Regular Paper

Proposal for New E-cash System Using Message Recovery Signature

Koji Hirohashi,† Mitsuru Tada†† and Eiji Okamoto†††

In this paper, we have examined the actual problem in the e-cash system (Nguyen, et al. 1998), and then proposed a new *untraceable off-line* e-cash system with the feature of Nyberg-Rueppel Signature (1993; 1995), which provides *message recovery*. Moreover, we have estimated the security in our e-cash system 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 system is more efficient than other e-cash systems (Brands 1994; Ferguson 1994).

1. Introduction

Electronic cash systems (e-cash systems) have become one of the most important researches from both practical and theoretical viewpoints. In e-cash systems, 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 Chaum [4], Damgård [6] and Pfitzmann, et al.[17]. 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 [1),2),5),9)~11),15),16),19)]. In Brands [2),3)], the e-cash system [2)] allows the attacker to forge coins by executions of the scheme in parallel. In other words, this system is weak for *parallel attack*. In Ferguson's [9),10)], the withdrawal scheme is not efficient because of enormous communication cost. The e-cash systems [15),16)] realize the dividability that a coin can be subdivided into many pieces. However, the e-cash system [15)] 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 e-cash system [16)] does not realize the unlinkability among coins divided from the same coin.

Nyberg and Rueppel [13)] 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[7)] and Schnorr[19)] signature schemes, cannot realize this property.

Utilizing the feature of this signature, Nguyen, et al.[12)] proposed the e-cash system with *message recovery* unlike the previous e-cash systems [1),2),5),9)~11),15),16),19)]. However, this e-cash system allows anyone to forge coins. The reason is that a user can make the coin

† PFU Limited
†† School of Information Science, Japan Advanced Institute of Science and Technology
††† University of Wisconsin, Milwaukee

satisfying the verification equations, even if he does not know the private keys a bank uses in the withdrawal scheme.

In this paper, we will first consider the actual problem in the e-cash system [12]. Secondly, we will propose a new *untraceable off-line* e-cash system with the property of Nyberg-Rueppel signature, which provides *message recovery*. In addition, we will estimate the security in the e-cash system.

2. Nyberg-Rueppel Signature Scheme

Here we review the signature scheme which we use in this paper, and which is a special case of Nyberg-Rueppel Signature Scheme [13),14)]. The system parameters consist of two primes p and q, such that $q|(p-1)$, and an element $g \in \mathbf{Z}_p^*$ whose order is q. (Since most operations are executed under modulo p, we will often omit the description of $\pmod p$, if that omission may not cause any confusion.) The signer's private key is $x \in \mathbf{Z}_q$, while the corresponding public key is $h := g^x$. To sign a message $m \in \mathbf{Z}_p$, the signer selects $k \in \mathbf{Z}_q$ at random, and then computes r and s such that

$$r := mg^{-k};$$
$$k = r' + sx \pmod q,$$

where $r' := r \pmod q$. The pair (r, s) turns out to be the signature for the message m. The message can be recovered by computing a verification equation:

$$m = g^r h^s r.$$

3. E-cash System [12)]

3.1 Preparation

Let p, q and g be two primes such that $q|(p-1)$ and an order-q element in $\mathbf{Z}_p^*$, respectively. Then, we suppose those are public. The bank $\mathcal{B}$ 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.

The user $\mathcal{U}$ 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 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:

Step 1. $\mathcal{B}$ chooses a random number $k \in \mathbf{Z}_q$, and then transfers $\delta := v^k$ to $\mathcal{U}$.

Step 2. $\mathcal{U}$ randomly generates $y, z_1, z_2 \in \mathbf{Z}_q^*$, and then computes $\alpha := w^y$, $\beta := v^y$ and $\lambda := h_1^{z_1} h_2^{z_2}$.

Step 3. Using a strong one-way hash function $\mathcal{H}$, $\mathcal{U}$ forms the message $m := \mathcal{H}(\alpha, \beta, \lambda)$, generates $a, b \in \mathbf{Z}_q^*$ at random, calculates $r := m\beta^a \delta^{by}$, and then sends $r' := rb^{-1} \pmod q$ to $\mathcal{B}$.

Step 4. $\mathcal{B}$ sends $s' := r'x + k \pmod q$ to $\mathcal{U}$.

Step 5. $\mathcal{U}$ removes the blind factor b, and then obtains $s := s'b + a \pmod q$.

Step 6. $\mathcal{U}$ verifies the validity of the coin by using the equation, $\mathcal{H}(\alpha, \beta, \lambda) = \beta^{-s}\alpha^r r$.

3.3 Payment Scheme

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

Step 1. $\mathcal{S}$ sends $d := \mathcal{H}(\mathcal{S} \parallel \texttt{Date} \parallel \texttt{Time} \parallel \cdots)$ to $\mathcal{U}$.

Step 2. $\mathcal{U}$ 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 $\mathcal{S}$.

Step 3. $\mathcal{S}$ 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.

3.4 Deposit Scheme

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

Step 1. $\mathcal{S}$ sends the payment transcript (M, d, r_1, r_2) to $\mathcal{B}$.

Step 2. $\mathcal{B}$ 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.

4. Problem in the E-cash System [12)]

In this system, 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.

Although Nguyen, et al.[12)] insist that we have to find how to frame up another valid signature from a certain one to forge a coin, we can give another way for forgery.

Now, we show the attack on the e-cash system [12)]. In the withdrawal, since α and β are the information which $\mathcal{B}$ do not know, it is pos-

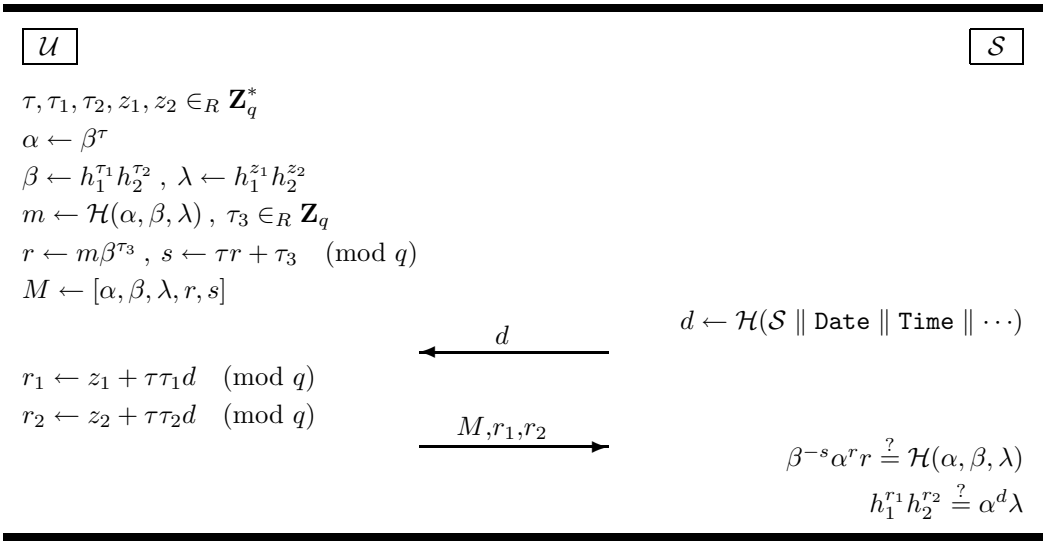

Fig. 1 Attack on the e-cash system[12].

sible 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 **Fig. 1**.

5. New E-cash System

In this section, we propose a new e-cash system using Nyberg-Rueppel Signature[13),14)], which provides *message recovery*.

5.1 System Setup

Let p and q be primes which satisfy $q|(p-1)$. We suppose both are public. Moreover, we suppose $g \in \mathbf{G}_q \backslash \{1\}$ is also public, where $\mathbf{G}_q$ is a subgroup of $\mathbf{Z}_p^*$ consisting of order-q elements. $\mathcal{H}$ is a strong one-way hash function mapping from $\{0,1\}^*$ to $\{0,1\}^\ell$ ($\ell \approx 160$). Let $\parallel$ denote concatenation. $\mathcal{B}$ generates three private keys x, x_1, $x_2 \in \mathbf{Z}_q^*$, 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) $u \in \mathbf{Z}_q^*$ 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 share the user identity u with $\mathcal{B}$.

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 (see **Fig. 2**):

Step 1. $\mathcal{B}$ generates a random number $k \in \mathbf{Z}_q$, and then sends $\delta := (h_1^u h_2)^k$ to $\mathcal{U}$. Moreover $\mathcal{B}$ generates the coin information c, which consists of value, expiration date, possibly some random bits and so on. But as mentioned later, $\mathcal{B}$ must not send c to $\mathcal{U}$ at this step.

Step 2. After choosing $y \in \mathbf{Z}_q^*$ at random, $\mathcal{U}$ calculates $\alpha := (h_1^u h_2)^y$. $\mathcal{U}$ also generates four random numbers $a, b, z_1, z_2 \in \mathbf{Z}_q$, and then computes $m := h_1^{z_1} h_2^{z_2}$ and $r := mg^a \alpha^b \delta$.

Step 3. $\mathcal{U}$ sends $r' := r + a \pmod q$ to $\mathcal{B}$.

Step 4. $\mathcal{B}$ sends $s' := \dfrac{r' + \mathcal{H}(c)x}{ux_1 + x_2} + k \pmod q$ and c generated at Step 1, to $\mathcal{U}$.

Step 5. $\mathcal{U}$ computes $s := s'y^{-1} + b \pmod q$.

Step 6. $\mathcal{U}$ 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 (see

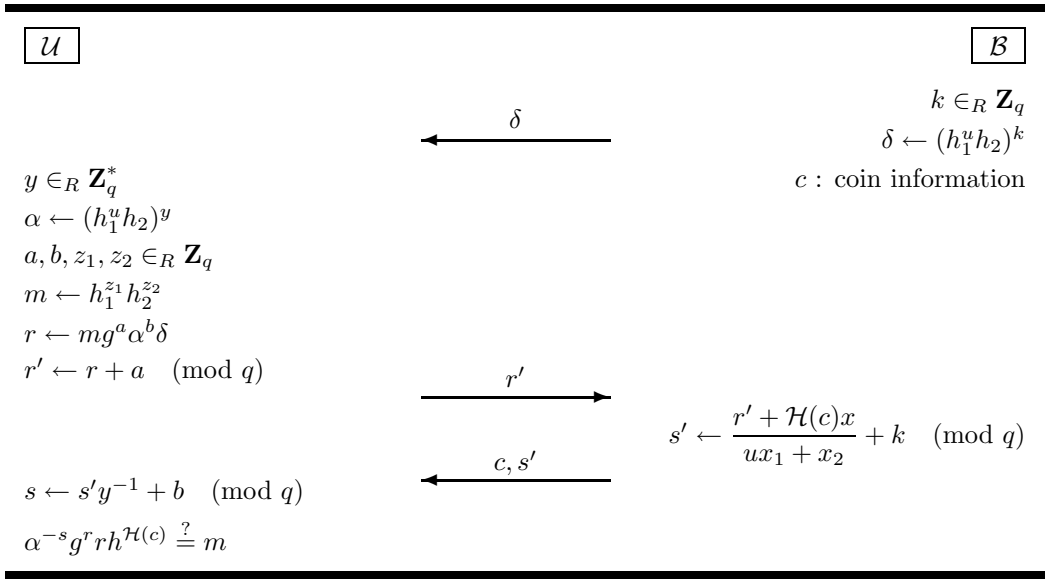

Fig. 2 Withdrawal scheme.

Fig. 3):

Step 1. $\mathcal{U}$ transfers the coin M to $\mathcal{S}$.

Step 2. $\mathcal{S}$ 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*.

Step 3. $\mathcal{U}$ sends the response (r_1, r_2), where $r_1 := z_1 + udy \pmod q$ and $r_2 := z_2 + dy \pmod q$, to $\mathcal{S}$.

Step 4. $\mathcal{S}$ accepts if and only if the verification, $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$ is successful.

5.5 Deposit Scheme

When $\mathcal{S}$ wants to deposit the coin M at $\mathcal{B}$, the following scheme is run (see **Fig. 4**):

Step 1. $\mathcal{S}$ sends the payment transcript $(M, I_{\mathcal{S}}, r_1, r_2)$ to $\mathcal{B}$.

Step 2. $\mathcal{B}$ computes $d = \mathcal{H}(M \| I_{\mathcal{S}})$.

Step 3. $\mathcal{B}$ accepts if and only if the verification, $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$ is successful.

6. Security

This section is due to Brands [2] and Schoenmakers [19] to some extent. Following Feige, et al.[8], 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.

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 is complete.

Proof.

The proof will be given in Appendix. ■

6.2 Privacy

We say that 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.

To prove the user's privacy in the payment, we show the following lemma:

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

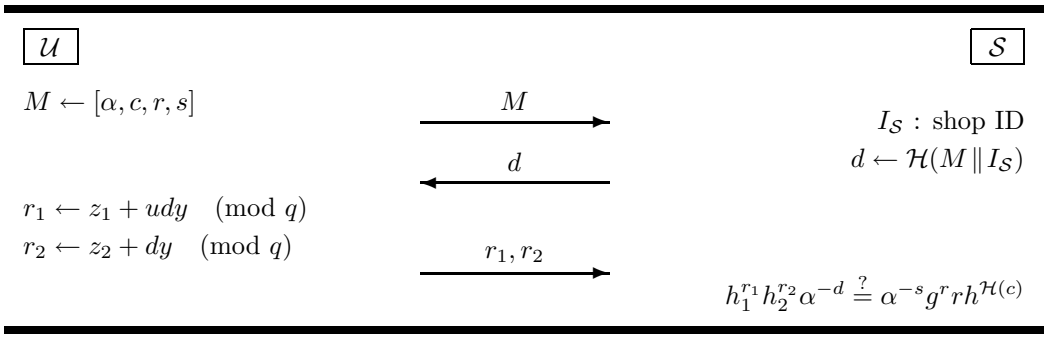

Fig. 3 Payment scheme.

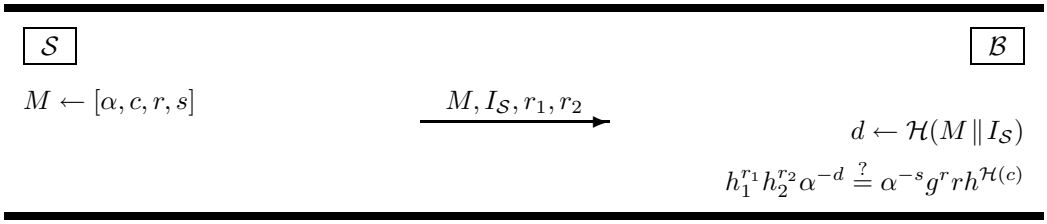

Fig. 4 Deposit scheme.

such that the views of $\widetilde{\mathcal{B}}$ and $\widetilde{\mathcal{S}}$ correspond to the withdrawal and payment of the same e-cash.

Proof.
The proof will be given in Appendix. ■

Proposition 3 New e-cash system protects the privacy of the user in the payment.

Proof.
The proof will be given in Appendix. ■

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 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 4 Illegal users cannot forge a coin in the new e-cash system.

Proof.
The proof will be given in Appendix. ■

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.

7. Performance Evaluation

In this section, we compare the efficiency of *new e-cash system* with that of the off-line e-cash systems[1),9),10)], which are famous and secure. The efficiency of e-cash systems is estimated by the cost of communication and computation. We suppose that the computation cost depends on the number of exponentiations in each scheme, and that the communication cost relies on the communication amount of parameters in each scheme. Now, we assume in Brands scheme[1)], $|p| = 1024$, $|q| = 160$, in Ferguson scheme[9),10)], $|n| = 1024$, $|v| = 160$, and in our system, $|p| = 1024$, $|q| = 160$, $|c| = 160$, where $|\cdot|$ denotes the binary length. Then, we get the following results on **Table 1**.

Table 1 Comparison between new e-cash system and other e-cash systems[1),9),10)].

	Communication Amount [bits]		Number of Exponentiation			
	Withdrawal	Payment	Withdrawal		Payment	
			$\mathcal{U}$	$\mathcal{B}$	$\mathcal{U}$	$\mathcal{S}$
Brands system[1)]	2368	5760	12	2	0	7
Ferguson system[9),10)]	10880	4416	19	9	5	8
New e-cash system	1504	1952	9	2	0	6

In the withdrawal, the communication amount of our system is smaller than those of Ferguson and Brands systems. Moreover, the number of exponentiations imposed on $\mathcal{U}$ is also smaller than those in any other systems[1),9),10)]. The number of exponentiations 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 those of other e-cash systems[1),9),10)]. In our system and Brands system, $\mathcal{U}$ do not need exponentiations. The number of exponentiations imposed on $\mathcal{S}$ is smaller than those of Ferguson and Brands systems.

Therefore, we see that *new e-cash system* is more efficient than other e-cash systems[1),9),10)].

8. Conclusion

In this paper, we have considered the actual problem in the e-cash system[12)], and then proposed a new *untraceable off-line* e-cash system using the property of Nyberg-Rueppel Signature[13),14)], which provides *message recovery*. In addition, we have estimated the security in the the proposed e-cash system, which consists of *completeness*, *privacy*, *forgery* and *double-spending detection*. Our e-cash system is more efficient than other e-cash systems[1),9),10)].

References

1) Brands, S.: Untraceable Off-line Cash in Wallet with Observers, *Advances in Cryptology – CRYPTO'93*, LNCS, Vol.773, pp.302–318, Springer-Verlag (1994).
2) Brands, S.: Off-line electronic cash based on secret-key certificates, Technical Report, CWI, CS-R9506.ps.Z (1995).
3) Brands, S.: A note on parallel executions of restrictive blind issuing protocols for secret-key certificates, Technical Report, CWI, CS-R9519.ps.Z (1995).
4) Chaum, D.: Security without Identification: Transaction Systems to Make Big Brother Obsolete, *Comm. ACM*, Vol.28, No.10, pp.1030–1044 (1985).
5) Chaum, D., Fiat, A. and Naor, M.: Untraceable Electronic Cash, *Advances in Cryptology – CRYPTO'88*, LNCS, Vol.403, pp.319–327, Springer-Verlag (1988).
6) Damgård, I.B.: Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals, *Advances in Cryptology – CRYPTO'88*, LNCS, Vol.403, pp.328–335, Springer-Verlag (1988).
7) ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm, *IEEE Trans. Inf. Theory*, Vol.31, No.4, pp.469–472 (1985).
8) Feige, U., Fiat, A. and Shamir, A.: Zero Knowledge Proofs of Identity, *Proc. 19th Annual ACM Symposium on Theory of Computing*, pp.210–217 (1987).
9) Ferguson, N.: Single Term Off-line Coins, *Advances in Cryptology – EUROCRYPT'93*, LNCS, Vol.765, pp.318–328, Springer-Verlag (1994).
10) Ferguson, N.: Extensions of Single-term Coins, *Advances in Cryptology – CRYPTO'93*, LNCS, Vol.773, pp.292–301, Springer-Verlag (1994).
11) Hirohashi, K., Tada, M. and Okamoto, E.: Study on a new e-cash system using two blind signatures, *Proc. 1999 Symposium on Cryptography and Information Security* (*SCIS'99*), pp.365–370 (1999).
12) Nguyen, K.Q., Mu, Y. and Varadharajan, V.: A New Digital Cash Scheme Based on Blind Nyberg-Rueppel Digital Signature, *Information Security*, LNCS, Vol.1396, pp.313–320, Springer-Verlag (1998).
13) Nyberg, K. and Rueppel, R.A.: A New Signature Scheme Based on the DSA Giving Message Recovery, *1st ACM Conference on Computer and Communications Security*, Nov. 3–5, pp.58–61 (1993).
14) Nyberg, K. and Rueppel, R.A.: Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem, *Advances in Cryptology – EUROCRYPT'94*, LNCS, Vol.950, pp.182–193, Springer-Verlag (1995).

15) Okamoto, T. and Ohta, K.: Universal Electronic Cash, *Advances in Cryptology – CRYPTO'91*, LNCS, Vol.576, pp.324–337, Springer-Verlag (1992).
16) Okamoto, T.: An Efficient Divisible Electronic Cash Scheme, *Advances in Cryptology – CRYPTO'95*, LNCS, Vol.963, pp.438–451, Springer-Verlag (1995).
17) Pfitzmann, B. and Waidner, M.: How to Break and Repair a "Provably Secure" Untraceable Payment System, *Advances in Cryptology – CRYPTO'91*, LNCS, Vol.576, pp.338–350, Springer-Verlag (1992).
18) Schnorr, C.P.: Efficient Signature Generation by Smart Cards, *J. Cryptology*, Vol.4, No.3, pp.161–174 (1991).
19) Schoenmakers, B.: An Efficient Electronic Payment System Withstanding Parrallel Attacks, Technical Report, CWI, CS-R9522.ps.Z (1995).

Appendix

Here we have the proofs of Proposition 1, Lemma 2 and Propositions 3 and 4.

Proposition 1 New e-cash system 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

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

and the second from

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

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. ■

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 first consider the relations that must be satisfied by the definition. The response s' of $\widetilde{\mathcal{B}}$ in the withdrawal scheme satisfies $(h_1^u h_2)^{s'} g^{-r'} h^{-\mathcal{H}(c)} = \delta$, since $\overline{\mathcal{U}}$ accepts in the withdrawal scheme. By Proposition 1, we can assume that the relation $h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}$ is satisfied in all views of $\widetilde{\mathcal{S}}$ in an executions of the payment scheme in which the payer follows the scheme.

We correspondingly define the following sets:

$$\begin{aligned}
&\mathrm{Views}(\widetilde{\mathcal{B}}) \\
&:= \{(c, \delta, r', s') | \delta \in \mathbf{G}_q \text{ and } r', s' \in \mathbf{Z}_q \\
&\quad \text{such that} \\
&\quad (h_1^u h_2)^{s'} g^{-r'} h^{-\mathcal{H}(c)} = (h_1^u h_2)^k = \delta\} ; \\
&\mathrm{Views}(\widetilde{\mathcal{S}}) \\
&:= \{(\alpha, c, r, s, d, r_1, r_2) | \alpha, r \in \mathbf{G}_q, \\
&\quad d \in \{0,1\}^\ell \text{ and } s, r_1, r_2 \in \mathbf{Z}_q \\
&\quad \text{such that} \\
&\quad h_1^{r_1} h_2^{r_2} \alpha^{-d} = \alpha^{-s} g^r r h^{\mathcal{H}(c)}\} ; \\
&\mathrm{Choices}(\mathcal{U}) \\
&:= \{(a, b, y, z_1, z_2) | a, b, z_1, z_2 \in \mathbf{Z}_q \\
&\quad \text{and } y \in \mathbf{Z}_q^*\} .
\end{aligned}$$

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 $r = mg^a \alpha^b \delta$ 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 1, considering that in the case the substitution in $(*)$ is allowed because $\widetilde{\mathcal{B}}$-view $\in$ Views$(\widetilde{\mathcal{B}})$. ■

Proposition 3 New e-cash system 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$(\mathcal{U})$. ■

Proposition 4 Illegal users cannot forge a coin in the new e-cash system.

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 $M_i = [\alpha_i, c_i, r_i, s_i]$ $(i = A, B)$. Assuming that

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

where $\mu_1 \neq 0$ and $\mu_2 \neq 0$, and where m_{AB} is a common m involved both in r_A and in r_B. In precise, $m = (\mu_1 + \mu_2) m_{AB}$. $\mathcal{U}_A$ and $\mathcal{U}_B$ wish to satisfy the equation, $r = \mu_1 r_A + \mu_2 r_B$. However, since

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

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:

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

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

If $\mathcal{B}$ sent the coin information c at Step 1 in our scheme, then $\mathcal{U}$ could forge a coin whose coin information is c', with pretending to withdraw a coin whose coin information is c. Let c' be the forged coin information. Then the forging scheme is as follows:

(1) Using c given at Step 1, $\mathcal{U}$ sets μ to be $\mathcal{H}(c')/\mathcal{H}(c)$.

(2) Then $\mathcal{U}$ computes $\widetilde{r} := mg^a\alpha^b\delta^\mu$, and sends $\widetilde{r'} := (\widetilde{r}+a)\mu^{-1} \pmod q$.

(3) After getting $\widetilde{s'} := \left(\widetilde{r'} + \mathcal{H}(c)x\right)/(ux_1 + x_2) + k \pmod q$, $\mathcal{U}$ figures out $\widetilde{s} := \widetilde{s'}y^{-1}\mu + b \pmod q$.

Then the coin $[\alpha, c', \widetilde{r}, \widetilde{s}]$ can satisfy the verification $\alpha^{-\widetilde{s}}g^{\widetilde{r}}\widetilde{r}h^{\mathcal{H}(c')} = m$, and is available both in the payment scheme and in the deposit scheme. Therefore $\mathcal{B}$ must not send c at Step 1.

[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, for example, $u = e_1u_A + e_2u_B$ for some constants e_1 and e_2, 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 $s'_i = \dfrac{r'_i + \mathcal{H}(c_i)x}{u_ix_1 + x_2} + k_i \pmod q$ $(i = A, B)$, they calculate $s = (\mu s'_A + s'_B)y^{-1} + b \pmod q$, and then confirm the following equation:

$$\begin{aligned} m &= \alpha^{-s}g^r r h^{\mathcal{H}(\mu c_A + c_B)} \\ &= mg^{r+a+\mathcal{H}(\mu c_A + c_B)x} \\ &\quad \cdot (h_1^u h_2)^{\frac{-(r+a)2^{-1} - \mu\mathcal{H}(c_A)x}{u_Ax_1+x_2} + \frac{-(r+a)2^{-1} - \mathcal{H}(c_B)x}{u_Bx_1+x_2}} \\ &\quad \cdot h_1^{\mu k_Au_A + k_Bu_B - u(\mu k_A + k_B)}. \end{aligned}$$

Then, the equation:

$$g^\theta (h_1^u h_2)^{\frac{\theta_1}{u_Ax_1+x_2} + \frac{\theta_2}{u_Bx_1+x_2}} \cdot h_1^{\mu k_Au_A + k_Bu_B - u(\mu k_A + k_B)} = 1,$$

where θ, θ_1 and θ_2 are defined as follows:

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

must be satisfied. Therefore, they can obtain the following equations by solving a quadratic identical equation with respect to x, x_1 and x_2:

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

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

(Received November 22, 1999)
(Accepted June 1, 2000)

Koji Hirohashi was born in Ishikawa, Japan, on January 9, 1974. He received the Batchelor's degree from Kanazawa University, Japan, in 1996 and the Master's degree from Japan Advanced Institute of Science and Technology (JAIST) in 1999. He joined PFU Limited since 1999. His interests are on Cryptology and Electronic Cash Systems.

Mitsuru Tada was born in Kyoto, Japan, on September 18, 1969. He received the B.S., M.I.S., and Dr.I.S. degrees from Tohoku University, Japan, in 1992, 1995, and 1998, respectively. He joined Japan Advanced Institute of Science and Technology (JAIST) in 1998, and is currently Associate of School of Information Science, JAIST. He received the SCIS paper award of IEICE in 2000. His interests are on Mathematical logic, Cryptology and Computational Complexity Theory. Dr. Tada is a member of IPSJ.

Eiji Okamoto received B.S., M.S. and Ph.D. degrees in electronics engineering at Tokyo Institute of Technology in 1973, 1975 and 1978, respectively. He worked for NEC from 1978 and studied communication theory and cryptography. From 1991, he was a professor of Information Science at JAIST, and from 1999 he is a professor at University of Wisconsin-Milwaukee and Toho University. He was a visiting professor of Mathematics at Texas A&M University, 1993–1994 and now he is a visiting professor at Chuo University. He received the best paper award of the IEICE in 1990 and the best author award of the IPSJ in 1993.