy \pmod q$ and $r_2 = z_2 + dy \pmod q$, and then sends (r_1,r_2) to $\mathcal{S}$.

Step4. $\mathcal{S}$ calculates the verification equation, $h_1^{r_1} h_2^{r_2} = \alpha^d m$. If the check is successful, then the coin is regarded to be valid.

3.1.4 Deposit Scheme

When $\mathcal{S}$ wants to deposit the coin received from $\mathcal{U}$, the following scheme is executed:

Step1. $\mathcal{S}$ sends the payment transcript (M,d,r_1,r_2) to $\mathcal{B}$.

Step2. $\mathcal{B}$ verifies the two verification equations, $r = \mathcal{H}(\alpha \| \beta \| m \| (\alpha\beta^c)^s(h_1^c h_2)^r \| c)$ and $h_1^{r_1} h_2^{r_2} = \alpha^d m$. If both are satisfied, then $\mathcal{B}$ accepts the coin.

3.2 E-cash System [15]

3.2.1 Preparation

Let p, q and g be two primes and a number, respectively, which satisfy $g^q = 1 \pmod{p}$. Then, we suppose those are public. $\mathcal{B}$ has a private key x. $\mathcal{B}$ selects w_1 and w_2 at random, and then computes $g_1 = g^{w_1}$ and $g_2 = g^{w_2}$ as well as $h_1 = g_1^x$ and $h_2 = g_2^x$. Then, we suppose g_1, g_2, h_1 and h_2 are also public. $\mathcal{U}$ has a pair of private and public keys (u, v), where $v = g_1^u g_2$. $\mathcal{B}$ registers the public key v as the user identity. $\mathcal{U}$ is given $w = v^x$ as the bank certificate of the user identity.

3.2.2 Withdrawal Scheme

When $\mathcal{U}$ wants to withdraw some coins, $\mathcal{B}$ and $\mathcal{U}$ must go through some authentication process. For each coin, the following scheme is run:

Step1. $\mathcal{B}$ chooses a random number $k \in \mathbb{Z}_q$, and then transfers $\delta = v^k$ to $\mathcal{U}$.

Step2. $\mathcal{U}$ randomly generates $y, z_1, z_2 \in \mathbb{Z}_q^*$, and then computes $\alpha = w^y$, $\beta = v^y$ and $\lambda = h_1^{z_1} h_2^{z_2}$.

Step3. Using a strong one-way hash function $\mathcal{H}$, $\mathcal{U}$ forms the message $m = \mathcal{H}(\alpha, \beta, \lambda)$, generates $a, b \in \mathbb{Z}_q^*$ at random, calculates $r = m\beta^a \delta^{by}$, and then sends $r' = rb^{-1} \pmod{q}$ to $\mathcal{B}$.

Step4. $\mathcal{B}$ sends $s' = r'x + k \pmod{q}$ to $\mathcal{U}$.

Step5. $\mathcal{U}$ removes the blind factor b, and then obtains $s = s'b + a \pmod{q}$.

Step6. $\mathcal{U}$ verifies the validity of the coin by using the equation, $\mathcal{H}(\alpha, \beta, \lambda) = \beta^{-s} \alpha^r r$.

3.2.3 Payment Scheme

When $\mathcal{U}$ wants to pay the coin $M = [\alpha, \beta, \lambda, r, s]$ to $\mathcal{S}$, the following scheme is performed:

Step1. $\mathcal{S}$ sends the random challenge $d = \mathcal{H}(\mathcal{S}\|Date\|Time\|\cdots)$ to $\mathcal{U}$.

Step2. $\mathcal{U}$ computes the response (r_1, r_2), where $r_1 = z_1 + udy \pmod q$ and $r_2 = z_2 + dy \pmod q$, and then sends M and (r_1, r_2) to $\mathcal{S}$.

Step3. $\mathcal{S}$ verifies the received coin by using the two verification equations, $\mathcal{H}(\alpha, \beta, \lambda) = \beta^{-s}\alpha^r r$ and $h_1^{r_1}h_2^{r_2} = \alpha^d\lambda$. If the checks are successful, then the coin is regarded to be valid.

3.2.4 Deposit Scheme

When $\mathcal{S}$ wants to deposit the coin M received from $\mathcal{U}$, the following scheme is executed:

Step1. $\mathcal{S}$ sends the payment transcript (M, d, r_1, r_2) to $\mathcal{B}$.

Step2. $\mathcal{B}$ confirms the two verification equations, $\mathcal{H}(\alpha, \beta, \lambda) = \beta^{-s}\alpha^r r$ and $h_1^{r_1}h_2^{r_2} = \alpha^d\lambda$. If both are satisfied, then $\mathcal{B}$ accepts the coin.

3.3 Problems in the E-cash Systems [14,15]

In these systems, anyone can forge the coin. Because

- $\mathcal{U}$ can make the coin parameters satisfying the verification equations even if he does not know the $\mathcal{B}$'s private keys.

First of all, we show the attack on the e-cash system [14]. Considering the verification equation:

$$r = \mathcal{H}(\alpha\|\beta\|m\|(\alpha\beta^c)^s(h_1^c h_2)^r\|c),$$

from $r = \mathcal{H}(\alpha\|\beta\|m\|t\|c)$, we can easily understand $t = (\alpha\beta^c)^s(h_1^c h_2)^r$. In the withdrawal, since α and β are the information which $\mathcal{B}$ do not know, it is possible for $\mathcal{U}$ to make τ_1 $(\neq 0)$ which satisfies the equation:

$$\alpha\beta^c = (h_1^c h_2)^{\tau_1}.$$

In addition, if $\mathcal{U}$ makes τ_2 ($\neq 0$), where $(h_1^c h_2)^{\tau_2} = (\alpha\beta^c)^s (h_1^c h_2)^r$, he can determine $s = (\tau_2 - r){\tau_1}^{-1}$ regardless of r. Then, $\mathcal{U}$ can get $t = (h_1^c h_2)^{\tau_2}$. Finally, if $\mathcal{U}$ calculates $r = \mathcal{H}(\alpha \| \beta \| m \| t \| c)$ by using $m = h_1^{z_1} h_2^{z_2}$, he can complete the forgery of the coin $M = [\alpha,\beta,m,r,s,c]$. In the payment, since $\mathcal{U}$ knows the powers of α and m, he can compute r_1 and r_2 satisfying $h_1^{r_1} h_2^{r_2} = \alpha^d m$. Consequently, $\mathcal{U}$ can pay the forged coin. Moreover, even if the double-spending appears, $\mathcal{B}$ cannot detect the illegal user. We reveal the actual example in Figure 3.1.

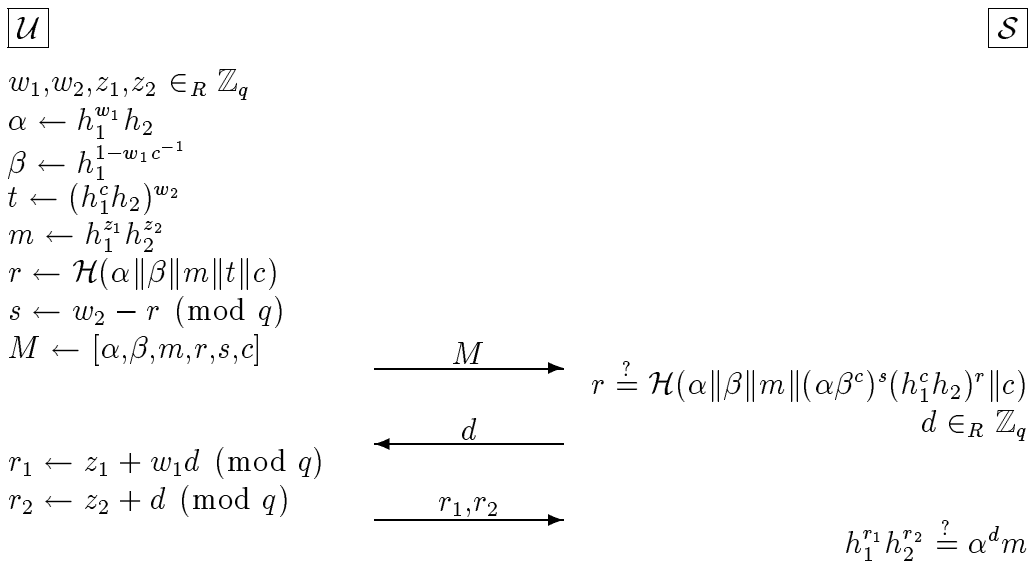

Figure 3.1: Attack on the E-cash System [14]

Similarly, we show the attack on the e-cash system [15]. In the withdrawal, since α and β are the information which $\mathcal{B}$ do not know, it is possible for $\mathcal{U}$ to make $\alpha = \beta^\tau$ and $\beta = h_1^{\tau_1} h_2^{\tau_2}$ ($\tau, \tau_1, \tau_2 \neq 0$). After computing $m = \mathcal{H}(\alpha, \beta, \lambda)$ by using $\lambda = h_1^{z_1} h_2^{z_2}$, $\mathcal{U}$ makes τ_3 ($\neq 0$), and then calculates the equation:

$$r = m\beta^{\tau_3}.$$

Then, from the verification equation, we can easily understand

$$\begin{aligned} r &= m\beta^s \alpha^{-r} \\ &= m\beta^{s-\tau r}. \end{aligned}$$

Therefore, when $\mathcal{U}$ determines s as $s = \tau r + \tau_3$, he can complete the forgery of the coin $M = [\alpha,\beta,\lambda,r,s]$. In the payment, since $\mathcal{U}$ knows the powers of α and λ, he can compute r_1 and r_2 satisfying $h_1^{r_1} h_2^{r_2} = \alpha^d \lambda$. Consequently, $\mathcal{U}$ can pay the forged coin. Moreover, even if the double-spending appears, $\mathcal{B}$ cannot detect the illegal user. We reveal the actual example in Figure 3.2.

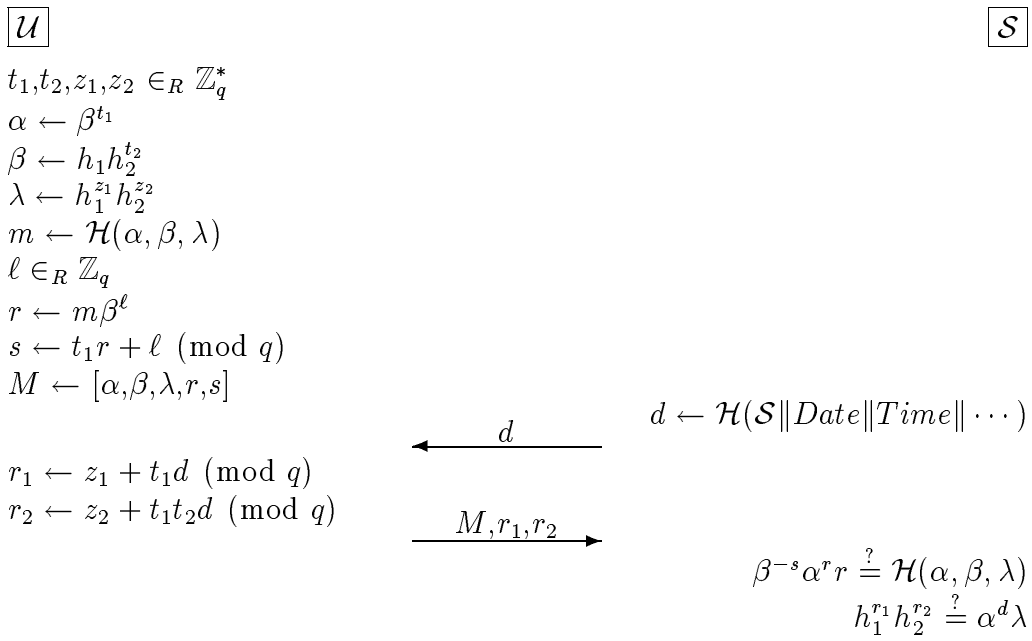

Figure 3.2: Attack on the E-cash System [15]

Chapter 4

New E-cash System 1

In this chapter, by the improvement of the system [13], we propose the new e-cash system using the two blind signature schemes, which are Schoenmakers blind signature scheme [22] and *partially blind signature* scheme [2], and then estimate the security and the performance in the system.

4.1 System Setup

Let p and q be primes which satisfy $q|p-1$. We suppose both are public. Moreover, we suppose $g \in \mathbb{G}_q \backslash \{1\}$ is also public when $\mathbb{G}_q$ is a subgroup of $\mathbb{Z}_p^*$ of order q. $\mathcal{H}$ is the strong one-way hash function mapping from $\{0,1\}^*$ to $\{0,1\}^\ell$ ($\ell \approx 128$). Let $\|$ denote concatenation. $\mathcal{B}$ generates three private keys x, x_1, $x_2 \in \mathbb{Z}_q^*$, and then computes $h = g^x$, $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$, which are public keys.

4.2 $\mathcal{U}$'s Account Establishment

$\mathcal{U}$ shows (by physical or other means) $u \in \mathbb{Z}_q^*$ to $\mathcal{B}$. If $h_1^u \neq 1$ and $h_1^u h_2 \neq 1$ are satisfied, then $\mathcal{B}$ registers u. In other words, $\mathcal{U}$ is assumed to have in common with $\mathcal{B}$ the user identity u.

4.3 Withdrawal Scheme

When $\mathcal{U}$ wants to withdraw some coins from $\mathcal{B}$, he must prove the ownership of his account by some means. Then, the following scheme is performed:

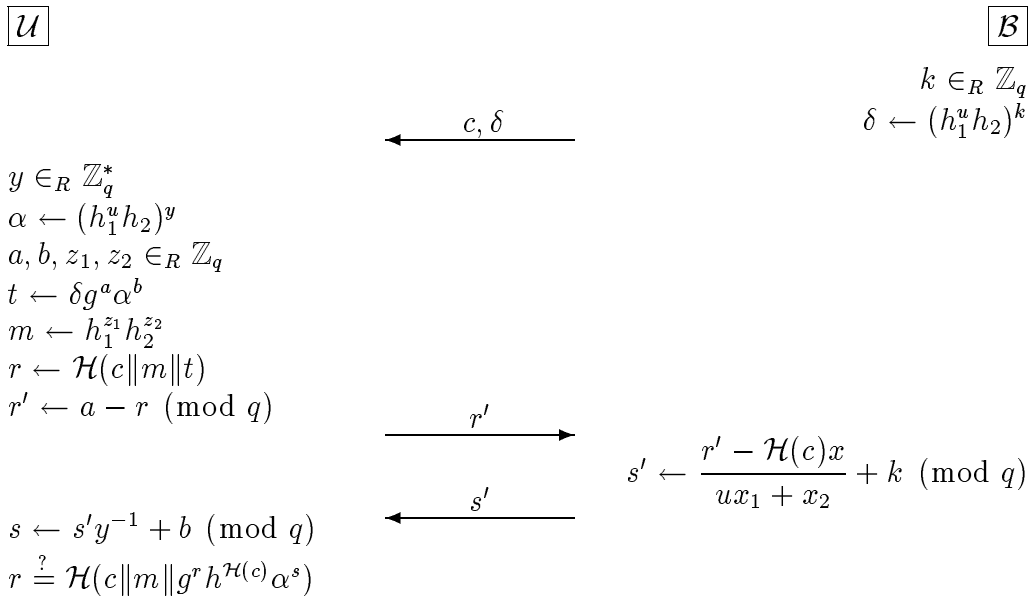

Figure 4.1: Withdrawal Scheme

Step1. $\mathcal{B}$ generates at random a number $k \in \mathbb{Z}_q$, and then sends c and $\delta = (h_1^u h_2)^k$ to $\mathcal{U}$. c is the coin information consisting of value, expiration date and so on.

Step2. $\mathcal{U}$ calculates $\alpha = (h_1^u h_2)^y$ after choosing $y \in \mathbb{Z}_q^*$ at random. $\mathcal{U}$ also generates four random numbers $a, b, z_1, z_2 \in \mathbb{Z}_q$, and then computes $t = \delta g^a \alpha^b$ and $m = h_1^{z_1} h_2^{z_2}$.

Step3. $\mathcal{U}$ calculates $r = \mathcal{H}(c\|m\|t)$, and then sends $r' = a - r \pmod q$ to $\mathcal{B}$.

Step4. $\mathcal{B}$ sends $s' = \dfrac{r' - \mathcal{H}(c)x}{ux_1 + x_2} + k \pmod q$ to $\mathcal{U}$.

Step5. $\mathcal{U}$ computes $s = s' y^{-1} + b \pmod q$.

Step6. $\mathcal{U}$ accepts if and only if $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)} \alpha^s)$.

4.4 Payment Scheme

When $\mathcal{U}$ wants to pay the coin $M = [\alpha,c,m,r,s]$ to $\mathcal{S}$, the following scheme is executed:

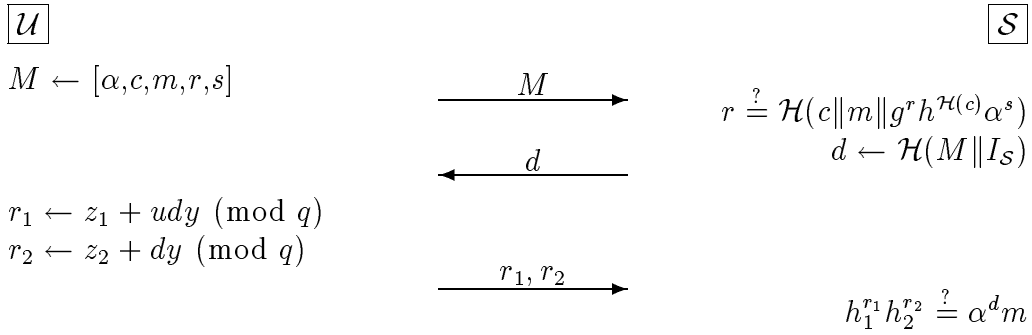

Figure 4.2: Payment Scheme

Step1. $\mathcal{U}$ transfers the coin M to $\mathcal{S}$.

Step2. $\mathcal{S}$ verifies the equation, $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)}\alpha^s)$, and then sends the challenge $d = \mathcal{H}(M\|I_{\mathcal{S}})$ to $\mathcal{U}$. $I_{\mathcal{S}}$ contains data and time of the payment, the shop identity, and possibly some random bits to deal with the problem of *double-deposits*.

Step3. $\mathcal{U}$ sends the response (r_1, r_2), where $r_1 = z_1 + udy \pmod q$ and $r_2 = z_2 + dy \pmod q$, to $\mathcal{S}$.

Step4. $\mathcal{S}$ checks the equation, $h_1^{r_1} h_2^{r_2} = \alpha^d m$.

Step5. $\mathcal{S}$ accepts if and only if the two verification equations are successful.

4.5 Deposit Scheme

When $\mathcal{S}$ wants to deposit the coin M at $\mathcal{B}$, the following scheme is run:

Step1. $\mathcal{S}$ sends the payment transcript $(M, I_{\mathcal{S}}, r_1, r_2)$ to $\mathcal{B}$.

Step2. $\mathcal{B}$ computes $d = \mathcal{H}(M\|I_{\mathcal{S}})$.

Step3. $\mathcal{B}$ accepts if and only if $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)}\alpha^s)$ and $h_1^{r_1} h_2^{r_2} = \alpha^d m$.

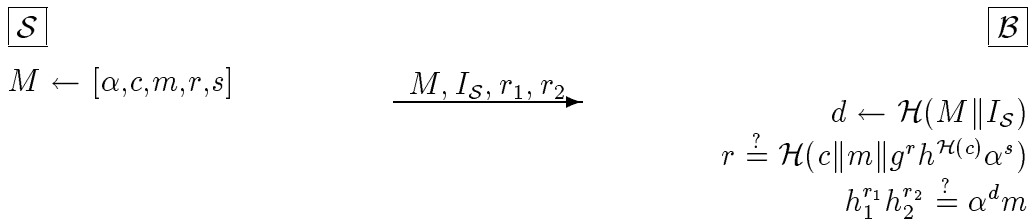

Figure 4.3: Deposit Scheme

4.6 Security

This section is due to [4,22] to some extent. Following Feige, Fiat and Shamir [10], we denote by $\overline{\mathcal{Z}}$ a party $\mathcal{Z}$ that follows the schemes, and by $\widetilde{\mathcal{Z}}$ a party $\mathcal{Z}$ with unlimited computing power that may deviate from the schemes in an arbitrary way. $\mathcal{Z}$ denotes either one of these.

4.6.1 Completeness

We say that an e-cash system is *complete* if the system satisfies all the following properties:

(1) If $\overline{\mathcal{U}}$ accepts in the withdrawal scheme, and sends the coin and the response in the payment scheme, then $\overline{\mathcal{S}}$ accepts.

(2) If $\overline{\mathcal{S}}$ accepts in the payment scheme, and deposits the payment transcript in the deposit scheme, then $\overline{\mathcal{B}}$ accepts.

Proposition 1 *New e-cash system 1 is complete.*

Proof. First, we prove the property (1). $\overline{\mathcal{S}}$ accepts if

$$r = \mathcal{H}(c \| m \| g^r h^{\mathcal{H}(c)} \alpha^s)$$

and

$$h_1^{r_1} h_2^{r_2} = \alpha^d m.$$

In the withdrawal, $\mathcal{U}$ computes

$$r = \mathcal{H}(c\|m\|\delta g^a \alpha^b)$$

and

$$m = h_1^{z_1} h_2^{z_2}.$$

Therefore, it suffices to prove that

$$g^r h^{\mathcal{H}(c)} \alpha^s = \delta g^a \alpha^b$$

and

$$h_1^{r_1} h_2^{r_2} = \alpha^d h_1^{z_1} h_2^{z_2}$$

for the assignments made by $\mathcal{U}$ in the schemes.

The first equality follows from

$$\begin{aligned} g^r h^{\mathcal{H}(c)} \alpha^s &= g^r g^{\mathcal{H}(c)x} \cdot (h_1^u h_2)^{ys} \\ &= g^{r+\mathcal{H}(c)x} \cdot g^{r'-\mathcal{H}(c)x} (h_1^u h_2)^k \alpha^b \\ &= g^r g^{a-r} (h_1^u h_2)^k \alpha^b \\ &= (h_1^u h_2)^k g^a \alpha^b \\ &\overset{(*)}{=} \delta g^a \alpha^b \end{aligned}$$

and the second from

$$\begin{aligned} h_1^{r_1} h_2^{r_2} &= h_1^{z_1 + udy} h_2^{z_2 + dy} \\ &= (h_1^u h_2)^{yd} \cdot h_1^{z_1} h_2^{z_2} \\ &= \alpha^d m. \end{aligned}$$

The substitution in $(*)$ is allowed because $\overline{\mathcal{U}}$ accepts in the withdrawal only if $(h_1^u h_2)^k = g^{-r'} h^{\mathcal{H}(c)} (h_1^u h_2)^{s'} = \delta$.

The other property (2) is immediately clear from the fact that the shop identity in $I_{\mathcal{S}}$ differs per shop and $\overline{\mathcal{S}}$ does not use the same value for $I_{\mathcal{S}}$ in two different payments, since the verification relations that are applied by $\overline{\mathcal{B}}$ in the deposit scheme are the same as those applied by $\overline{\mathcal{S}}$ in the payment scheme.

■

4.6.2 Privacy

We say that the an e-cash system protects the privacy of the user in the payment if the system holds the following property:

- If $\mathcal{U}$ follows the schemes, and does not double-spend, then no shared information can be developed between $\mathcal{B}$ and $\mathcal{S}$ in the executions of the withdrawal and payment schemes that $\mathcal{U}$ takes part in.

First, we show the following lemma:

Lemma 1 *For any $\overline{\mathcal{U}}$, for any possible view of $\widetilde{\mathcal{B}}$ in an execution of the withdrawal scheme in which $\overline{\mathcal{U}}$ accepts and for any possible view of $\widetilde{\mathcal{S}}$ in an execution of the payment scheme in which the payer follows the scheme, there is exactly one set of random choices that $\overline{\mathcal{U}}$ could have made in the execution of the withdrawal scheme such that the views of $\widetilde{\mathcal{B}}$ and $\widetilde{\mathcal{S}}$ correspond to the withdrawal and payment of the same e-cash.*

Proof. We define the following sets:

- Viewer($\widetilde{\mathcal{B}}$) $= \{(c,\delta,r',s') \mid \delta \in \mathbb{G}_q$ and $r', s' \in \mathbb{Z}_q$ such that $g^{-r'}h^{\mathcal{H}(c)}(h_1^u h_2)^{s'} = (h_1^u h_2)^k = \delta\}$
- Viewer($\widetilde{\mathcal{S}}$) $= \{(\alpha, c, m, r, s, d, r_1, r_2) \mid \alpha, m \in \mathbb{G}_q$, $r, d \in \{0,1\}^\ell$ and $s, r_1, r_2 \in \mathbb{Z}_q$ such that $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)}\alpha^s)$ and $h_1^{r_1}h_2^{r_2} = \alpha^d m\}$
- Choices($\mathcal{U}$) $= \{(a, b, y, z_1, z_2) \mid a, b, z_1, z_2 \in \mathbb{Z}_q$ and $y \in \mathbb{Z}_q^*\}$

We have only to show that for all $\widetilde{\mathcal{B}}$-view $\in$ Views($\widetilde{\mathcal{B}}$) and for all $\widetilde{\mathcal{S}}$-view $\in$ Views($\widetilde{\mathcal{S}}$), there is exactly one tuple $(a, b, y, z_1, z_2) \in$ Choices($\mathcal{U}$) such that $\widetilde{\mathcal{B}}$-view and $\widetilde{\mathcal{S}}$-view correspond to the withdrawal and payment of the same e-cash.

First, y is uniquely determined from α and $v = h_1^u h_2$ as $y = \log_v \alpha$. From r_1, u, d and y, we see that the choice $z_1 = r_1 - udy \pmod q$ must have been made, and from r_2, d and y, it follows that $z_2 = r_2 - dy \pmod q$ must have been chosen. The choice r together with r' determines a as $a = r + r' \pmod q$. Finally, the numbers s, s' and y determine b as $b = s - s'y^{-1} \pmod q$.

For these choices of the five variables, all the assignments and verifications in the two scheme executions would be satisfied by definition, except for the assignments $t = \delta g^a \alpha^b$, $m = h_1^{z_1} h_2^{z_2}$ and $r = \mathcal{H}(c\|m\|t)$ that must have been made by $\mathcal{U}$ in the withdrawal scheme. To prove that these assignments hold as well, we notice that from $\widetilde{\mathcal{S}}$-view $\in$ Views($\widetilde{\mathcal{S}}$) we have that

$$r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)} \alpha^s)$$

and

$$h_1^{r_1} h_2^{r_2} = \alpha^d m.$$

Therefore, the proof is completed if

$$g^r h^{\mathcal{H}(c)} \alpha^s = \delta g^a \alpha^b$$

and

$$h_1^{r_1} h_2^{r_2} = \alpha^d h_1^{z_1} h_2^{z_2}$$

for (a, b, y, z_1, z_2) made above. This is obvious in the proof of *proposition 1*, considering that in the case the substitution in $(*)$ is allowed because $\widetilde{\mathcal{B}}$-view $\in$ Views($\widetilde{\mathcal{B}}$).

■

Proposition 2 *New e-cash system 1 protects the privacy of the user in the payment.*

Proof. This is an immediate consequence of *lemma 1* and the fact that $\mathcal{U}$ in the withdrawal scheme generates (a, b, y, z_1, z_2) uniformly at random from Choices($\mathcal{U}$).

■

4.6.3 Forgery

To forge a coin, the two verification equations, $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)} \alpha^s)$ and $h_1^{r_1} h_2^{r_2} = \alpha^d m$, must be satisfied. We say that illegal users cannot forge a coin in an e-cash system if the system is protected from all the following attacks:

- **Forgery without the Withdrawal Scheme**
  - [**Attack 1**] Some users make a coin without the use of coin parameters.
  - [**Attack 2**] Some users make a coin from two (or more) coins.
- **Forgery in the Withdrawal Scheme**
  - [**Attack 3**] A user executes the withdrawal scheme by himself, and then frames up a coin.
  - [**Attack 4**] Two (or more) users simultaneously execute the withdrawal scheme in parallel, and then frame up a coin with cooperation (*parallel attack*).

Proposition 3 *Illegal users cannot forge a coin in new e-cash system 1.*

Proof.

[**Attack 1**]

In this attack, some users must make a coin only from the two verification equations, $r = \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)}\alpha^s)$ and $h_1^{r_1}h_2^{r_2} = \alpha^d m$. First of all, considering $h_1^{r_1}h_2^{r_2} = \alpha^d m$, since $\mathcal{U}$ cannot know $\mathcal{B}$'s private keys (x,x_1,x_2) because of the difficulty of the discrete logarithm problem, some users should determine α as $\alpha = h_1^{\varepsilon_1}h_2^{\varepsilon_2}$, where $\varepsilon_1 \neq 0$ and $\varepsilon_2 \neq 0$. Since δ, g^a, α^b and $h^{\mathcal{H}(c)}$ are quite independent of r and s, from

$$g^r h^{\mathcal{H}(c)}\alpha^s = \delta g^a \alpha^b,$$

some users can obtain the following equation:

$$g^r(h_1^{\varepsilon_1}h_2^{\varepsilon_2})^s = g^D,$$

where $g^D = \delta g^a \alpha^b h^{-\mathcal{H}(c)}$. However, as

$$s = \frac{D - r}{\varepsilon_1 x_1 + \varepsilon_2 x_2},$$

the relationship between r and s requires $\mathcal{B}$'s private keys (x_1, x_2).

[Attack 2]

This is the attack that some users make a coin by mixing two (or more) different coins. Now, we suppose that two users $\mathcal{U}_A$ and $\mathcal{U}_B$ have two coins M_A and M_B, respectively, where $M_i = [\alpha_i, c_i, m_{AB}, r_i, s_i]$ $(i = A, B)$. Assuming that

$$\begin{aligned} r &= \mathcal{H}(c\|m\|g^r h^{\mathcal{H}(c)}\alpha^s) \\ &= \mathcal{H}(\mu_1 c_A + \mu_2 c_B\|(\mu_1+\mu_2)m_{AB}\|g^{\mu_1 r_A+\mu_2 r_B} h^{\mathcal{H}(\mu_1 c_A+\mu_2 c_B)}\alpha^{\mu_1 s_A+\mu_2 s_B}), \end{aligned}$$

where $\mu_1 \neq 0$ and $\mu_2 \neq 0$, $\mathcal{U}_A$ and $\mathcal{U}_B$ wish to satisfy the equation, $r = \mu_1 r_A + \mu_2 r_B$. However, since

$$\mu_1 r_A + \mu_2 r_B = \mu_1 \mathcal{H}(c_A\|m_{AB}\|g^{r_A} h^{\mathcal{H}(c_A)}\alpha_A^{s_A}) + \mu_2 \mathcal{H}(c_B\|m_{AB}\|g^{r_B} h^{\mathcal{H}(c_B)}\alpha_B^{s_B}),$$

we see that generally $r \neq \mu_1 r_A + \mu_2 r_B$.

[Attack 3]

As $\mathcal{B}$'s signature s' contains $\dfrac{r'}{ux_1 + x_2}$, it is impossible for $\mathcal{U}$ to frame up the user identity without $\mathcal{B}$'s private keys (x_1, x_2) in the withdrawal. Now, we consider the forgery of the coin value. In the withdrawal scheme, $\mathcal{U}$ computes $t = \delta^\mu g^a \alpha^b$ and $r = \mathcal{H}(\mu c\|m\|t)$, where $\mu \neq 0$, and then sends $r' = (a-r)\mu^{-1} \pmod q$ to $\mathcal{B}$. Getting $s' = \dfrac{r' - \mathcal{H}(c)x}{ux_1 + x_2} + k \pmod q$, $\mathcal{U}$ calculates $s = s'\mu y^{-1} + b \pmod q$, and then verifies the following equation:

$$\begin{aligned} t &= g^r h^{\mathcal{H}(\mu c)}\alpha^s \\ &= g^r h^{\mathcal{H}(\mu c)} \cdot g^{a-r} h^{-\mu\mathcal{H}(c)} \delta^\mu \alpha^b \\ &= t h^{\mathcal{H}(\mu c) - \mu\mathcal{H}(c)}. \end{aligned}$$

However, we see that generally $\mathcal{H}(\mu c) \neq \mu\mathcal{H}(c)$.

Remark:

This attack succeeds in the e-cash system [13]. The reason is that $\mathcal{B}$ determines s' as $s' = \dfrac{r' - cx}{ux_1 + x_2} + k \pmod q$. Therefore, *new e-cash system 1* prevents the attack by $s' = \dfrac{r' - \mathcal{H}(c)x}{ux_1 + x_2} + k \pmod q$.

[Attack 4]

Now, we suppose that two users $\mathcal{U}_A$ and $\mathcal{U}_B$ perform the withdrawal scheme in parallel. First, $\mathcal{U}_A$ and $\mathcal{U}_B$ get (c_A, δ_A) and (c_B, δ_B), respectively, where $\delta_i = (h_1^{u_i} h_2)^{k_i}$ $(i = A, B)$.

Assuming that u includes u_A and u_B, they compute $\alpha = (h_1^u h_2)^y$, $t = \delta_A^{\mu}\delta_B g^a \alpha^b$ and $r = \mathcal{H}(\mu c_A + c_B \| m \| t)$, where $\mu \neq 0$. They send ${r'}_A = (a - r)(2\mu)^{-1} \pmod q$ and ${r'}_B = (a - r)2^{-1} \pmod q$, respectively. After obtaining $({s'}_A, {s'}_B)$, respectively, where ${s'}_i = \dfrac{{r'}_i - \mathcal{H}(c_i)x}{u_i x_1 + x_2} + k_i \pmod q \ (i = A, B)$, they calculate $s = (\mu {s'}_A + {s'}_B)y^{-1} + b \pmod q$, and then confirm the following equation:

$$
\begin{aligned}
t &= g^r h^{\mathcal{H}(\mu c_A + c_B)} \alpha^s \\
&= g^r h^{\mathcal{H}(\mu c_A + c_B)} \cdot (h_1^u h_2)^{\frac{(a-r)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{(a-r)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2} + \mu k_A + k_B} \alpha^b \\
&= g^r h^{\mathcal{H}(\mu c_A + c_B)} (h_1^u h_2)^{\frac{(a-r)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{(a-r)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2} + \mu k_A + k_B} \cdot t \delta_A^{-\mu} \delta_B^{-1} g^{-a} \\
&= t \cdot g^{r-a} h^{\mathcal{H}(\mu c_A + c_B)} (h_1^u h_2)^{\frac{(a-r)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{(a-r)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2} + \mu k_A + k_B} \delta_A^{-\mu} {\delta_B}^{-1} \\
&= t \cdot g^{r-a+\mathcal{H}(\mu c_A + c_B)x} (h_1^u h_2)^{\frac{(a-r)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{(a-r)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2}} h_1^{u(\mu k_A + k_B) - (\mu k_A u_A + k_B u_B)}.
\end{aligned}
$$

Then, the equation:

$$
g^{\theta} (h_1^u h_2)^{\frac{\theta_1}{u_A x_1 + x_2} + \frac{\theta_2}{u_B x_1 + x_2}} h_1^{u(\mu k_A + k_B) - (\mu k_A u_A + k_B u_B)} = 1,
$$

where

$$
\begin{aligned}
\theta &= r - a + \mathcal{H}(\mu c_A + c_B)x; \\
\theta_1 &= (a - r)2^{-1} - \mu\mathcal{H}(c_A)x; \\
\theta_2 &= (a - r)2^{-1} - \mathcal{H}(c_B)x,
\end{aligned}
$$

must be satisfied. Therefore, they can get the following equations:

$$
\begin{cases}
u_A u_B \theta + u(u_A \theta_2 + u_B \theta_1) = 0; \\
u(\theta_1 + \theta_2) + u_A(\theta + \theta_2) + u_B(\theta + \theta_1) = 0; \\
\theta + \theta_1 + \theta_2 = 0; \\
\mu k_A u_A u_B (u - u_A) + k_B u_A u_B (u - u_B) = 0; \\
\mu k_A (u_A + u_B)(u - u_A) + k_B (u_A + u_B)(u - u_B) = 0; \\
\mu k_A (u - u_A) + k_B (u - u_B) = 0.
\end{cases}
$$

However, it is possible to satisfy these equations only if $u_A = u_B$.

■

4.6.4 Double-spending Detection

If $\mathcal{U}$ has double-spent a coin, $\mathcal{B}$ will be able to obtain the responses (r_1, r_2) and (r_1', r_2') for two different challenges d and d', where $r_1 = z_1 + udy \pmod q$, $r_2 = z_2 + dy \pmod q$, $r_1' = z_1 + ud'y \pmod q$ and $r_2' = z_2 + d'y \pmod q$. Then, $\mathcal{B}$ can compute

$$\begin{aligned} r_1 - r_1' &= u(d - d')y; \\ r_2 - r_2' &= (d - d')y. \end{aligned}$$

From $u(d - d')y$ and $(d - d')y$, $\mathcal{B}$ can easily obtain u. Therefore, $\mathcal{B}$ can determine the double-spender.

4.7 Performance Evaluation

In this section, we compare the efficiency of *new e-cash system 1* with that of the off-line e-cash systems [3,11,12], which are famous and secure. The efficiency of e-cash systems is estimated by the cost of communication and computation. We suppose that the communication cost depends on the number of exponential operation in each scheme, and that the computation cost relies on the communication amount of parameters in each scheme. Now, we assume in Brands scheme [3], $|p| = 1024$, $|q| = 160$, in Ferguson scheme [11,12], $|n| = 1024$, $|v| = 160$ and in our system, $|p| = 1024$, $|q| = 160$, $|c| = 160$, where $|\cdot|$ denotes binary length. Then, we get the following results on Table 4.1.

	Communication Amount [bits]		Number of Exponentiation			
	Withdrawal	Payment	Withdrawal		Payment	
			$\mathcal{U}$	$\mathcal{B}$	$\mathcal{U}$	$\mathcal{S}$
Brands System [3]	2368	5760	15	3	0	7
Ferguson System [11,12]	10880	4416	17	9	1	8
New E-cash System 1	1504	2944	10	3	0	6

Table 4.1: Comparison between E-cash Systems

In the withdrawal, the communication amount of our system is smaller than Ferguson and Brands systems. Moreover, the number of exponential operation imposed on $\mathcal{U}$ is

also smaller than other systems [3,11,12]. The number of exponential operation imposed on $\mathcal{B}$ in our system is the same as Brands system, and the number in both systems is smaller than that in Ferguson system.

In the payment, the communication amount of our system is smaller than other e-cash systems [3,11,12]. In our system and Brands system, $\mathcal{U}$ do not need exponentiations. The number of exponential operation imposed on $\mathcal{S}$ is smaller than Ferguson and Brands systems.

Therefore, we see that *new e-cash system 1* is more efficient than other e-cash systems [3,11,12].

Chapter 5

New E-cash System 2

In this chapter, we propose the new e-cash system using Nyberg-Rueppel signature scheme [16,17], which provides *message recovery*, and then estimate the security and the performance in the system.

5.1 System Setup

Let p and q be primes which satisfy $q|p-1$. We suppose both are public. Moreover, we suppose $g \in \mathbb{G}_q \backslash \{1\}$ is also public when $\mathbb{G}_q$ is a subgroup of $\mathbb{Z}_p^*$ of order q. $\mathcal{H}$ is the strong one-way hash function mapping from $\{0,1\}^*$ to $\{0,1\}^\ell$ ($\ell \approx 128$). Let $\|$ denote concatenation. $\mathcal{B}$ generates three private keys x, x_1, $x_2 \in \mathbb{Z}_q^*$, and then computes $h = g^x$, $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$, which are public keys.

5.2 $\mathcal{U}$'s Account Establishment

$\mathcal{U}$ shows (by physical or other means) $u \in \mathbb{Z}_q^*$ to $\mathcal{B}$. If $h_1^u \neq 1$ and $h_1^u h_2 \neq 1$ are satisfied, then $\mathcal{B}$ registers u. In other words, $\mathcal{U}$ is assumed to have in common with $\mathcal{B}$ the user identity u.

5.3 Withdrawal Scheme

When $\mathcal{U}$ wants to withdraw some coins from $\mathcal{B}$, he must prove the ownership of his account by some means. Then, the following scheme is performed:

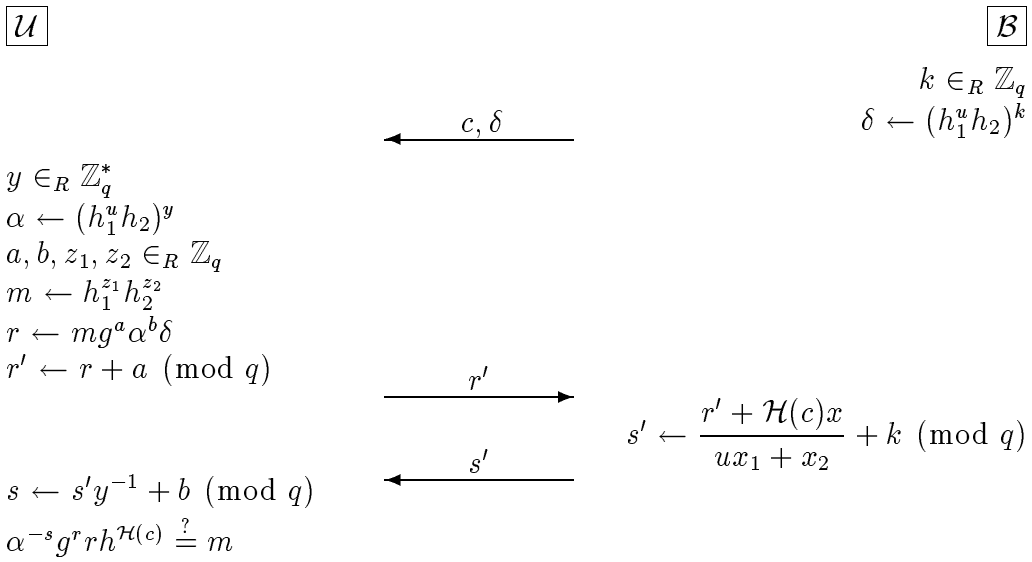

Figure 5.1: Withdrawal Scheme

Step1. $\mathcal{B}$ generates a random number $k \in \mathbb{Z}_q$, and then sends c and $\delta = (h_1^u h_2)^k$ to $\mathcal{U}$. c is the coin information consisting of value, expiration date and so on.

Step2. $\mathcal{U}$ calculates $\alpha = (h_1^u h_2)^y$ after choosing $y \in \mathbb{Z}_q^*$ at random. $\mathcal{U}$ also generates four random numbers $a, b, z_1, z_2 \in \mathbb{Z}_q$, and then computes $m = h_1^{z_1} h_2^{z_2}$ and $r = m g^a \alpha^b \delta$.

Step3. $\mathcal{U}$ sends $r' = r + a \pmod q$ to $\mathcal{B}$.

Step4. $\mathcal{B}$ sends $s' = \dfrac{r' + \mathcal{H}(c)x}{ux_1 + x_2} + k \pmod q$ to $\mathcal{U}$.

Step5. $\mathcal{U}$ computes $s = s' y^{-1} + b \pmod q$.

Step6. $\mathcal{U}$ accepts if and only if $\alpha^{-s} g^r r h^{\mathcal{H}(c)} = m$.

5.4 Payment Scheme

When $\mathcal{U}$ wants to pay the coin $M = [\alpha,c,r,s]$ to $\mathcal{S}$, the following scheme is executed:

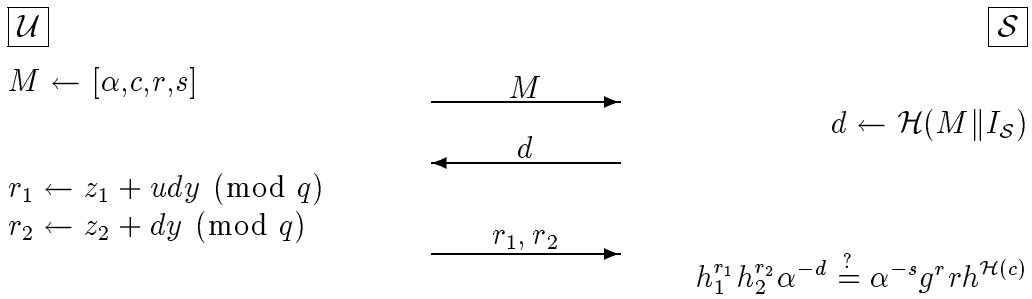

Figure 5.2: Payment Scheme

Step1. $\mathcal{U}$ transfers the coin M to $\mathcal{S}$.

Step2. $\mathcal{S}$ sends the challenge $d = \mathcal{H}(M \| I_{\mathcal{S}})$ to $\mathcal{U}$. $I_{\mathcal{S}}$ contains data and time of the payment, the shop identity, and possibly some random bits to deal with the problem of *double-deposits*.

Step3. $\mathcal{U}$ sends the response (r_1, r_2), where $r_1 = z_1 + udy \pmod q$ and $r_2 = z_2 + dy \pmod q$, to $\mathcal{S}$.

Step4. $\mathcal{S}$ checks the equation, $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$.

Step5. $\mathcal{S}$ accepts if and only if the verification equation is successful.

5.5 Deposit Scheme

When $\mathcal{S}$ wants to deposit the coin M at $\mathcal{B}$, the following scheme is run:

Step1. $\mathcal{S}$ sends the payment transcript $(M, I_{\mathcal{S}}, r_1, r_2)$ to $\mathcal{B}$.

Step2. $\mathcal{B}$ computes $d = \mathcal{H}(M \| I_{\mathcal{S}})$.

Step3. $\mathcal{B}$ accepts if and only if $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$.

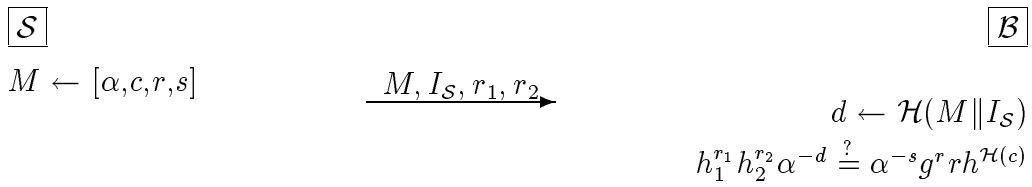

Figure 5.3: Deposit Scheme

5.6 Security

The system parameters and definitions are the same as *new e-cash system 1.*

5.6.1 Completeness

We can see that the statement of the proposition in the previous chapter also hold for *new e-cash system 2.*

Proposition 4 *New e-cash system 2 is complete.*

Proof. First, we prove the property (1). $\overline{\mathcal{S}}$ accepts if

$$h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}.$$

In the withdrawal, $\mathcal{U}$ computes

$$r = m g^a \alpha^b \delta$$

and

$$m = h_1^{z_1} h_2^{z_2} = \alpha^{-s} g^r r h^{\mathcal{H}}(c).$$

Therefore, it suffices to prove that

$$\alpha^s g^{-r} h^{-\mathcal{H}(c)} = g^a \alpha^b \delta$$

and

$$h_1^{r_1} h_2^{r_2} \alpha^{-d} = h_1^{z_1} h_2^{z_2}$$

for the assignments made by $\mathcal{U}$ in the schemes.

The first equality follows from

$$\begin{aligned}
\alpha^s g^{-r} h^{-\mathcal{H}(c)} &= (h_1^u h_2)^{ys} \cdot g^{-r} g^{-\mathcal{H}(c)x} \\
&= g^{r'+\mathcal{H}(c)x} (h_1^u h_2)^k \alpha^b \cdot g^{-r} g^{-\mathcal{H}(c)x} \\
&= g^{r+a} \alpha^b (h_1^u h_2)^k g^{-r} \\
&= g^a \alpha^b (h_1^u h_2)^k \\
&\overset{(*)}{=} g^a \alpha^b \delta
\end{aligned}$$

and the second from

$$\begin{aligned}
h_1^{r_1} h_2^{r_2} \alpha^{-d} &= h_1^{z_1 + u\,dy} h_2^{z_2 + dy} \cdot (h_1^u h_2)^{-yd} \\
&= h_1^{z_1 + u\,dy} h_2^{z_2 + dy} \cdot h_1^{-u\,dy} h_2^{-dy} \\
&= h_1^{z_1} h_2^{z_2} \\
&= m.
\end{aligned}$$

The substitution in $(*)$ is allowed because $\overline{\mathcal{U}}$ accepts in the withdrawal only if $(h_1^u h_2)^k = (h_1^u h_2)^{s'} g^{-r'} h^{-\mathcal{H}(c)} = \delta$.

The other property (2) is immediately clear from the fact that the shop identity included in $I_{\mathcal{S}}$ differs per shop and $\overline{\mathcal{S}}$ does not use the same value for $I_{\mathcal{S}}$ in two different payments, since the verification relation that is applied by $\overline{\mathcal{B}}$ in the deposit scheme is the same as that applied by $\overline{\mathcal{S}}$ in the payment scheme.

■

5.6.2 Privacy

We can see that the statements of the lemma and the proposition in the previous chapter also hold for *new e-cash system 2*.

Lemma 2 *For any $\overline{\mathcal{U}}$, for any possible view of $\widetilde{\mathcal{B}}$ in an execution of the withdrawal scheme in which $\overline{\mathcal{U}}$ accepts and for any possible view of $\widetilde{\mathcal{S}}$ in an execution of the payment scheme in which the payer follows the scheme, there is exactly one set of random choices that $\overline{\mathcal{U}}$ could have made in the execution of the withdrawal scheme such that the views of $\widetilde{\mathcal{B}}$ and $\widetilde{\mathcal{S}}$ correspond to the withdrawal and payment of the same e-cash.*

Proof. We define the following sets:

- Viewer($\widetilde{\mathcal{B}}$) = $\{(c,\delta,r',s') \mid \delta \in \mathbb{G}_q$ and $r', s' \in \mathbb{Z}_q$ such that $(h_1^u h_2)^{s'} g^{-r'} h^{-\mathcal{H}(c)} = (h_1^u h_2)^k = \delta\}$
- Viewer($\widetilde{\mathcal{S}}$) = $\{(\alpha, c, r, s, d, r_1, r_2) \mid \alpha, r \in \mathbb{G}_q$, $d \in \{0,1\}^\ell$ and $s, r_1, r_2 \in \mathbb{Z}_q$ such that $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}\}$
- Choices($\mathcal{U}$) = $\{(a, b, y, z_1, z_2) \mid a, b, z_1, z_2 \in \mathbb{Z}_q$ and $y \in \mathbb{Z}_q^*\}$

We have only to show that for all $\widetilde{\mathcal{B}}$-view $\in$ Views($\widetilde{\mathcal{B}}$) and for all $\widetilde{\mathcal{S}}$-view $\in$ Views($\widetilde{\mathcal{S}}$), there is exactly one tuple $(a, b, y, z_1, z_2) \in$ Choices($\mathcal{U}$) such that $\widetilde{\mathcal{B}}$-view and $\widetilde{\mathcal{S}}$-view correspond to the withdrawal and payment of the same e-cash.

First, y is uniquely determined from α and $v = h_1^u h_2$ as $y = \log_v \alpha$. From r_1, u, d and y, we see that the choice $z_1 = r_1 - udy \pmod q$ must have been made, and from r_2, d and y, it follows that $z_2 = r_2 - dy \pmod q$ must have been chosen. The choice r together with r' determines a as $a = r' - r \pmod q$. Finally, the numbers s, s' and y determine b as $b = s - s'y^{-1} \pmod q$.

For these choices of the five variables, all the assignments and verifications in the two schemes executions would be satisfied by definition, except for the assignments $m = h_1^{z_1} h_2^{z_2} (= \alpha^{-s} g^r r h^{\mathcal{H}(c)})$ and $r = m g^a \alpha^b \delta$ that must have been made by $\mathcal{U}$ in the withdrawal scheme. To prove that these assignments hold as well, we notice that from $\widetilde{\mathcal{S}}$-view $\in$ Views($\widetilde{\mathcal{S}}$) we have that

$$h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}.$$

Therefore, the proof is completed if

$$\alpha^s g^{-r} h^{-\mathcal{H}(c)} = g^a \alpha^b \delta$$

and

$$h_1^{r_1} h_2^{r_2} \alpha^{-d} = h_1^{z_1} h_2^{z_2}$$

for (a, b, y, z_1, z_2) made above. This is obvious in the proof of *proposition 4*, considering that in the case the substitution in $(*)$ is allowed because $\widetilde{\mathcal{B}}$-view $\in$ Views($\widetilde{\mathcal{B}}$).

■

Proposition 5 *New e-cash system 2 protects the privacy of the user in the payment.*

Proof. This is an immediate consequence of *lemma 2* and the fact that $\mathcal{U}$ in the withdrawal scheme generates (a, b, y, z_1, z_2) uniformly at random from Choices($\mathcal{U}$).

■

5.6.3 Forgery

To forge a coin, the verification equation, $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$ $(= m)$, must be satisfied. We consider that the attacks realizing the forgery are the same as *new e-cash system 1.*

Proposition 6 *Illegal users cannot forge a coin in new e-cash system 2.*

Proof.

[Attack 1]

In this attack, some users must make a coin only from the verification equation, $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$. First of all, considering $h_1^{r_1} h_2^{r_2} \alpha^{-d} = m$, since $\mathcal{U}$ cannot know $\mathcal{B}$'s private keys (x, x_1, x_2) because of the difficulty of the discrete logarithm problem, some users should determine α as $\alpha = h_1^{\varepsilon_1} h_2^{\varepsilon_2}$, where $\varepsilon_1 \neq 0$ and $\varepsilon_2 \neq 0$. Since δ, g^a, α^b and $h^{\mathcal{H}(c)}$ are quite independent of r and s, from

$$\alpha^s g^{-r} h^{-\mathcal{H}(c)} = g^a \alpha^b \delta,$$

some users can obtain the following equation:

$$(h_1^{\varepsilon_1} h_2^{\varepsilon_2})^s g^{-r} = g^D,$$

where $g^D = g^a \alpha^b \delta h^{\mathcal{H}(c)}$. However, as

$$s = \frac{D + r}{\varepsilon_1 x_1 + \varepsilon_2 x_2},$$

the relationship between r and s requires $\mathcal{B}$'s private keys (x_1, x_2).

[Attack 2]

This is the attack that some users make a coin by mixing two (or more) different coins. Now, we suppose that two users $\mathcal{U}_A$ and $\mathcal{U}_B$ have two coins M_A and M_B, respectively, where $M_i = [\alpha_i, c_i, m_{AB}, r_i, s_i]$ $(i = A, B)$. Assuming that

$$\begin{aligned} r &= mg^{-r}\alpha^s h^{-\mathcal{H}(c)} \\ &= (\mu_1+\mu_2)m_{AB}g^{-\mu_1 r_A-\mu_2 r_B}\alpha^{\mu_1 s_A+\mu_2 s_B}h^{-\mathcal{H}(\mu_1 c_A+\mu_2 c_B)}, \end{aligned}$$

where $\mu_1 \neq 0$ and $\mu_2 \neq 0$, $\mathcal{U}_A$ and $\mathcal{U}_B$ wish to satisfy the equation, $r = \mu_1 r_A + \mu_2 r_B$. However, since

$$\mu_1 r_A + \mu_2 r_B = m_{AB}(\mu_1 g^{-r_A}\alpha_A^{s_A}h^{-\mathcal{H}(c_A)} + \mu_2 g^{-r_B}\alpha_B^{s_B}h^{-\mathcal{H}(c_B)}),$$

we see that generally $r \neq \mu_1 r_A + \mu_2 r_B$.

[Attack 3]

As $\mathcal{B}$'s signature s' contains $\dfrac{r'}{ux_1+x_2}$, it is impossible for $\mathcal{U}$ to frame up the user identity without $\mathcal{B}$'s private keys (x_1, x_2) in the withdrawal. Now, we consider the forgery of the coin value. In the withdrawal scheme, $\mathcal{U}$ computes $r = mg^a\alpha^b\delta^\mu$, where $\mu \neq 0$, and then sends $r' = (r+a)\mu^{-1} \pmod q$ to $\mathcal{B}$. After getting $s' = \dfrac{r'+\mathcal{H}(c)x}{ux_1+x_2} + k \pmod q$, $\mathcal{U}$ calculates $s = s'\mu y^{-1} + b \pmod q$, and then verifies the following equation:

$$\begin{aligned} m &= \alpha^{-s}g^r r h^{\mathcal{H}(\mu c)} \\ &= g^{-(r+a)}h^{-\mu\mathcal{H}(c)}\delta^{-\mu}\alpha^{-b}\cdot g^r \cdot mg^a\alpha^b\delta^\mu \cdot h^{\mathcal{H}(\mu c)} \\ &= mh^{\mathcal{H}(\mu c)-\mu\mathcal{H}(c)}. \end{aligned}$$

However, we see that generally $\mathcal{H}(\mu c) \neq \mu\mathcal{H}(c)$.

[Attack 4]

Now, we suppose that two users $\mathcal{U}_A$ and $\mathcal{U}_B$ perform the withdrawal scheme in parallel. First, $\mathcal{U}_A$ and $\mathcal{U}_B$ get (c_A, δ_A) and (c_B, δ_B), respectively, where $\delta_i = (h_1^{u_i}h_2)^{k_i}$ $(i = A, B)$. Assuming that u includes u_A and u_B, they compute $\alpha = (h_1^u h_2)^y$ and $r = mg^a\alpha^b\delta_A^\mu\delta_B$, where $\mu \neq 0$. They send ${r'}_A = (r+a)(2\mu)^{-1} \pmod q$ and ${r'}_B = (r+a)2^{-1} \pmod q$, respectively. Getting $({s'}_A, {s'}_B)$, respectively, where ${s'}_i = \dfrac{{r'}_i+\mathcal{H}(c_i)x}{u_i x_1+x_2} + k_i \pmod q$ $(i =$

A, B), they calculate $s = (\mu s'_A + s'_B)y^{-1} + b \pmod q$, and then confirm the following equation:

$$\begin{aligned}
m &= \alpha^{-s} g^r r h^{\mathcal{H}(\mu c_A + c_B)} \\
&= (h_1^u h_2)^{\frac{-(r+a)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{-(r+a)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2} - (\mu k_A + k_B)} \alpha^{-b} \cdot g^r \cdot m g^a \alpha^b \delta_A^\mu \delta_B \cdot h^{\mathcal{H}(\mu c_A + c_B)} \\
&= m \cdot g^{r+a} h^{\mathcal{H}(\mu c_A + c_B)} (h_1^u h_2)^{\frac{-(r+a)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{-(r+a)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2} - (\mu k_A + k_B)} \delta_A^\mu \delta_B \\
&= m \cdot g^{r+a+\mathcal{H}(\mu c_A + c_B)x} (h_1^u h_2)^{\frac{-(r+a)2^{-1} - \mu\mathcal{H}(c_A)x}{u_A x_1 + x_2} + \frac{-(r+a)2^{-1} - \mathcal{H}(c_B)x}{u_B x_1 + x_2}} h_1^{\mu k_A u_A + k_B u_B - u(\mu k_A + k_B)}.
\end{aligned}$$

Then, the equation:

$$g^\theta (h_1^u h_2)^{\frac{\theta_1}{u_A x_1 + x_2} + \frac{\theta_2}{u_B x_1 + x_2}} h_1^{\mu k_A u_A + k_B u_B - u(\mu k_A + k_B)} = 1,$$

where

$$\begin{aligned}
\theta &= r + a + \mathcal{H}(\mu c_A + c_B)x; \\
\theta_1 &= -(r+a)2^{-1} - \mu\mathcal{H}(c_A)x; \\
\theta_2 &= -(r+a)2^{-1} - \mathcal{H}(c_B)x,
\end{aligned}$$

must be satisfied. Therefore, they can obtain the following equations:

$$\begin{cases}
u_A u_B \theta + u(u_A \theta_2 + u_B \theta_1) = 0; \\
u(\theta_1 + \theta_2) + u_A(\theta + \theta_2) + u_B(\theta + \theta_1) = 0; \\
\theta + \theta_1 + \theta_2 = 0; \\
\mu k_A u_A u_B (u - u_A) + k_B u_A u_B (u - u_B) = 0; \\
\mu k_A (u_A + u_B)(u - u_A) + k_B (u_A + u_B)(u - u_B) = 0; \\
\mu k_A (u - u_A) + k_B (u - u_B) = 0.
\end{cases}$$

However, it is possible to satisfy these equations only if $u_A = u_B$.

■

5.6.4 Double-spending Detection

If $\mathcal{U}$ has double-spent a coin, $\mathcal{B}$ will be able to obtain the responses (r_1, r_2) and (r'_1, r'_2) for two different challenges d and d', where $r_1 = z_1 + udy \pmod q$, $r_2 = z_2 + dy \pmod q$, $r'_1 = z_1 + ud'y \pmod q$ and $r'_2 = z_2 + d'y \pmod q$. Then, $\mathcal{B}$ can compute

$$\begin{aligned}
r_1 - r'_1 &= u(d - d')y; \\
r_2 - r'_2 &= (d - d')y.
\end{aligned}$$

From $u(d-d')y$ and $(d-d')y$, $\mathcal{B}$ can easily obtain u. Therefore, $\mathcal{B}$ can determine the double-spender.

5.7 Performance Evaluation

In the previous chapter, we have understood that *new e-cash system 1* is more efficient than the e-cash systems in [3,11,12]. Now, we compare *new e-cash system 2* with *new e-cash system 1* from the viewpoint of *efficiency*. When we assume in both systems, $|p| = 1024$, $|q| = 160$, $|c| = 160$, we obtain the following results on Table 5.1.

	Communication Amount [bits]		Number of Exponentiation			
	Withdrawal	Payment	Withdrawal		Payment	
			$\mathcal{U}$	$\mathcal{B}$	$\mathcal{U}$	$\mathcal{S}$
New E-cash System 1	1504	2944	10	3	0	6
New E-cash System 2	1504	1920	10	3	0	6

Table 5.1: Comparison between Proposed E-cash Systems

Compared with *new e-cash system 1*, we see that *new e-cash system 2* improves the communication amount in the payment. In other words, *new e-cash system 2* decreases the communication cost, and is more efficient. This is because $\mathcal{S}$ can recover m even if $\mathcal{U}$ do not send m to $\mathcal{S}$. However, we will consider that *new e-cash system 1* is a little securer than *new e-cash system 2* in point of accidental attacks.

Chapter 6

Conclusion

In this paper, we have considered the actual problems in the e-cash systems [14,15], and then proposed two new *untraceable off-line* e-cash systems. One is *new e-cash system 1* using the two blind signature schemes presented in [2,22], which are based on Schnorr signature scheme [21], and is the system made by improving [13]. The other is *new e-cash system 2* using the property of Nyberg-Rueppel signature [16,17], which provides *message recovery*. In addition, we have estimated the security of the two proposed e-cash systems, which consists of *completeness*, *privacy*, *forgery* and *double-spending detection*. Our e-cash systems are more efficient than other e-cash systems [3,11,12].

References

[1] M.Abe, E.Fujisaki, "How to Date Blind Signatures", LNCS 1163, Advances in Cryptology - ASIACRYPT '96, Springer-Verlag, pp.244-251, 1996.

[2] M.Abe, J.Camenisch, "Partially Blind Signature Schemes", Proc. of the 1997 SCIS, SCIS'97-33D, 1997.

[3] S.Brands, "Untraceable Off-line Cash in Wallet with Observers", LNCS 773, Advances in Cryptology - CRYPTO '93, Springer-Verlag, pp.302-318, 1994.

[4] S.Brands, "Off-line electronic cash based on secret-key certificates", Technical Report, CWI, ftp://ftp.cwi.nl:/pub/CWIreports/AA/CS-R9506.ps.Z, 1995.

[5] S.Brands, "A note on parallel executions of restrictive blind issuing protocols for secret-key certificates", Technical Report, CWI, ftp://ftp.cwi.nl:/pub/CWIreports /AA/CS-R9519.ps.Z, 1995.

[6] D.Chaum, "Security without Identification: Transaction Systems to Make Big Brother Obsolete", Comm. of the ACM, 28, 10, pp.1030-1044, 1985.

[7] D.Chaum, A.Fiat, M.Naor, "Untraceable Electronic Cash", LNCS 403, Advances in Cryptology - CRYPTO '88, Springer-Verlag, pp.319-327, 1988.

[8] I.B.Damgård, "Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals", LNCS 403, Advances in Cryptology - CRYPTO '88, Springer-Verlag, pp.328-335, 1988.

[9] T.ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm", IEEE Trans. inform. Theory, IT-31(4), July, pp.469-472, 1985.

[10] U.Feige, A.Fiat, A.Shamir, "Zero Knowledge Proofs of Identity", Proc. the 19th Annual ACM Symposium on Theory of Computing, pp.210-217, 1987.

[11] N.Ferguson, "Single Term Off-line Coins", LNCS 765, Advances in Cryptology - EUROCRYPT '93, Springer-Verlag, pp.318-328, 1994.

[12] N.Ferguson, "Extensions of Single-term Coins", LNCS 773, Advances in Cryptology - CRYPTO '93, Springer-Verlag, pp.292-301, 1994.

[13] K.Hirohashi, M.Tada, E.Okamoto, "Study on a new e-cash system using two blind signatures", Proc. of the 1999 SCIS, pp.365-370, 1999.

[14] S.Miyazaki, K.Sakurai, "A method of embedding certificates in untraceable electronic money", Proc. of the 1998 SCIS, SCIS'98-3.4.C, 1998.

[15] K.Q.Nguyen, Y.Mu, V.Varadharajan, "A New Digital Cash Scheme Based on Blind Nyberg-Rueppel Digital Signature", LNCS 1396, Information Security, Springer-Verlag, pp.313-320, 1998.

[16] K.Nyberg, R.A.Rueppel, "A New Signature Scheme Based on the DSA Giving Message Recovery", 1st ACM Conference on Computer and Communications Security, November 3-5, pp.58-61, 1993.

[17] K.Nyberg, R.A.Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem", LNCS 950, Advances in Cryptology - EUROCRYPT '94, Springer-Verlag, pp.182-193, 1995.

[18] T.Okamoto, K.Ohta, "Universal Electronic Cash", LNCS 576, Advances in Cryptology - CRYPTO '91, Springer-Verlag, pp.324-337, 1992.

[19] T.Okamoto, "An Efficient Divisible Electronic Cash Scheme", LNCS 963, Advances in Cryptology - CRYPTO '95, Springer-Verlag, pp.438-451, 1995.

[20] B.Pfitzmann, M.WAidner, "How to Break and Repair a "Provably Secure" Untraceable Payment Syatem", LNCS 576, Advances in Cryptology - CRYPTO '91, Springer-Verlag, pp.338-350, 1992.

[21] C.P.Schnorr, "Efficient Signature Generation by Smart Cards", Journal of CRYPTOLOGY, 4(3), pp.161-174, 1991.

[22] B.Schoenmakers, "An Efficient Electronic Payment System Withstanding Parrallel Attacks", Technical Report, CWI, ftp://ftp.cwi.nl:/pub/CWIreports/AA/CS-R9522.ps.Z, 1995.

Publications

[1] K.Hirohashi, M.Tada, E.Okamoto, "Study on a new e-cash system using two blind signatures", Proc. of the 1999 SCIS, pp.365-370, 1999.

[2] K.Hirohashi, M.Tada, E.Okamoto, "Proposal for New E-cash System with Message Recovery", Submitting to IEICE Trans. Fundamentals.