Unlinkable Divisible Electronic Cash

Toru Nakanishi and Yuji Sugiyama

Department of Communication Network Engineering,
Faculty of Engineering, Okayama University,
3-1-1 Tsushimanaka, Okayama 700-8530, Japan
{nakanisi,sugiyama}@cne.okayama-u.ac.jp

Abstract. Recently, some divisible electronic cash (e-cash) systems have been proposed. However, in existing divisible e-cash systems, efficiency or unlinkability is not sufficiently accomplished. In the existing efficient divisible cash systems, all protocols are conducted in the order of the polynomial of $\log N$ where N is the divisibility precision (i.e., (the total coin amount)/ (minimum divisible unit amount)), but payments divided from a coin are linkable (i.e., anyone can decide whether the payments are made by the same payer). The linked payments help anyone to trace the payer, if N is large. On the other hand, in the existing unlinkable divisible e-cash system, the protocols are conducted in the order of the polynomial of N, and thus it is inefficient for large N. In this paper, an unlinkable divisible e-cash system is proposed, where all protocols are conducted in the order of $(\log N)^2$.

Keywords: Electronic cash, Divisibility, Unlinkability, Group signature

1 Introduction

As the core to realizing the electronic commerce, the electronic cash (e-cash) is in great demand. In e-cash systems, a customer withdraws electronic *coins* from a bank, and the customer pays the coins to a shop in the *off-line* manner. The off-line means that the customer has no need to communicate with the bank or a trusted third party during the payment. Finally, the shop deposits the paid coins to the bank.

To protect the privacy of customers, each payment should be anonymous, and furthermore *unlinkability* should be satisfied. The unlinkability means that any other one except the trusted third party cannot determine whether two payments are made by the same customer. In linkable anonymous e-cash systems, the linked payments enable the other one to trace the payer by other means (i.e., correlating the payments' locality, date, frequency, etc.), as noted in [1].

In practice, it is desirable that e-cash systems are *divisible*, which means that payments of any amount up to the monetary amount of a withdrawn coin can be made. Hereafter, let N be (the total coin amount)/ (minimum divisible unit amount). N indicates the divisibility precision, and thus N needs to be large from the viewpoint of convenience. For example, when the total coin amount is \$1000 and the minimum divisible unit amount is 1 cent, N is about 2^{17}.

Therefore, the computational complexity for N is an important criterion in the divisible e-cash systems. In [2, 3], the efficient divisible e-cash systems where all protocols are conducted in $O(poly(\log N))$ are proposed, where *poly* means the polynomial. However, these systems in [2, 3] do not satisfy the unlinkability among the payments derived from the same coin. Thus, the larger N grows, the more easily the payer may be traced owing to the linked payments.

In [4], as a variant of divisible e-cash systems, an electronic coupon (e-coupon) system is proposed. In this system, a withdrawn coin, called a ticket, is divided into sub-tickets, and only the sub-tickets can be spent. The advantage of this system is to satisfy the unlinkability among all payments including ones from the same ticket, and thus both divisibility and unlinkability hold. On the other hand, the computational complexity requires $O(poly(N))$.

In this paper, a divisible e-cash system is proposed, where (1) the unlinkability among all payments holds, and (2) the computational complexity of all protocols is $O((\log N)^2)$. This e-cash system is based on the above e-coupon system. In the e-coupon system, the payment is accomplished by proving the ownership of a withdrawn ticket, which is the bank's digital signature, without revealing the ticket. Furthermore, to detect over-spending, the payer is forced to send values which are the same if and only if the payer uses the same sub-ticket. In the divisible e-cash system of this paper, the binary tree approach is adopted to realize $O(poly(\log N))$ computational complexity as well as the divisible e-cash systems [2, 3]. In this approach, a withdrawn coin has a binary tree, where the root represents the monetary amount of the coin and the other nodes represent the half of the amount of the parent node. In addition to the proof of the ownership of the coin as well as in the e-coupon system, the payer is forced to send values which are linked if and only if the nodes with the parent-child relationship are used for payments or the same node is used twice or more, which implies over-spending.

This paper is organized as follows: Section 2 describes a model and requirements for a divisible e-cash system. In Section 3, the binary tree approach and cryptographic primitives used in the proposed e-cash system are shown. In Section 4, a divisible e-cash system satisfying the requirements based on the model is proposed. Section 5 discusses the security and efficiency of the proposed system. Section 6 concludes this paper.

2 Model and Requirements

We adopt the model of "escrow cash" [5] to protect illegal acts of anonymous customers. In this model, trusted third parties, called trustees, participate in the system. The trustees cooperatively can revoke anonymity of payments to protect the illegal acts as money laundering, blackmailing attack [6] and so on. Though, in this paper, one trustee has the authority of the revocation for simplicity, it is easily extended into the the model of multiple trustees by using the threshold cryptosystems [7].

The requirements for divisible e-cash systems are as follows [2, 5]:

Unforgeability: A coin and a transcript of a payment can not be forged.

No over-spending: The customer who over-spends a coin can be identified.

No swindling: No one except the customer who withdraws a coin can spend the coin. The deposit information can not be forged.

Anonymity: No one except the payer and the trustee can trace the payer from the payment.

Unlinkability: No one except the payer and the trustee can determine whether any pair of payments is executed by the same customer, unless the payments cause over-spending.

Anonymity revocation: Anonymity of a transcript of a payment can be revoked only by the trustee and when necessary, where the following revocation procedures should be accomplished:

Owner tracing: To identify the payer of a targeted payment.

Coin tracing: To link a targeted withdrawal of a coin to the payments derived from the coin.

Only the transcript for which a judge's order is given must be de-anonymized.

Divisibility: Payments of any amount up to the monetary amount of a withdrawn coin can be made.

Off-line-ness: During payments, the payer communicates only with the shop.

3 Preliminaries

3.1 Binary Tree Approach

In the proposed e-cash system, the binary tree approach is adopted to accomplish the divisibility as well as the divisible e-cash systems in [2, 3]. Thus, before describing our system, we review this approach.

Each coin of $w = 2^{\ell-1}$ worth is assigned to a binary tree of ℓ levels. Each node of the tree is assigned to a denomination. The root node, denoted n_0, indicates the monetary amount w of the coin, and any other node $n_{j_1 \cdots j_u}$ $(2 \leq u \leq \ell)$ indicates half of the amount of the parent node $n_{j_1 \cdots j_{u-1}}$, where $j_1 = 0$ and $j_i \in \{0, 1\}$ for $i = 2, \ldots, u$. A binary tree of 3 levels is illustrated in Figure 1.

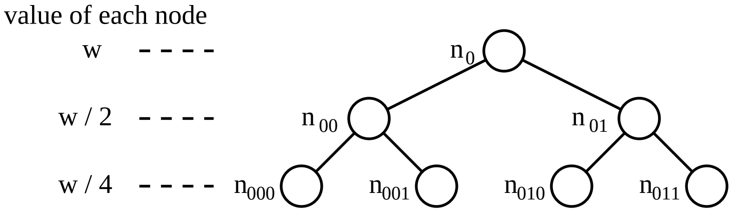

Fig. 1. A binary tree of 3 levels

To accomplish the divisibility, the following rule is used:

Divisibility rule: When a node is used, all descendant and ancestor nodes are not used. Furthermore, any node is not used multiple times.

This rule is satisfied if and only if over-spending is protected [2].

In the proposed system, each node $n_{j_1 \cdots j_u}$ possesses a value $F_{j_1 \cdots j_u}$, which is called F value. F value F_{j_1} of the root node is proper to the coin, and F value $F_{j_1 \cdots j_u}$ of any other node is applied by a one-way bijection for F value $F_{j_1 \cdots j_{u-1}}$ of the parent node. Thus, the nodes which have the parent-child relationship can be linked by a sequence of bijections, while the nodes without the relationship does not have such a sequence. Through this link, over-spending can be detected, and the over-spender can be identified by the trustee.

3.2 Signatures Based on Zero-knowledge Proofs of Knowledge

As well as the e-coupon system [4], the proposed e-cash system uses the extension of the group signature scheme of [8], where, as primitives to prove the knowledge of secret values without leaking any useful information, signatures based on zero-knowledge proofs of knowledge (SPK's) are used. Since the proposed system also uses some types of SPK's, this subsection reviews the SPK's. These are converted from zero-knowledge proofs of knowledge (PK's) by the so-called Fiat-Shamir heuristic [9]. That is, the prover determines the challenge by applying a collision-resistant hash-function to the commitment and the signed message and then computes the response as usual. The resulting signature consists of the challenge and the response. Such SPK's can be proven to be secure in the random oracle model [10] given the security of the underlying PK's. Let $SPK\{(\alpha, \beta, \ldots) : \mathit{Predicates}\}(m)$ be the signature on message m proving that the signer knows $\alpha, \beta, \ldots$ satisfying the proven predicates *Predicates*. In this notation, Greek letters denote the secret knowledge and the other letters denote public parameters between the signer and the verifier. In the proposed system as well as the group signature scheme [8] which are based on the hardness of the discrete logarithm problem, the relations among the discrete logarithms from cyclic groups are used as the proved predicates. In the following, let G and G_1 be cyclic groups with order q and q_1, respectively. The discrete logarithm of $y \in G$ to the base $z \in G$ is $x \in Z_q$ satisfying $y = z^x$ if such an x exists. We denote $x = \log_z y$. This is extended to the representation of $y \in G$ to the bases $z_1, z_2, \ldots z_k \in G$ which is $x_1, x_2, \ldots x_k \in Z_q$ satisfying $y = z_1^{x_1} \cdot z_2^{x_2} \cdots z_k^{x_k}$ if such x_i's exist. The double discrete logarithm of $y_1 \in G_1$ to the bases $z_1 \in G_1$ and $z \in G$ is $x \in Z_q$ satisfying $y_1 = z_1^{(z^x)}$ if such an x exists. The e-th root of the discrete logarithm of $y \in G$ to the base $z \in G$ is $x \in Z_q$ satisfying $y = z^{(x^e)}$ if such an x exists.

The first type of SPK is the signature proving the knowledge of representations of $y_1, \ldots, y_w \in G$ to the bases $z_1, \ldots, z_v \in G$ on message m, and it is denoted as

$$SPK\{(\alpha_1, \ldots, \alpha_u) : (y_1 = \textstyle\prod_{j=1}^{l_1} z_{b_{1j}}^{\alpha_{e_{1j}}}) \wedge \cdots \wedge (y_w = \textstyle\prod_{j=1}^{l_w} z_{b_{wj}}^{\alpha_{e_{wj}}})\}(m),$$

where constants $l_i \in \{1, \dots v\}$ indicate the number of bases on representation of y_i, the indices $e_{ij} \in \{1, \dots, u\}$ refer to the elements $\alpha_1, \dots, \alpha_u$ and the indices $b_{ij} \in \{1, \dots, v\}$ refer to the elements $z_1, \dots, z_v$. For example, $SPK\{(\alpha, \beta) : y_1 = z_1^{\alpha} \wedge y_2 = z_1^{\beta} z_2^{\alpha}\}(m)$ is a SPK on m of an entity knowing the discrete logarithm of y_1 to the base z_1 and a representation of y_2 to the bases z_1 and z_2, where the z_2-part of this representation equals the discrete logarithm of y_1 to the base z_1. The second type is a SPK proving the knowledge of the e-th root of the discrete logarithm of $y \in G$ to the base $z \in G$ on m, and is denoted as

$$SPK\{\beta : y = z^{\beta^e}\}(m).$$

The third type is a SPK proving the knowledge of the e-th root of the z_2-part of a representation of $y \in G$ to the bases $z_1, z_2 \in G$ on m, and is denoted as

$$SPK\{(\gamma, \delta) : y = z_1^{\gamma} z_2^{\delta^e}\}(m).$$

The efficient constructions of these types of signatures are concretely described in [8].

The fourth type is a SPK proving the knowledge of the discrete logarithm of $y \in G$ to the base $z \in G$ and the double discrete logarithm of $\tilde{y}_1 \in G_1$ to the bases $\tilde{z}_1 \in G_1$ and $\tilde{z} \in G$ on m, where the discrete logarithm of y to the base z equals the double discrete logarithm of $\tilde{y}_1$ to the bases $\tilde{z}_1$ and $\tilde{z}$. This is denoted as

$$SPK\{\epsilon : y = z^{\epsilon} \wedge \tilde{y}_1 = \tilde{z}_1^{(\tilde{z}^{\epsilon})}\}(m).$$

This is described in [11]. Note that there is a difference between this type of SPK used in this paper and that in [11]. The difference is the orders of G_1 and G. The orders of G_1 and G in this paper are different, and are prime or not prime, though the orders in [11] are prime. This difference does not affect the proof that the underlying PK is zero-knowledge proof of knowledge. Since the construction in [11] utilizes a cut-and-choose method, this is less efficient than the constructions of the other types which do not utilize the method.

4 An Unlinkable Divisible Electronic Cash System

In this section, an unlinkable divisible e-cash system is constructed by using the extension of the group signature scheme [8], as well as the unlinkable e-coupon system [4]. The group signature schemes allow a group member to anonymously sign on a group's behalf. Furthermore, the anonymity of the signature can be revoked by the trusted party. In the scheme of [8], the group consists of owners of unforgeable certificates issued from the group manager. In the e-coupon system, the certificate is used as a ticket issued from the bank and the group signature is used as a transcript of a payment. This simple replacement brings the system the anonymity, unlinkability, unforgeability, no swindling, off-line-ness, and owner tracing of the anonymity revocation. Furthermore, in the e-coupon system, mechanisms to detect the payments derived from the same sub-ticket and to enable coin tracing are added. The former mechanism is that a payer is forced to send values which are the same if and only if the payer uses the same sub-ticket. The latter mechanism is that, in a withdrawal, a customer is forced to send the encryption of a value, which is linked to payments derived from the withdrawal, with the trustee's key. In the proposed divisible e-cash system, this mechanism of coin tracing is adopted, and the mechanism to detect over-spending is modified as the payer is forced to send values which are linked if and only if the nodes with the parent-child relationship are used for payments or the same node is used twice or more. In the concrete, the payer sends F value $F_{j_1 \cdots j_u}$ for the payment of the node $n_{j_1 \cdots j_u}$, as noted in Section 3.1.

Assume that each participant publishes the public key of any digital signature scheme and keeps the corresponding secret key. Hereafter, except in the payment protocol, the values sent from each participant are signed on the digital signature scheme. Furthermore, assume that all customers and shops open their accounts on the bank. Let $\tilde{0}$ be the empty string. If A is a set, $a \in_R A$ means that a is chosen at random from A according to the uniform distribution. Let $\langle g \rangle$ be a cyclic group with generator g.

4.1 Setup

To set up the cash system, the bank and trustee generate public and secret keys. The public keys described in this setup protocol are assigned to a single monetary amount $w = 2^{\ell-1}$. If multiple monetary amount is adopted, the setup is executed for each monetary amount.

1. The bank computes an RSA modulus n, two public exponents $e_1, e_2 > 1$, and two integers $f_1, f_2 > 1$. Note that e_1, e_2, f_1 and f_2 must satisfy that solving the congruence $f_1 x^{e_1} + f_2 \equiv v^{e_2} \pmod{n}$ is infeasible. The choices for e_1, e_2, f_1 and f_2 are discussed in [8]. Then, the bank computes a cyclic group $G_n = \langle g_n \rangle$ of order n which is a subgroup of $Z^*_{p_2}$ for a prime $p_2 = 2n + 1$. Similarly, the bank computes a cyclic group $G_{p_i} = \langle g_{p_i} \rangle$ of order p_i which is a subgroup of $Z^*_{p_{i+1}}$ for a prime $p_{i+1} = 2p_i + 1$ with all i $(2 \leq i \leq \ell)$. In these cases, the bank redoes the above procedure from the computation of n if $2n + 1, 2p_2 + 1$, ..., or $2p_\ell + 1$ is not prime. Furthermore, the bank chooses elements $h, \tilde{h} \in G_n$, $h_{(2,0)}, h_{(2,1)} \in G_{p_2}, \ldots,$ $h_{(\ell,0)}, h_{(\ell,1)} \in G_{p_\ell}$ whose discrete logarithms to the bases $g_n, g_{p_2}, \ldots, g_{p_\ell}$ are unknown, respectively. Note that $G_{p_2}, \ldots, G_{p_\ell}$ are constructed so that functions $h_{(2,0)}^{x_n}, h_{(3,0)}^{x_{p_2}}, \ldots, h_{(\ell,0)}^{x_{p_{\ell-1}}}, h_{(2,1)}^{x_n}, h_{(3,1)}^{x_{p_2}}, \ldots, h_{(\ell,1)}^{x_{p_{\ell-1}}}$ on inputs $x_n \in G_n$, $x_{p_2} \in G_{p_2}, \ldots, x_{p_{\ell-1}} \in G_{p_{\ell-1}}$ can be defined well as one-way bijections. Finally, the bank publishes $\mathcal{Y} = (n, e_1, e_2, f_1, f_2, G_n, G_{p_2}, \ldots, G_{p_\ell}, g_n, g_{p_2}, \ldots,$ $g_{p_\ell}, h, \tilde{h}, h_{(2,0)}, \ldots, h_{(\ell,0)}, h_{(2,1)}, \ldots, h_{(\ell,1)})$ as the public key, and keeps the factorization of n secret.
2. For all i $(0 \leq i \leq \ell - 1)$ and all $J \in \{0, 1\}^i$, the bank makes database $\mathcal{F}_{0J}$ empty, which holds F values included in the transcripts of payments using the node n_{0J} to detect over-spending in the below deposit protocol.

3. The trustee chooses $\rho \in_R Z_n^*$ to compute $y_R = h^\rho$. Then, the trustee makes y_R public, and keeps ρ secret.

4.2 Withdrawal

To withdraw a coin, a customer conducts the following protocol with the bank. This is the same as that of the e-coupon protocol, which corresponds to the issue of a membership certificate in the group signature scheme.

1. A customer chooses $x \in_R Z_n^*$ to compute $y = x^{e_1} \bmod n$ and $z = g_n^y$. Then, the customer chooses $r_1, r_2 \in_R Z_n^*$ to compute $\tilde{y} = r_1^{e_2}(f_1 y + f_2) \bmod n, C_1 = h^{r_2}\tilde{h}^y$, $C_2 = y_R^{r_2}$. Furthermore, the customer computes the following SPK's:
$$\begin{aligned} V_1 &= SPK\{\alpha : z = g_n^{\alpha^{e_1}}\}(\tilde{0}), \\ V_2 &= SPK\{\beta : g_n^{\tilde{y}} = (z^{f_1} g_n^{f_2})^{\beta^{e_2}}\}(\tilde{0}), \\ V_3 &= SPK\{(\gamma, \delta) : C_1 = h^\gamma \tilde{h}^\delta \wedge C_2 = y_R^\gamma \wedge z = g_n^\delta\}(\tilde{0}). \end{aligned}$$
The customer sends the bank $(\tilde{y}, z, C_1, C_2, V_1, V_2, V_3)$.
2. If V_1, V_2 and V_3 are correct, the bank sends the customer $\tilde{v} = \tilde{y}^{1/e_2} \bmod n$ and charges the customer's account the amount w.
3. The customer computes $v = \tilde{v}/r_1 \bmod n$ to obtain the coin (x, v), where $v \equiv (f_1 x^{e_1} + f_2)^{1/e_2} \pmod n$.

4.3 Payment

Assume that each shop owns a unique identifier. Let m be the concatenation of the identifier of the shop obtaining the payment and the time when the payment is made. In the payment protocol, the customer pays the shop any amount $\tilde{w}$ $(\leq w = 2^{\ell-1})$. Let $[\tilde{w}_\ell \cdots \tilde{w}_1]$ be the binary representation of $\tilde{w}$. Then, if $\tilde{w}_{\ell-u+1} = 1$ $(1 \leq u \leq \ell)$, the customer pays a node $n_{j_1 \cdots j_u}$ among the nodes in the u-th level that do not violate the divisible rule, as well as [2]. Here, the payment protocol for a node $n_{j_1 \cdots j_u}$ is shown. By executing this payment protocol for multiple nodes parallel, the payment for any amount is accomplished.

During the payment, the payer sends the bank F value of the paid node together with the group signature. F value of the root node, denoted F_{j_1}, is $\tilde{h}^y$. F value of a node $n_{j_1 \cdots j_u}$, denoted $F_{j_1 \cdots j_u}$, is $h_{(u,j_u)}^{F_{j_1 \cdots j_{u-1}}}$ where $F_{j_1 \cdots j_{u-1}}$ is F value of the parent node. F values of a binary tree of 3 levels are illustrated in Figure 2.

The detailed payment protocol is as follows:

1. The customer computes $\tilde{C}_1 = h^{\tilde{r}} g_n^y$ and $\tilde{C}_2 = y_R^{\tilde{r}}$ for $\tilde{r} \in_R Z_n^*$. Furthermore, the customer computes the following SPK's:
$$\begin{aligned} \tilde{V}_1 &= SPK\{(\alpha, \beta) : \tilde{C}_1 = h^\alpha g_n^{\beta^{e_1}}\}(m), \\ \tilde{V}_2 &= SPK\{(\gamma, \delta) : \tilde{C}_1^{f_1} g_n^{f_2} = h^\gamma g_n^{\delta^{e_2}}\}(m), \\ \tilde{V}_3 &= SPK\{(\epsilon, \zeta) : \tilde{C}_1 = h^\epsilon g_n^\zeta \wedge \tilde{C}_2 = y_R^\epsilon\}(m). \end{aligned}$$

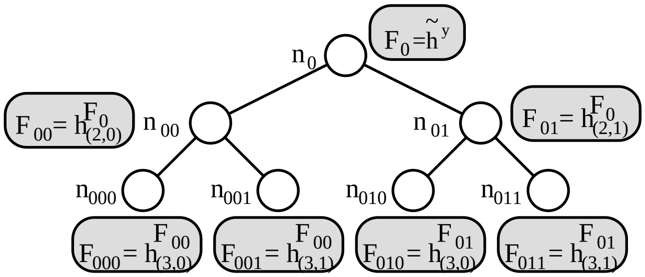

Fig. 2. F values of a binary tree of 3 levels

Note that these are the same as the group signature in [8]. Then, the customer conducts the followings according to the level u of the paid node:

Case of $u = 1$: The customer computes F value of the paid root node, $F_{j_1} = \tilde{h}^y$, and the following SPK's which proves the validity of F value:

$$\tilde{V}_4 = SPK\{(\eta, \theta) : F_{j_1} = \tilde{h}^\eta \wedge \tilde{C}_1 = h^\theta g_n^\eta\}(m).$$

Then, the customer sends the shop $A = (j_1, F_{j_1}, \tilde{C}_1, \tilde{C}_2, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4)$ as the payment.

Case of $u = 2$: The customer computes F values of the root node and the paid child node, $F_{j_1} = \tilde{h}^y, F_{j_1j_2} = h_{(2,j_2)}^{F_{j_1}}$. Then, the customer computes $\tilde{g}_n = g_n^{\tilde{r}_1}, \tilde{F}_1 = \tilde{g}_n^y$ for $\tilde{r}_1 \in_R Z_n^*$, and the following SPK's which proves the validity of $F_{j_1j_2}$:

$$\tilde{V}_4 = SPK\{(\eta, \theta) : \tilde{F}_1 = \tilde{g}_n^\eta \wedge \tilde{C}_1 = h^\theta g_n^\eta\}(m),$$
$$\tilde{V}_5 = SPK\{\iota : \tilde{F}_1 = \tilde{g}_n^\iota \wedge F_{j_1j_2} = h_{(2,j_2)}^{\tilde{h}^\iota}\}(m).$$

Then, the customer sends the shop $A = (j_1j_2, F_{j_1j_2}, \tilde{g}_n, \tilde{F}_{j_1}, \tilde{C}_1, \tilde{C}_2, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_5)$ as the payment.

Cases of $3 \leq u \leq \ell$: The customer computes F values from the root node to the paid node $n_{j_1\cdots j_u}$, $F_{j_1} = \tilde{h}^y, F_{j_1j_2} = h_{(2,j_2)}^{F_{j_1}}, \ldots, F_{j_1\cdots j_u} = h_{(u,j_u)}^{F_{j_1\cdots j_{u-1}}}$. Then, to commit F values of nodes except the paid node, the customer computes $\tilde{g}_n = g_n^{\tilde{r}_1}, \tilde{g}_{p_2} = g_{p_2}^{\tilde{r}_2}, \ldots, \tilde{g}_{p_{u-1}} = g_{p_{u-1}}^{\tilde{r}_{u-1}}, \tilde{F}_1 = \tilde{g}_n^y, \tilde{F}_2 = \tilde{g}_{p_2}^{F_{j_1}}, \ldots, \tilde{F}_{u-1} = \tilde{g}_{p_{u-1}}^{F_{j_1\cdots j_{u-2}}}$ for $\tilde{r}_1 \in_R Z_n^*, \tilde{r}_2 \in_R Z_{p_2}^*, \ldots, \tilde{r}_{u-1} \in_R Z_{p_{u-1}}^*$. Furthermore, to prove the validity of F value of the paid node by using the committed F values, the customer computes the following SPK's:

$$\tilde{V}_4 = SPK\{(\eta, \theta) : \tilde{F}_1 = \tilde{g}_n^\eta \wedge \tilde{C}_1 = h^\theta g_n^\eta\}(m),$$
$$\tilde{V}_{5,1} = SPK\{\iota_1 : \tilde{F}_1 = \tilde{g}_n^{\iota_1} \wedge \tilde{F}_2 = \tilde{g}_{p_2}^{\tilde{h}^{\iota_1}}\}(m),$$

$$\begin{aligned}
\tilde{V}_{5,2} &= SPK\{\iota_2 : \tilde{F}_2 = \tilde{g}_{p_2}^{\iota_2} \wedge \tilde{F}_3 = \tilde{g}_{p_3}^{h_{(2,j_2)}^{\iota_2}}\}(m),\\
&\cdots\\
\tilde{V}_{5,u-2} &= SPK\{\iota_{u-2} : \tilde{F}_{u-2} = \tilde{g}_{p_{u-2}}^{\iota_{u-2}} \wedge \tilde{F}_{u-1} = \tilde{g}_{p_{u-1}}^{h_{(u-2,j_{u-2})}^{\iota_{u-2}}}\}(m),\\
\tilde{V}_{5,u-1} &= SPK\{\iota_{u-1} : \tilde{F}_{u-1} = \tilde{g}_{p_{u-1}}^{\iota_{u-1}} \wedge F_{j_1 \cdots j_u} = h_{(u,j_u)}^{h_{(u-1,j_{u-1})}^{\iota_{u-1}}}\}(m).
\end{aligned}$$

Finally, the customer sends the shop $A = (j_1 j_2 \cdots j_u, F_{j_1 \cdots j_u}, \tilde{g}_n, \tilde{g}_{p_2}, \ldots, \tilde{g}_{p_{u-1}}, \tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_{u-1}, \tilde{C}_1, \tilde{C}_2, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_{5,1}, \ldots, \tilde{V}_{5,u-1})$ as the payment.

2. The shop verifies that A is correctly formed. If the shop is successful, this payment is permitted.

4.4 Deposit

In the deposit protocol, over-spending is checked on the divisibility rule shown in Section 3.1. If over-spending does not occur, the paid amount is deposited in the account of the shop. Otherwise, the over-spender can be identified by the below owner tracing protocol. When the node $n_{j_1 \cdots j_u}$ is used for the payment, the transcript of the payment includes F value $F_{j_1 \cdots j_u}$. If the same node is used, the sameness of F value indicates over-spending. If the nodes $n_{j_1 \cdots j_u}$ and $n_{j_1 \cdots j_{u'}}$ $(u < u')$ with the parent-child relationship are used which also means over-spending, the corresponding $F_{j_1 \cdots j_u}$ and $F_{j_1 \cdots j_{u'}}$ have relations as $F_{j_1 \cdots j_{u+1}} = h_{(u+1, j_{u+1})}^{F_{j_1 \cdots j_u}}, \ldots, F_{j_1 \ldots j_{u'}} = h_{(u', j_{u'})}^{F_{j_1 \cdots j_{u'-1}}}$ for F values of the intermediate nodes, $F_{j_1 \cdots j_{u+1}}, \ldots, F_{j_1 \cdots j_{u'-1}}$. Thus, the relation enables the bank to detect over-spending. The following is the detailed protocol to deposit the payment of the node $n_{j_1 \cdots j_u}$. For the payment of multiple nodes, this protocol is executed multiple times.

1. The shop sends the bank the transcript of the payment A.
2. The bank verifies that the transcript is correctly formed. Then, the bank checks whether the payment causes the node $n_{j_1 \ldots j_u}$ to be over-spent as follows:
   (a) If $u \geq 2$, for all databases $\mathcal{F}_{j_1 \cdots j_i}$ $(1 \leq i \leq u-1)$ and all $\hat{F}_i \in \mathcal{F}_{j_1 \cdots j_i}$, the bank computes $\hat{F}_{i+1} = h_{(i+1, j_{i+1})}^{\hat{F}_i}$, …, $\hat{F}_u = h_{(u, j_u)}^{\hat{F}_{u-1}}$, and checks $\hat{F}_u = F_{j_1 \cdots j_u}$.
   (b) For the database $\mathcal{F}_{j_1 \cdots j_u}$ and all $\hat{F}_u \in \mathcal{F}_{j_1 \cdots j_u}$, the bank checks $\hat{F}_u = F_{j_1 \cdots j_u}$.
   (c) If $u \leq \ell - 1$, for all i $(u+1 \leq i \leq \ell)$, all $\hat{j}_{u+1}, \ldots, \hat{j}_i \in \{0, 1\}$, all databases $\mathcal{F}_{j_1 \cdots j_u \hat{j}_{u+1} \cdots \hat{j}_i}$ and all $\hat{F}_i \in \mathcal{F}_{j_1 \cdots j_u \hat{j}_{u+1} \cdots \hat{j}_i}$, the bank computes $F_{u+1} = h_{(u+1, \hat{j}_{u+1})}^{F_u}$, …, $F_i = h_{(i, \hat{j}_i)}^{F_{i-1}}$ where $F_u = F_{j_1 \cdots j_u}$, and checks $\hat{F}_i = F_i$.

When any check is successful, the paid node is over-spent. Then, the over-spender can be identified by the owner tracing protocol. Otherwise, the amount of the node is deposited in the shop's account, and $F_{j_1 \cdots j_u}$ is kept in the bank's databases $\mathcal{F}_{j_1 \cdots j_u}$, while the transcript A is also kept since it can be used as the witness if over-spending occurs in the future.

4.5 Anonymity Revocation

When a judge's order of the anonymity revocation is given, the following owner or coin tracing protocols for a targeted payment or withdrawal is executed, respectively. Furthermore, when over-spending is detected in the deposit protocol, owner tracing for the over-spent payment is executed. The owner tracing protocol is the same as the identification of the signer in the original group signature scheme. The coin tracing protocol is arranged from that of the e-coupon system.

Owner tracing:

1. The bank sends the trustee a transcript of the targeted payment A, which includes $\tilde{C}_1$ and $\tilde{C}_2$.
2. The trustee verifies that the transcript is correctly formed. If it is correctly formed, the trustee sends the bank $\hat{z} = \tilde{C}_1 / \tilde{C}_2{}^{1/\rho}$ and $SPK\{\alpha : \tilde{C}_1 = \hat{z}\tilde{C}_2{}^{\alpha} \wedge h = y_R^{\alpha}\}(\tilde{0})$. This SPK proves that $(\tilde{C}_1, \tilde{C}_2)$ is decrypted into $\hat{z}$.
3. The bank searches z identical with $\hat{z}$ to present the customer's signature on z, which indicates the payer of A.

Coin tracing:

1. The bank sends the trustee the transcript of the targeted withdrawal $(\tilde{y}, z, C_1, C_2, V_1, V_2, V_3)$.
2. The trustee verifies that the transcript is correctly formed. If it is correctly formed, the trustee sends the bank $\hat{h} = C_1 / C_2^{1/\rho}$ and $SPK\{\alpha : C_1 = \hat{h} C_2^{\alpha} \wedge h = y_R^{\alpha}\}(\tilde{0})$. This SPK proves that (C_1, C_2) is decrypted into $\hat{h}$, which should equals $\tilde{h}^y$.
3. For the sent $\hat{h}$, the bank (and shops) checks the following for F value, $F_{j_1 \cdots j_u}$, sent during payment of the node $n_{j_1 \cdots j_u}$ $(1 \leq u \leq \ell)$:

   Case of $u = 1$: They checks $F_{j_1} = \hat{h}$.

   Case of $u = 2$: They checks $F_{j_1 j_2} = h_{(2,j_2)}^{\hat{h}}$.

   Cases of $3 \leq u \leq \ell$: They computes $F_2 = h_{(2,j_2)}^{F_1}, \ldots F_{u-1} = h_{(u-1,j_{u-1})}^{F_{u-2}}$, where $F_1 = \hat{h}$, to check $F_{j_1 \cdots j_u} = h_{(u,j_u)}^{F_{u-1}}$.

   If any check is successful, the transcript is derived from the targeted withdrawal.

5 Discussion

The e-cash system proposed in this paper as well as the original e-coupon system and group signature scheme is based on the infeasibility to compute and compare the discrete logarithms, the security of the ElGamal encryption [12] and blind RSA signature [13], and the infeasibility to compute (x, v) satisfying $f_1 x^{e_1} + f_2 \equiv v^{e_2} \pmod{n}$.

It is discussed that the proposed system satisfies the requirements in Section 2.

Unforgeability: From the infeasibility to compute (x, v) satisfying $f_1 x^{e_1} + f_2 \equiv v^{e_2} \pmod{n}$, it is infeasible to forge a coin. From the soundness of the SPK's, it is infeasible to compute the transcript of the payment without a coin.

No over-spending: The SPK's during payment assure that F value of the paid node is correct. Assume that a customer over-spends a coin. If the customer spends the same node twice or more, over-spending is detected in Step 2 (b) of the deposit protocol owing the sameness of F value. If the customer spends nodes $n_{j_1 \cdots j_u}$ and $n_{j_1 \cdots j_{u'}}$ $(1 \leq u < u' \leq \ell)$ with the parent-child relationship, the corresponding $F_{j_1 \cdots j_u}$ and $F_{j_1 \cdots j_{u'}}$ have relations as $F_{j_1 \cdots j_{u+1}} = h_{(u+1, j_{u+1})}^{F_{j_1 \cdots j_u}}, \ldots, F_{j_1 \ldots j_{u'}} = h_{(u', j_{u'})}^{F_{j_1 \cdots j_{u'-1}}}$ for F values of the intermediate nodes, $F_{j_1 \cdots j_{u+1}}, \ldots, F_{j_1 \cdots j_{u'-1}}$. Thus, in Step 2 (a) or (c) of the deposit protocol, over-spending is detected. Since the SPK's also assure that $(\tilde{C}_1, \tilde{C}_2)$ is the ElGamal encryption of z, the bank cooperating with the trustee can identify the over-spender in the owner tracing protocol.

No swindling: The blind signature prevents anyone except a customer who withdrew a coin (x, v) from obtaining the coin. Because of the secrecy of the SPK and the infeasibility to compute the discrete logarithm, no other party can obtain (x, v) from a transcript of a payment. Thus, no other party can spend the coin of a valid customer. The deposit information is a transcript of a payment. Since the transcript is unforgeable and no other party can spend the coin of a valid customer, the deposit information cannot be forged.

Anonymity: In the confirmation of the anonymity and unlinkability, the payments of the case of $3 \leq u \leq \ell$ are only discussed, since the other cases can be discussed similarly. To identify the payer, it is required to decide whether y which is used to compute the withdrawal $(\tilde{y}, z, C_1, C_2, V_1, V_2, V_3)$ and y which is used to compute the payment $(j_1 j_2 \cdots j_u, F_{j_1 \cdots j_u}, \tilde{g}_n, \tilde{g}_{p_2}, \ldots, \tilde{g}_{p_{u-1}}, \tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_{u-1}, \tilde{C}_1, \tilde{C}_2, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_{5,1}, \ldots, \tilde{V}_{5,u-1})$ are the same. In both transactions, since (C_1, C_2) and $(\tilde{C}_1, \tilde{C}_2)$ are the ElGamal encryptions and $\tilde{y}$ is a blinded message on the blind RSA signature, they reveal no information about y. Furthermore, $V_1, V_2, V_3, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_{5,1}, \ldots, \tilde{V}_{5,u-1}$ are SPK's, and thus they also reveal no information. Therefore, the possibly available values are z together with its public base g_n in the withdrawal, and the revealed F value $F_{j_1 \cdots j_u}$ and the committed F values $\tilde{F}_1, \ldots, \tilde{F}_{u-1}$ together with their public bases $\tilde{h}, h_{(2,0)}, \ldots, h_{(\ell,0)}, h_{(2,1)}, \ldots, h_{(\ell,1)}$ and the random bases $\tilde{g}_n, \tilde{g}_{p_2}, \ldots, \tilde{g}_{p_{u-1}}$ in the payment. When the revealed F value is used, the above decision is performed by deciding whether $\log_{g_n} z$ and $\log_{\tilde{h}}(\log_{h_{(2,j_2)}}(\cdots(\log_{h_{(u,j_u)}} F_{j_1 \cdots j_u})\cdots))$ are the same. However, the latter decision is infeasible owing to the following proof:

Assume on the contrary that a probabilistic polynomial time algorithm M decides whether $\log_{g_n} z$ and $\log_{\tilde{h}}(\log_{h_{(2,j_2)}}(\cdots(\log_{h_{(u,j_u)}} F_{j_1\cdots j_u})\cdots))$ are the same with a non-negligible probability. Then, the following probabilistic polynomial time algorithm $\bar{M}$ with the inputs $\bar{h}_1, \bar{h}'_1, \bar{z}_1, \bar{z}'_1 \in G_n$ can be constructed:

First, $\bar{M}$ chooses $\dot{h}_2 \in_R G_{p_2}, \ldots, \dot{h}_u \in_R G_{p_u}$. Next, from them and the input $\bar{z}'_1$, $\bar{M}$ computes $\dot{z}_2 = \dot{h}_2^{\bar{z}'_1}, \dot{z}_3 = \dot{h}_3^{\dot{z}_2}, \ldots, \dot{z}_u = \dot{h}_u^{\dot{z}_{u-1}}$. Finally, $\bar{M}$ runs M with the inputs $g_n = \bar{h}_1$, $z = \bar{z}_1$, $\tilde{h} = \bar{h}'_1$, $h_{(2,j_2)} = \dot{h}_2, \ldots$, $h_{(u,j_u)} = \dot{h}_u$, and $F_{j_1\cdots j_u} = \dot{z}_u$.

Then, since $\log_{g_n} z = \log_{\bar{h}_1} \bar{z}_1$ and $\log_{\tilde{h}}(\log_{h_{(2,j_2)}}(\cdots(\log_{h_{(u,j_u)}} F_{j_1\cdots j_u})\cdots)) = \log_{\bar{h}'_1} \bar{z}'_1$, $\bar{M}$ can decide whether $\log_{\bar{h}_1} \bar{z}_1$ and $\log_{\bar{h}'_1} \bar{z}'_1$ are the same with the non-negligible probability. This contradicts the infeasibility to decide the sameness of discrete logarithms. Thus, the decision of $\log_{g_n} z = \log_{\tilde{h}}(\log_{h_{(2,j_2)}}(\cdots(\log_{h_{(u,j_u)}} F_{j_1\cdots j_u})\cdots))$ is also infeasible. This proof also holds on the cases of the committed F values $\tilde{F}_1, \ldots, \tilde{F}_{u-1}$. Therefore, the anonymity is assured.

Unlinkability: To link two payments $(j_1 j_2 \cdots j_u, F_{j_1\cdots j_u}, \tilde{g}_n, \tilde{g}_{p_2}, \ldots, \tilde{g}_{p_{u-1}}, \tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_{u-1}, \tilde{C}_1, \tilde{C}_2, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_{5,1}, \ldots, \tilde{V}_{5,u-1})$ and $(j'_1 j'_2 \cdots j'_{u'}, F'_{j'_1\cdots j'_{u'}}, \tilde{g}'_n, \tilde{g}'_{p_2}, \ldots, \tilde{g}'_{p_{u'-1}}, \tilde{F}'_1, \tilde{F}'_2, \ldots, \tilde{F}'_{u'-1}, \tilde{C}'_1, \tilde{C}'_2, \tilde{V}'_1, \tilde{V}'_2, \tilde{V}'_3, \tilde{V}'_4, \tilde{V}'_{5,1}, \ldots, \tilde{V}'_{5,u'-1})$ for $u \leq u'$, it is required to decide whether y which used to compute them are the same. Since the use of the same y implies that F values of the same nodes are the same, the link is also performed by deciding whether F values of the same nodes are the same. The possibly available values are the reveled F values and the committed F values together with the bases used to compute them, as mentioned in the anonymity.

If the paid nodes $n_{j_1\cdots j_u}$ and $n_{j'_1\cdots j'_{u'}}$ have the parent-child relationship which it means over-spending, the payments are linkable as shown in the confirmation of no over-spending. Otherwise, $j_1 = j'_1$, …, $j_v = j'_v$ and $j_{v+1} \neq j'_{v+1}$ for some $v < u$. Then, the common youngest ancestor node of the paid nodes is $n_{j_1\cdots j_v}$. When the reveled F values $F_{j_1\cdots j_u}$ and $F'_{j'_1\cdots j'_{u'}}$ in the payments are used to link them, the link is reduced to decide whether

$$\begin{aligned}&\log_{h_{(v+1,j_{v+1})}}(\cdots(\log_{h_{(u,j_u)}} F_{j_1\cdots j_u})\cdots))\\ &= \log_{h_{(v+1,j'_{v+1})}}(\cdots(\log_{h_{(u',j'_{u'})}} F'_{j'_1\cdots j'_{u'}})\cdots))\end{aligned}$$

holds, which means to decide whether $F_{j_1\cdots j_v}$ and $F'_{j'_1\cdots j'_v}$ are the same. This decision is infeasible by the similar proof shown in the anonymity. This proof also holds on the cases of the decision between the committed F values, and the decision between the F value and the committed F value. Therefore, the unlinkability is assured.

Anonymity revocation:

Owner tracing: Since the SPK's assure that $(\tilde{C}_1, \tilde{C}_2)$ is the ElGamal encryption of z, the bank cooperating with the trustee can identify the payer from the targeted payment in the owner tracing protocol, where

$(\tilde{C}_1, \tilde{C}_2)$ in the other payments are not decrypted and thus the payments remain anonymous.

Coin tracing: The SPK's during the withdrawal assure that (C_1, C_2) is the ElGamal encryption of h^y, and the SPK's during the payment assure $F_{j_1} = \tilde{h}^y, \ldots, F_{j_1 \cdots j_u} = h_{(u,j_u)}^{F_{j_1 \cdots j_{u-1}}}$. Thus, the bank and shops cooperating with the trustee can trace the transcripts of the payments with $F_{j_1} = \tilde{h}^y$ as shown in the coin tracing protocol. Since (C_1, C_2) in the other withdrawals are not decrypted and thus the other payments remain anonymous.

From the description of the protocols, it is shown straightforwardly that the divisibility and off-line-ness hold.

Next, the efficiency of the proposed system for N, which is the divisibility precision, is discussed. The setup and withdrawal protocols are conducted in $O(\log N)$ and $O(1)$, respectively. To pay any amount of a coin, $O(\log N)$ nodes can be used. The protocols after the payment are conducted in $O(\log N)$ per a node. Thus these protocols are conducted in $O((\log N)^2)$.

6 Conclusion

In this paper, a divisible e-cash system has been proposed, where (1) the unlinkability among all payments holds, and (2) the computational complexity of all protocols is $O((\log N)^2)$. Since a type of SPK (i.e., the proof that a discrete logarithm and a double discrete logarithm are the same) utilizes a cut-and-choose method, the proposed system is less efficient than systems in [2, 3]. Therefore, an open problem is to propose the efficient unlinkable divisible e-cash system where any cut-and-choose method is not used. In addition, the security of our system is based on the heuristic assumption as the infeasibility to compute (x, v) satisfying $f_1 x^{e_1} + f_2 \equiv v^{e_2} \pmod{n}$. Thus, another open problem is to propose the system where the security is proved based on the cryptographic assumptions theoretically clarified.

References

1. Pfitzmann, B. and Waidner, M.: How to Break and Repair a "Provably Secure" Untraceable Payment System, *Advances in Cryptology — CRYPTO'91*, LNCS 576, Springer–Verlag, pp. 338–350 (1992).
2. Okamoto, T.: An Efficient Divisible Electronic Cash Scheme, *Advances in Cryptology — CRYPTO'95*, LNCS 963, Springer–Verlag, pp. 438–451 (1995).
3. Chan, A., Frankel, Y. and Tsiounis, Y.: Easy Come - Easy Go Divisible Cash, *Advances in Cryptology — EUROCRYPT'98*, LNCS 1403, Springer–Verlag, pp. 561–575 (1998).
4. Nakanishi, T., Haruna, N. and Sugiyama, Y.: Unlinkable Electronic Coupon Protocol with Anonymity Control, *Proc. of Second International Information Security Workshop, ISW'99*, LNCS 1729, Springer–Verlag, pp. 37–46 (1999).

5. Fujisaki, E. and Okamoto, T.: Practical Escrow Cash Schemes, *IEICE trans. on Fundamentals.*, Vol. E81-A, No. 1, pp. 11–19 (1998).
6. von Solms, S. and Naccache, D.: On Blind Signatures and Perfect Crimes, *Computers and Security*, Vol. 11, No. 6, pp. 581–583 (1992).
7. Frankel, Y. D. Y.: Threshold Cryptosystems, *Advances in Cryptology — CRYPTO'89*, LNCS 435, Springer–Verlag, pp. 307–315 (1990).
8. Camenisch, J. and Stadler, M.: Efficient Group Signature Schemes for Large Groups, *Advances in Cryptology — CRYPTO'97*, LNCS 1294, Springer–Verlag, pp. 410–424 (1997).
9. Fiat, A. and Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems, *Advances in Cryptology — CRYPTO '86*, LNCS 263, Springer–Verlag, pp. 186–194 (1987).
10. Bellare, M. and Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, *Proc. of First Annual Conference on Computer and Communications Security*, Association for Computing Machinery, pp. 62–73 (1993).
11. Stadler, M.: Publicly Verifiable Secret Sharing, *Advances in Cryptology — EUROCRYPT'96*, LNCS 1070, Springer–Verlag, pp. 190–199 (1996).
12. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms, *Advances in Cryptology — CRYPTO'84*, LNCS 196, Springer–Verlag, pp. 10–18 (1985).
13. Chaum, D.: Blind Signatures for Untraceable Payments, *Advances in Cryptology: Proceedings of CRYPTO '82*, Plenum Press, pp. 199–203 (1983).