Complete Fair Tracing E-cash System with Provable Security

Bin Lian[1,2], Gongliang Chen[1] and Jianhua Li[1]

[1]*School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, China*
[2]*Ningbo Institute of Technology, Zhejiang University, Ningbo, China*

lianbin_a@163.com, chengl@sjtu.edu.cn, Lijh888@sjtu.edu.cn

Abstract

E-cash enables consumers to spend e-coin anonymously in information system, but unconditional anonymity of E-cash may be misused by malicious user, therefore fair E-cash was put forward. In fair E-cash system, the anonymity of customer can be revoked when e-coins are involved in crimes, and e-coin can be traced also. However, it is challenging work that providing a practical and complete tracing E-cash scheme. After pointing out some unpractical or incomplete designs in existing schemes, we present a practical scheme with complete tracing, including trusted authority's unconditional tracing, bank's repeat-spending tracing and loss-coin tracing which is neglected in prior schemes. Moreover, for resolving efficiency problem of complete fair tracing, we design a nested structure of signature of knowledge in payment protocol, which has an obvious efficiency advantage comparing with conventional signature of knowledge. Then we provide proofs of all security attributes of our E-cash system, and compare our scheme with some current schemes in efficiency. Our original design of nested structure of knowledge proof will also make other applications of knowledge-controllable-leak more efficient.

Keywords: *E-cash, complete fair tracing, signature of knowledge, knowledge-controllable-leak, provable security*

1. Introduction

Preserving the anonymity of customers is a primary feature of E-cash [1]. But considering this could be misused for illegal purposes, such as money laundering or perfect blackmailing, S. von Solms and D. Naccache [2] pointed out unconditional anonymity should be replaced by conditional anonymity, then fair E-cash was put forward independently by [3] and [4] for the first time. That means in fair E-cash schemes, the spender remains anonymous if he uses E-cash legitimately, but if something serious, *e.g.*, crime is involved in, e-coin and its owner can be traced unconditionally.

Unfortunately, we found many schemes have not resolved fair tracing problem practically. To illustrate this, we just take several schemes [5-12, 42] for example. [5-12, 42] using different cryptography tools have something in common, which is the trusted third party (TTP) is indispensable to tracing e-coin owner/e-coin. So even if a tiny e-coin is spent twice, TTP must be in demand under any circumstance. But this is not the case in reality. In real life, the authorities usually act as TTP, so the duty and power of TTP and bank are different. Only if crimes are involved in E-cash system,

TTP perform unconditional coin tracing and/or unconditional coin-owner tracing. Otherwise, if something trivial such as account opening, slight double-spending, *etc*. happens, bank should deal with it without TTP.

On the other side, being different from [5-12, 42], [13-19, 40] have achieved double-spending tracing without TTP, but they all do not realize unconditional tracing, for it seems that there are no simple and efficient methods to do so in those schemes [13-19, 40], and this is also the reason why TTP performs double-spending tracing instead of bank [5-12, 42]. We will give deep analysis on the reason later. From the point of view of system security, unconditional tracing is necessary, not dispensable. It is not only for the reason stated above, but also because double-spending tracing can only be done after double-spending, but unconditional tracing can be done anytime. And double-spending tracing usually traces the dishonest owner, but unconditional tracing can trace e-coin and the owner.

In addition, we think at least three points else of tracing should be achieved, not all of which are realized well in many schemes. First, when do some tracing, corresponding proof should be provided, e.g. double-spending tracing corresponding to the proof of double-spending. Second, after any tracing, customer should not be slandered to be responsible for fabricated proof of payment or whatever else. And third, when customer loses his e-coins which haven't been spent by anyone, he should be able to get back his money from bank.

In brief, our definition of complete fair tracing is tracing in various situations performed by respective appropriate entities, and it fulfills the other requirements mentioned above.

There are two further explications of our design.

First, let us analyze what gets in the way of providing a practical design of fair tracing in those schemes mentioned above, *i.e*., why it is difficult that bank performs double-spending tracing and TTP performs unconditional tracing in those schemes. In general, E-cash scheme can be designed based on one of two fundamental cryptographic tools, which is blind signature [1] or signature of knowledge [20]. On one hand, for unconditional tracing, there are always some invariants in demand in system protocols. So the difficulty in schemes based on blind signature [13, 14, 18, 19] is how to generate an efficient signature which is blind to signer, but convinces him that relevant particular invariants can be extracted from the phase of generating signature and the phase of showing it unconditionally. On the other hand, for double-spending tracing, information of spender need to leak partly in payment protocols. So the difficulty in schemes based on signature of knowledge [7, 8, 9, 11] is how to modify statistical zero-knowledge protocols into knowledge-controllable-leak protocols with provable security.

Second, we would like to distinguish between unconditional coin tracing and loss-coin tracing. The former should be performed by TTP when necessary. However, the latter ignored by almost all schemes should be performed by bank as in real life. A somewhat similar concept of recoverable e-coin is first proposed in [21], and [22] give a implement scheme. Recoverable e-coin means that if customer loses his e-coin or the memorizer of e-coin breaks down, customer can execute recovery protocol with bank for regenerating the coin. But some unsolved problems [21, 22] make it infeasible. The first problem is how customer can convince bank his ownership of the lost e-coins about which he has no information since he lost it completely. The second problem is how to distinguish recovery e-coin and the identical lost one when they appear. The two problems are partly solved by [23], but it is still not a perfect scheme. First, TTP (Recovery Centre) must participate in withdrawal protocol inefficiently for recovering

e-coin [23], as we describe about fair tracing, those should be the duty of bank, not TTP. Second, after recovering one coin, the anonymity of previous spending and subsequent spending the remaining coins is not preserved [23].

Our goal is to design a complete fair tracing scheme of E-cash without the problems mentioned above. The E-cash system can practically perform tracing in various situations conforming to the reality. For achieving fair tracing efficiently, we design a nested structure of signature of knowledge in payment protocol, and prove the security of its application. And all security features are also proved under standard assumptions.

The remainder of this paper is organized as follows. Section 2 gives preliminaries of our scheme. In Section 3, we provide a model to describe the operation mode of our system. Section 4 presents our scheme. Security proofs of the scheme are given in Section 5. In Section 6, we analyze system efficiency. Finally, the paper concludes in Section 7.

2. Preliminaries

2.1. Cryptographic Assumptions

The Strong-RSA Assumption (S-RSA) was independently introduced by [24] and [25]. It strengthens the widely accepted RSA Assumption that finding e^{th}–roots modulo n — where e is the public, and thus fixed — is hard to the assumption that finding an e^{th}–roots modulo n for any $e > 1$ is hard.

Problem 1 (Strong RSA Problem). Let n=pq be an RSA-like modulus and let G be a cyclic subgroup of $\mathbb{Z}^*_n$ of order #G, $\lceil \log_2(\#G) \rceil = l_G$. Given n and $z \in G$, find $u \in G$ and $e \in \mathbb{Z}_{>1}$ satisfying $z \equiv u^e \pmod n$.

Assumption 1 (S-RSA Assumption). There exists a probabilistic polynomial-time algorithm K which on input a security parameter l_G outputs a pair (n, z) such that, for all probabilistic polynomial-time algorithms P, the probability that P can solve the S-RSA Problem is negligible.

The Diffie-Hellman Assumption presented by [26] is in two styles: the Computational Diffie-Hellman Assumption (CDH) and the Decisional Diffie-Hellman Assumption (DDH). A thorough discussion on the subject can be seen in [27].

Problem 2 (Decisional Diffie-Hellman Problem). Let $G=\langle g \rangle$ be a cyclic group generated by g of order u=#G with $\lceil \log_2(u) \rceil = l_G$. Given g, g^x, g^y and $g^z \in G$, decide whether g^{xy} and g^z are equal.

Assumption 2 (DDH Assumption). There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between (g, g^x, g^y, g^z) and (g, g^x, g^y, g^{xy}) with $x,y,z \in_R \mathbb{Z}_u$.

2.2. Signatures of Knowledge [28]

All these signatures of knowledge given below can be proved secure in the random oracle model [29], and their interactive versions are statistical (honest-verifier) zero-knowledge proofs of knowledge. In the following, we consider three building blocks: signature of knowledge of (1) a discrete logarithm; (2) equality of two discrete logarithms; and (3) a discrete logarithm lying in a given interval. All of these are constructed over a cyclic group G

= 〈g〉 the order of which #G is unknown; However its bit-length l_G (*i.e.*, the integer l_G s.t. $2^{l_G-1} \leq \#G < 2^{l_G}$) is publicly known. Fujisaki and Okamota show that [25], under the S-RSA, the standard proofs of knowledge protocols that work for a group of known order are also proofs of knowledge in this setting. We assume a collision-resistant hash function H: $\{0,1\}^* \rightarrow \{0,1\}^k$ which maps a binary string of arbitrary length to a k-bit hash value. We also assume a security parameter $\varepsilon > 1$.

Definition 1. Let y, g∈G. A pair (c,s) $\in \{0,1\}^k \times \pm\{0,1\}^{\varepsilon(l_G+k)+1}$ verifying $c=H(y \| g \| g^s y^c \| m)$ is a signature of knowledge of the discrete logarithm of $y=g^x$ w.r.t. base g, on a message $m \in \{0,1\}^*$. It is denoted as SPK(α: $y=g^\alpha$) (m).

The party in possession of the secret $x=\log_g y$ is able to compute the signature by choosing a random $t \in \pm\{0,1\}^{\varepsilon(l_G+k)}$ and then computing c and s as: $c=H(y \| g \| g^t \| m)$ and s=t-cx (in $\mathbb{Z}$).

Definition 2. Let y_1, y_2, g, h ∈G. A pair (c,s) $\in \{0,1\}^k \times \pm\{0,1\}^{\varepsilon(l_G+k)+1}$ verifying $c=H(y_1 \| y_2 \| g \| h \| g^s y_1^c \| h^s y_2^c \| m)$ is a signature of knowledge of the discrete logarithm of both $y_1=g^x$ w.r.t. base g and $y_2=h^x$ w.r.t. base h, on a message $m \in \{0,1\}^*$. It is denoted as SPK(α: $y_1=g^\alpha \wedge y_2=h^\alpha$) (m).

The party in possession of the secret x is able to compute the signature, provided that $x=\log_g y_1=\log_h y_2$, by choosing a random $t \in \pm\{0,1\}^{\varepsilon(l_G+k)}$ and then computing c and s as: $c=H(y_1 \| y_2 \| g \| h \| g^t \| h^t \| m)$ and s=t-cx (in $\mathbb{Z}$).

Definition 3. Let y, g∈G. A pair (c,s) $\in \{0,1\}^k \times \pm\{0,1\}^{\varepsilon(l+k)+1}$ verifying $c=H(y \| g \| g^{s-cX} y^c \| m)$ is a signature of knowledge of the discrete logarithm $\log_g y$ that lies in $]X-2^{\varepsilon(l+k)}, X+2^{\varepsilon(l+k)}[$, on a message $m \in \{0,1\}^*$. It is denoted as SPK(α: $y = g^\alpha \wedge \alpha \in]X-2^{\varepsilon(l+k)}, X+2^{\varepsilon(l+k)}[$) (m).

From the knowledge of $x= \log_g y \in]X-2^l, X+2^l[$, this signature is obtained by choosing a random $t \in \{0,1\}^{\varepsilon(l+k)}$ and computing c and s as: $c=H(y \| g \| g^t \| m)$, $s=t-c(x-X)$ (in $\mathbb{Z}$).
Remark. Although the party knows a secret x in $]X-2^l, X+2^l[$, the signature only guarantees that x lies in $]X-2^{\varepsilon(l+k)}, X+2^{\varepsilon(l+k)}[$. See more security proofs and extended modes in [30]. The general method on designing zero knowledge proofs can be seen in [31].

2.3. The Representation Problem[13]

Definition 4. Let $k \geq 2$ be a constant, q be a prime. A generator-tuple of length k is a k-tuple $(g_1,\ldots, g_k)$ with, for all i, $j \in \{1,\ldots, k\}$, $g_i \in G_q \setminus \{1\}$ and $g_i \neq g_j$ if $i \neq j$. An index-tuple of length k is a k-tuple $(a_1,\ldots, a_k)$ with $a_i \in \mathbb{Z}_q$ for all $i \in \{1,\ldots, k\}$. For any $h \in G_q$, a representing index-tuple (also called a representation) of h with respect to a generator-tuple $(g_1,\ldots, g_k)$ is an index-tuple $(a_1,\ldots, a_k)$ such that $\prod_{1 \leq i \leq k} g_i^{a_i} = h$.

The representation problem is to find a representation of h with respect to $(g_1,\ldots, g_k)$.

Proposition 1. Define V to be the set of all functions of the form $q(\cdot) / r(\cdot)$, such that $q(\cdot)$ and $r(\cdot)$ are polynomials with integer domain and integer coefficients, and $q(k) > r(k) \geq 1$ for all sufficiently large k. For any functions $f_1(\cdot)$, $f_2(\cdot)$, $f_3(\cdot)$, $f_4(\cdot) \in V$, the following four statements are equivalent:

(1) There exists a polynomial-time algorithm $A_{(1)}$ which, on inputs a generator-tuple of length k and $h \in G_q$, outputs a representing index-tuple of h with probability of success at least $1/ f_1(|p|)$ for all sufficiently large p.

(2) There exists a polynomial-time algorithm $A_{(2)}$ which, on inputs a generator-tuple of length k, outputs a nontrivial representing index-tuple of 1 with probability of success at least $1/ f_2(|p|)$ for all sufficiently large p.

(3) There exists a $h \in G_q \setminus \{1\}$ and a polynomial-time algorithm $A_{(3)}$ which, on inputs a generator-tuple of length k, outputs a representing index-tuple of h with probability of success at least $1/ f_3(|p|)$ for all sufficiently large p.

(4) There exists a polynomial-time algorithm $A_{(4)}$ which solves the Discrete Log problem with probability of success at least $1/ f_4(|p|)$ for all sufficiently large p.

Proposition 1 states that the representation problem for groups of prime order is equivalent in computational difficulty to the Discrete Log problem.

Corollary 1. Under the Discrete Log assumption, there cannot exist a polynomial-time algorithm which, on input a generator-tuple $(g_1,\ldots, g_k)$ chosen at random, outputs a number $h \in G_q$ and two different representing index-tuples of h with nonnegligible probability.

2.4. Number-Theoretic Foundations of Our Scheme

Considering the security of system, it is necessary to choose a safe RSA modulus n, *i.e.*, $n=pq$, with $p \neq q$, $p=2p'+1$, $q=2q'+1$, and p, q, p', q' are all odd prime. So we can restrict operation to the subgroup of quadratic residues modulo *n*, *i.e.*, the cyclic subgroup $QR(n)$ generated by an element of order $p'q'$ which has no small factors.

Note that the Corollary 2 not only shows how to achieve the security feature described above, but also provides a method to verify the order of an element without knowing it. And it is useful in the system setup of our scheme.

Theorem 1. The condition for existence of primitive root modulo n is n = 2, 4, p^{α}, $2p^{\alpha}$, and α is any integer, p is odd prime.

Proposition 2. Let n=pq, where $p \neq q$, $p=2p'+1$, $q=2q'+1$, and p,q,p',q' are all odd prime. Integer a s.t. gcd(a,n)=1. $ord_n(a)$, i.e. the order of the a is one of the set $\{1,2,p',q',2p',2q',p'q',2p'q'\}$. Moreover, if $ord_n(a)$ is equal to $p'q'$ or $2p'q' \Leftrightarrow \gcd(a\pm1,n)=1$.

Corollary 2. Let $p',q' \in_R I_{lp}$ (where $I_d = \pm\{0,1\}^d$, $\in_R$ denote choosing at random), n be as in Proposition 2. For integer x s.t. gcd(x,n)=1 and $\gcd(x\pm1,n)=1$, $a \equiv x^2 \pmod n$, $\langle a \rangle \subset \mathbb{Z}^*_n$ is a cyclic subgroup of order $p'q'$. Moreover, for $k \in_R I_{lp-1}$ s.t. $y \equiv a^k \pmod n$, $\langle y \rangle \subset \mathbb{Z}^*_n$ is a cyclic subgroup of order $p'q'$.

3. Our e-cash Model

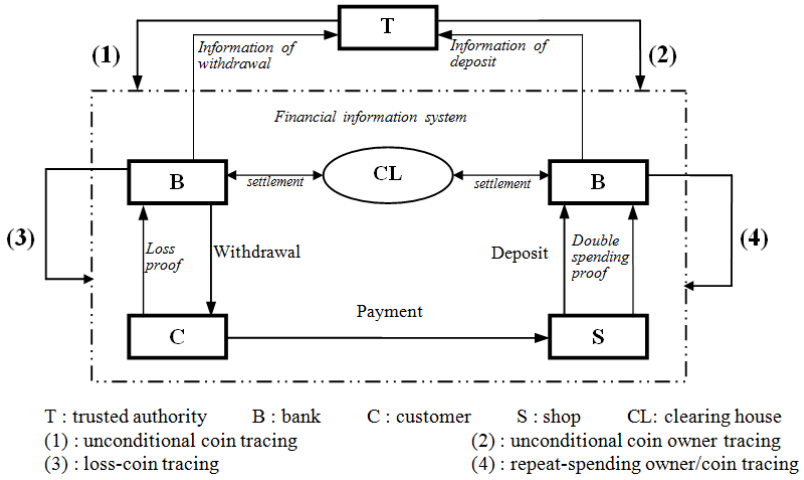

Figure 1. Complete Fair Tracing E-cash Model

4. Complete Fair Tracing e-cash

4.1. System Setup

Let $\varepsilon > 1$, k and l_p be security parameters. Let λ_1, λ_2, γ_1 and γ_2 denote lengths satisfying $\lambda_2 > 4l_p$, $\lambda_1 > \varepsilon(\lambda_2 + k) + 2$, $2^{\gamma_1}-2^{\gamma_2} > 2^{\lambda_1+1}+2^{\lambda_2+1}+2^{2\lambda_1+2}+2^{2\lambda_2+2}+2^{\lambda_1+\lambda_2+3}$ and $\gamma_1 > \varepsilon(\gamma_2 + k) + 2$. Define $\Lambda =]2^{\lambda_1}-2^{\lambda_2}, 2^{\lambda_1}+2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1}-2^{\gamma_2}, 2^{\gamma_1}+2^{\gamma_2}[$. Let H be a collision-resistant hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$. ($\varepsilon$ controls the tightness of the statistical zero-knowledge and l_p sets the modulus size [32].)

B (bank)'s Setup:

Select random secret l_p-bits primes p', q' such that $p = 2p'+1$ and $q = 2q'+1$ are primes. Provide a proof that $n = pq$ is the product of two safe primes [33].

T (trusted authority)'s Setup:

Choose random elements $a,a_0,a_1,g,g_1,g_2,h_1,h_2$ of $QR(n)$ (of order $p'q'$, all characteristics including randomness of elements can be verified, see Corollary 2).

$K_1,K_2 \in_R I_{lp-1}$ and set $h=a_1^{K_1} \pmod n$ and $a_2=h^{K_2} \pmod n$ (the order of h and a_2 is also $p'q'$, see Corollary 2).

C (customer)'s Setup:

$SK_C \in_R I_{2lp}$ and set $PK_C = g^{SK_C} \pmod n$.

The system public key of complete fair tracing E-cash is $PK = (n,a,a_0,a_1,a_2,g,g_1,g_2,h,h_1,h_2PK_C)$, the private key of bank is $SK_B=p'q'$, the private key of trusted authority is $SK_T=(K_1,K_2)$, and the private key of customer is SK_C.

4.2. Withdrawal Protocol

C **B**

$\tilde{x},\tilde{x}_1,\tilde{x}_2 \in_R]0, 2^{\lambda_2}[$, $\tilde{r},\tilde{r}_1,\tilde{r}_2 \in_R]0, n^2[$
$C_1 = g^{\tilde{x}} h^{\tilde{r}} \pmod n$
$C'_1 = g_1^{\tilde{x}_1} h_1^{\tilde{r}_1} \pmod n$ $C''_1 = g_2^{\tilde{x}_2} h_2^{\tilde{r}_2} \pmod n$
$U = SPK(\alpha,\beta,\alpha_1,\beta_1,\alpha_2,\beta_2 : C_\Pi = C_1 \cdot C'_1 \cdot C''_1 = g^{\alpha} h^{\beta} g_1^{\alpha_1} h_1^{\beta_1} g_2^{\alpha_2} h_2^{\beta_2})$

$C_1, C'_1, C''_1, U, r_C, PK_C$ →

Verifies $C_1, C'_1, C''_1 \in QR(n)$
Verifies U $\tilde{\alpha},\tilde{\alpha}_1,\tilde{\alpha}_2 \in_R \mathbb{Z}^*_{2^{\lambda_2}}$
$\tilde{\beta},\tilde{\beta}_1,\tilde{\beta}_2 \in_R]0,2^{\lambda_2}[$

← $\tilde{\alpha},\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\beta},\tilde{\beta}_1,\tilde{\beta}_2$

$x = 2^{\lambda_1} + (\tilde{\alpha}\tilde{x} + \tilde{\beta} \pmod{2^{\lambda_2}})$
$e_1 = 2^{\lambda_1} + (\tilde{\alpha}_1\tilde{x}_1 + \tilde{\beta}_1 \pmod{2^{\lambda_2}})$
$e_2 = 2^{\lambda_1} + (\tilde{\alpha}_2\tilde{x}_2 + \tilde{\beta}_2 \pmod{2^{\lambda_2}})$
$C_2 = a^x \pmod n$
$C'_2 = a_1^{e_1} \pmod n$ $C''_2 = a_2^{e_2} \pmod n$
$r \in_R \{0, 1\}^{2l_p}$
$A_1 = h^{e_1} PK_C{}^r \pmod n$ $A_2 = g^r \pmod n$
$W = SPK(\alpha,\beta,\gamma,\alpha_1,\beta_1,\gamma_1,\alpha_2,\beta_2,\gamma_2,\delta,\varepsilon:$

$\alpha,\alpha_1,\alpha_2 \in]-2^{\lambda_2}, 2^{\lambda_2}[$
$\wedge\ C_2/a^{2^{\lambda_1}} = a^{\alpha} \wedge C_1^{\tilde{\alpha}} g^{\beta} = g^{\alpha}(g^{2^{\lambda_2}})^{\beta} h^{\gamma}$
$\wedge\ C'_2/a_1^{2^{\lambda_1}} = a_1^{\alpha_1} \wedge C'_1{}^{\tilde{\alpha}_1} g_1^{\beta_1} = g_1^{\alpha_1}(g_1^{2^{\lambda_2}})^{\beta_1} h_1^{\gamma_1}$
$\wedge\ C''_2/a_2^{2^{\lambda_1}} = a_2^{\alpha_2} \wedge C''_1{}^{\tilde{\alpha}_2} g_2^{\beta_2} = g_2^{\alpha_2}(g_2^{2^{\lambda_2}})^{\beta_2} h^{\gamma_2}$
$\wedge\ A_1 = h^{\delta} PK_C{}^{\varepsilon} \wedge C'_2 = a_1^{\delta} \wedge A_2 = g^{\varepsilon})$

$J = SPK(\alpha : PK_C = g^{\alpha})(H(r_C \| A_1 \| A_2 \| timestamp))$

$C_2, C'_2, C''_2, A_1, A_2, W, J$ →

Verifies $C_2, C'_2, C''_2 \in QR(n)$
Verifies W, J. $e \in_R \Gamma$ prime
$A \equiv (a_0 C_2 C'_2 C''_2)^{1/e} \pmod n$
Debits **C**'s account

← A, e

Verifies $A^e \equiv a_0 C_2 C'_2 C''_2 \pmod n$
$\equiv a_0 a^x a_1^{e_1} a_2^{e_2} \pmod n$

In the withdrawal protocol, C and B jointly generate the coin secret (e_1, e_2, x). Then, **C** obtains a coin (e_1,e_2,x,[A,e], r_C) s.t. $A^e \equiv a_0 a^x a_1^{e_1} a_2^{e_2} \pmod n$ in which e_1, e_2 and x are only known to C. r_C is series number for loss register. The pairs (A_1,A_2) is an ElGamal encryption of h^{e_1} prepared for loss-coin tracing, and W is to ensure (e_1,e_2,x) are generated correctly and confirm "unconditional/loss coin tracing". (C'_2, C''_2) is used to ensure "repeat-spending owner tracing" and "unconditional coin owner tracing".

4.3. Payment Protocol

C **S**

$w \in_R I_{2lp}$
$T_1 = Ag^w \pmod n$ $T_2 = h^{e_1} \pmod n$
using $A^e = a_0 a^x a_1^{e_1} a_2^{e_2} \pmod n$, then
$a_0 = T_1^e / (a^x a_1^{e_1} a_2^{e_2} g^{ew}) \pmod n$
$r_1 \in_R I_{\varepsilon(\gamma_2+k)}$ $r_2, r_3 \in_R I_{\varepsilon(\lambda_2+k)}$
$r_4 \in_R I_{\varepsilon(\gamma_1+2lp+k+1)}$
$d_1 = T_1^{r_1} / (a^{r_2} a_1^{e_2} a_2^{r_3} g^{r_4}) \pmod n$
$d_2 = h^{e_2} \pmod n$
$d_2' = h^{r_3} \pmod n$

T_1, T_2, d_1, d_2, d_2' →
$c_1 = H(T_1 \| T_2 \| d_1 \| d_2 \| d_2' \| tr)$
← c_1

$s_1 = r_1 - c_1(e - 2^{\gamma_1})$
$s_2 = r_2 - c_1(x - 2^{\lambda_1})$
$s_\sigma = -e_2 + c_1 e_1$
$s_3 = r_3 - c_1(e_2 - 2^{\lambda_1})$
$s_4 = r_4 - c_1 ew$
(all in $\mathbb{Z}$) } (*)

$s_1, s_2, s_\sigma, s_3, s_4$ →

Verifies $c_1 = H(T_1 \| T_2 \|$
$a_0^{c_1} T_1^{s_1 - c_1 2^{\gamma_1}} / (a^{s_2 - c_1 2^{\lambda_1}} a_1^{-s_\sigma} a_2^{s_3 - c_1 2^{\lambda_1}} g^{s_4}) \pmod n$
$\| T_2^{c_1} h^{-s_\sigma} \pmod n \| d_2^{c_1} h^{s_3 - c_1 2^{\lambda_1}} \pmod n \| tr)$
$s_1 \in I_{\varepsilon(\gamma_2+k)+1}$, $s_2, s_3 \in I_{\varepsilon(\lambda_2+k)+1}$,
$s_\sigma \in I_{(\lambda 1+k+1)}$, $s_4 \in I_{\varepsilon(\gamma_1+2lp+k+1)+1}$

In this protocol, **C** should confirm that $c_1 \neq 0$. The *tr* used to generate c_1 includes information such as the identification number of **S (shop)**, timestamp and a random value, *etc*.

Let us analyze one point of our original design. Considering the primary feature of E-cash, *i.e.*, anonymity of customer, now there are two major ways to implement it. The first one is using blindness of signature to cut off the connection between signing and verifying signature (for schemes based on blind signature, see our classification of E-cash schemes in Section 1), and the second one is using zero-knowledge proof to cut off the connection between signing and verifying signature (for schemes based on signature of knowledge). However, for schemes based on signature of knowledge, the attribute of statistical zero-knowledge is redundant for security of E-cash scheme, because an e-coin is spent only once, it is different from other applications of signature of knowledge such as group signature which needs statistical security. So our payment equation set (*) is not statistical zero-knowledge, and we also design a nested structure of signature of knowledge to improve computation efficiency.

Therefore the payment protocol may be considered that *tr* is group signed with one-off anonymity. How to leak the spender's information is easy to see from repeat-spending owner tracing below, and whether the information-leak is controlled in a secure manner can be seen in part 3 of Section 5.

4.4. Loss Register Protocol

When customer loses all information of E-coin with loss register series number r_{Cx}, he sends ($r_{C1},\ldots,r_{Cx-1}, r_{Cx+1},\ldots,r_{Cn}$) of his remaining E-coin to bank, then completes the loss register protocol. As r_{Ci} is not used in payment protocol, it has no influence on anonymity of customer. Note that only after confirming T_{2x} hadn't been used in system, i.e. after a system period, customer can get his refund. Our loss register protocol can work on network executed by computer or mobile phone.

C **B**

$LR_x = \{(r_{C1},\ldots,r_{Cx-1}, r_{Cx+1}, \ldots,r_{Cn}) \| PK_C\}$

LR_x →

Search for the records corresponding to r_{Cx}

← $r_{Cx}, A_{1x}, A_{2x}, timestamp_x, J_x$

Verifies J_x, $timestamp_x$
Confirms r_{Cx} not in $\{r_{C1},\ldots,r_{Cx-1},r_{Cx+1}, \ldots,r_{Cn}\}$
$T_{2x}=h^{e_1} = A_{1x} / A_{2x}^{SK_C} \pmod n$
$J'_x=SPK(\alpha:PK_C=g^{\alpha} \wedge A_{1x}/T_{2x}=A_{2x}^{\alpha})(timestamp'_x)$

T_{2x} , J'_x →

Verifies J'_x, and search T_{2x} in database, if not found, publishes T_{2x} for **Loss-coin tracing**

4.5. Deposit and Practical Tracing Protocol

S sends $Proof_{payment}$ = ($T_1,T_2,c_1,s_1,s_2,s_\sigma,s_3,s_4,tr$) to B (note that $d_2 = T_2^{c_1}h^{-s_\sigma} \pmod n$). After verifying $c_1 = H(T_1 \| T_2 \| a_0^{c_1}T_1^{s_1-c_1 2^{\gamma_1}}/(a^{s_2-c_1 2^{\lambda_1}}a_1^{-s_\sigma}a_2^{s_3-c_1 2^{\lambda_1}}g^{s_4}) \pmod n \| T_2^{c_1}h^{-s_\sigma} \pmod n \| d_2^{c_1}h^{s_3-c_1 2^{\lambda_1}} \pmod n \| tr)$, B searches T_2 in database, if not found, stores $Proof_{payment}$. This is the normal case. If T_2 has already been in database, B checks c_1. If it is the same one, that is S deposits repeatedly. Otherwise, C had spent the e-cash more than once.

Repeat-spending owner tracing :

$$\begin{cases} s_\sigma = -e_2 + c_1 e_1 \\ s'_\sigma = -e_2 + c'_1 e_1 \end{cases}$$

B can figure out e_1 and e_2 from the equation set. ($\mathrm{ord}_n(a) = \mathrm{ord}_n(a_0) = \mathrm{ord}_n(a_1) = \mathrm{ord}_n(a_2) = \mathrm{ord}_n(g) = \mathrm{ord}_n(h) = p'q'$)

$$e_1 = \frac{s_\sigma - s'_\sigma}{c_1 - c'_1} \pmod{p'q'}$$

$$e_2 = \frac{c'_1 s_\sigma - c_1 s'_\sigma}{c_1 - c'_1} \pmod{p'q'}$$

So "repeat-spending owner tracing" is achieved as B checks $C'_2=a_1^{e_1} \pmod n$ or $C''_2=a_2^{e_2} \pmod n$ that received from C in withdrawal. For avoiding the double-spending coin is spent again, B do the following.

Repeat-spending coin tracing :

After receiving double-spending coin, B publish $T_2=h^{e_1} \pmod n$ and $d_2=h^{e_2} \pmod n$, so it can be identified averting its repeat-spending.

When special situations such as crimes emerge,

Unconditional coin owner tracing : T is given d_2 which appears in payment, then $C''_2 = a_2^{e_2} = (h^{K_2})^{e_2} = (h^{e_2})^{K_2} = {d_2}^{K_2} \pmod n$. So cooperating with B , the e-cash owner will be found. Moreover, *SPK* (α: $C''_2 = {d_2}^{\alpha} \wedge a_2 = h^{\alpha}$) also can be given as a proof.

Unconditional coin tracing : T is given C'_2 which appears in withdrawal, then $T_2 = h^{e_1} = (a_1^{K_1})^{e_1} = (a_1^{e_1})^{K_1} = {C'_2}^{K_1} \pmod n$. So everyone can identify the coin when it is spent. Likewise, *SPK* (α: $T_2 = {C'_2}^{\alpha} \wedge h = a_1^{\alpha}$) can be given as a proof.

5. Security of the Proposed Scheme

5.1. Unforgeability of Coin

Theorem 2. Under the S-RSA assumption, probabilistic polynomial-time (PPT) adversaries (even consist of all participants except **B**) cannot, with non-negligible probability (in l_p), output a coin (e_1,e_2,x,[A,e]) s.t. $A^e = a_0 a^x a_1^{e_1} a_2^{e_2} \pmod n$ with $e_1, e_2, x \in \Lambda$ and $e \in \Gamma$ that is different from all the coins obtained in the withdrawal protocol (where the withdrawal protocol can be performed in arbitrary manners).

Proof. Let *M* be an attacker that is allowed to adaptively run withdrawal protocol and thereby obtains coins, (e_{1j}, e_{2j}, x_j, [A_j, e_j]) j =1,…,*K*. It will show that if *M* outputs ($\hat{e}_1$, $\hat{e}_2$, $\hat{x}$ [$\hat{A}$, $\hat{e}$]), *s.t.* $\hat{A}^{\hat{e}} = a_0 a^{\hat{x}} a_1^{\hat{e}_1} a_2^{\hat{e}_2} \pmod n$ *with* $\hat{e}_1, \hat{e}_2, \hat{x} \in \Lambda$ *and* $\hat{e} \in \Gamma$, and ($\hat{e}_1, \hat{e}_2, \hat{x}, \hat{A}, \hat{e}$) $\neq (e_{1j}, e_{2j}, x_j, A_j, e_j)$ for all $1 \le j \le K$ with non-negligible probability, then the S-RSA assumption does not hold.

There are two possibilities: one is that gcd($\hat{e}$, e_j) = 1 for all j ($1 \le j \le K$); other is that gcd($\hat{e}$, e_j) = e_j for at least one j ($1 \le j \le K$). Correspondingly, there are two algorithms as follows. Given (n, v), we repeatedly play a random one of the following two algorithms with *M* and hope to calculate (u, d) $\in \mathbb{Z}^*_n \times \mathbb{Z}_{>1}$ satisfying $u^d \equiv v \pmod n$ from *M*'s answers.

The algorithm 1:

1. Select $e_{11},\ldots,e_{1K},e_{21},\ldots,e_{2K},x_1,\ldots,x_K \in \Lambda$ and $e_1,\ldots,e_K \in \Gamma$.
2. Set $a = v^{\prod_{1 \le l \le K} e_l} \pmod n$.
3. Choose $r, \theta \in_R \Lambda$, $K_1, K_2 \in_R I_{lp-1}$, $g \in_R QR(n)$ and set $a_0 = a^r \pmod n$, $a_1 = a^{\theta} \pmod n$, $h = a_1^{K_1} = a^{\theta K_1} \pmod n$, $a_2 = h^{K_2} = a^{\theta K_1 K_2} = a^{\sigma} \pmod n$.
4. For all $1 \le i \le K$, compute $A_i = v^{(x_i + r + \theta e_{1i} + \sigma e_{2i}) \prod_{1 \le l \le K;\, l \ne i} e_l} \pmod n$. (so, $A_i^{e_i} = a_0 a^{x_i} a_1^{e_{1i}} a_2^{e_{2i}} \pmod n$, $1 \le i \le K$)
5. Run the Withdrawal Protocol *K* times with *M* on input (n,a,a_0,a_1,a_2,g,g_1,g_2,h,h_1,$h_2 PK_C$). At the *i*-th run, receive the commitment C_{1i},C'_{1i},C''_{1i} and U_i from *M*. Use the proof of knowledge U_i to extract $\tilde{x}_i$, $\tilde{x}_{1i}$, $\tilde{x}_{2i}$, $\tilde{r}_i$, $\tilde{r}_{1i}$, $\tilde{r}_{2i}$ such that $C_{1i} = g^{\tilde{x}_i} h^{\tilde{r}_i} \pmod n$, $C'_{1i} = g_1^{\tilde{x}_{1i}} h_1^{\tilde{r}_{1i}} \pmod n$, $C''_{1i} = g_2^{\tilde{x}_{2i}} h_2^{\tilde{r}_{2i}} \pmod n$ (this involves rewinding of M). Choose $\tilde{\alpha}_i$, $\tilde{\beta}_i$, $\tilde{\alpha}_{1i}$, $\tilde{\beta}_{1i}$, $\tilde{\alpha}_{2i}$, $\tilde{\beta}_{2i}$ such that the prepared $x_i = 2^{\lambda_1} + (\tilde{\alpha}_i \tilde{x}_i + \tilde{\beta}_i \pmod{2^{\lambda_2}})$, $e_{1i} = 2^{\lambda_1} + (\tilde{\alpha}_{1i} \tilde{x}_{1i} + \tilde{\beta}_{1i} \pmod{2^{\lambda_2}})$, $e_{2i} = 2^{\lambda_1} + (\tilde{\alpha}_{2i} \tilde{x}_{2i} + \tilde{\beta}_{2i} \pmod{2^{\lambda_2}})$ and send $\tilde{\alpha}_i$, $\tilde{\beta}_i$, $\tilde{\alpha}_{1i}$, $\tilde{\beta}_{1i}$, $\tilde{\alpha}_{2i}$, $\tilde{\beta}_{2i}$ to *M*. Follow the protocol and then send *M* the prepared [A_i, e_i].

   After these *K* withdrawals, *M* outputs ($\hat{e}_1$,$\hat{e}_2$,$\hat{x}$,[$\hat{A}$,$\hat{e}$]), *s.t.* $\hat{A}^{\hat{e}} = a_0 a^{\hat{x}} a_1^{\hat{e}_1} a_2^{\hat{e}_2} \pmod n$ with $\hat{e}_1, \hat{e}_2, \hat{x} \in \Lambda$ *and* $\hat{e} \in \Gamma$, and ($\hat{e}_1, \hat{e}_2, \hat{x}, \hat{A}, \hat{e}$) $\neq (e_{1j}, e_{2j}, x_j, A_j, e_j)$ for all $1 \le j \le K$.

6. If there exists j ($1 \le j \le K$) such that $\gcd(\hat{e}, e_j)\neq 1$ then output $\perp$ and quit. Otherwise, let $\tilde{e} := (\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_2)\prod_{1\le l\le K}e_l$. Now we have $v^{\tilde{e}} = v^{\prod_{1\le l\le K}e_l(\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_1)} = a_0a^{\hat{x}}a_1^{\hat{e}_1}a_2^{\hat{e}_2}=\hat{A}^{\hat{e}} \pmod n$. Since $\gcd(\hat{e},e_j)=1$ for all $1\le j\le K$, then $\gcd(\hat{e},\tilde{e})=\gcd(\hat{e}, (\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_2))$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta, \in \mathbb{Z}$ s.t. $\alpha\hat{e} + \beta\tilde{e} = \gcd(\hat{e}, (\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_2))$. Let $u := v^{\alpha}\hat{A}^{\beta} \pmod n$ and $d := \hat{e}/\gcd(\hat{e}, (\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_2)) > 1$ ($\because \hat{x},r,\theta,\hat{e}_1,\sigma,\hat{e}_2 \in \Lambda$, $\hat{e} \in \Gamma$, $2^{\gamma_1}-2^{\gamma_2} > 2^{\lambda_1+1}+2^{\lambda_2+1}+2^{2\lambda_1+2}+2^{2\lambda_2+2}+2^{\lambda_1+\lambda_2+3} > 2^{\lambda_1+1}+2^{\lambda_2+1}+2^{2\lambda_1+1}+2^{2\lambda_2+1}+2^{\lambda_1+\lambda_2+2}$ $\therefore \hat{e}>(\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_2)$), and then $\because v^{\tilde{e}} = \hat{A}^{\hat{e}} \pmod n$ $\therefore u^d=v^{\alpha d}\hat{A}^{\beta d}=v^{(\alpha\hat{e}+\beta\tilde{e})/\gcd(\hat{e},(\hat{x}+r+\theta\hat{e}_1+\sigma\hat{e}_1))}=v \pmod n$. Output (u,d).

The algorithm 2:

1. Select $e_{11},\ldots,e_{1K},e_{21},\ldots,e_{2K},x_1,\ldots,x_K\in\Lambda$ and $e_1,\ldots,e_K\in\Gamma$.

2. Choose $j\in_R\{1,\ldots, K\}$ and set $a = v^{\prod_{1\le l\le K;\ l\neq j} e_l} \pmod n$.

3. Choose $r,\theta\in_R\Lambda$, $K_1,K_2\in_R I_{l_p-1}$, $g\in_R QR(n)$ and set $A_j=a^r \pmod n$, $a_1=a^{\theta} \pmod n$, $h=a_1^{K_1}=a^{\theta K_1} \pmod n$, $a_2=h^{K_2}= a^{\theta K_1K_2}= a^{\sigma} \pmod n$, and $a_0=A_j^{e_j}/a^{x_j}a_1^{e_{1j}}a_2^{e_{2j}} \pmod n$.

4. For all $1\le i \le K$, $i\neq j$, compute $A_i= v^{(x_i+e_jr-x_j-\theta e_{1j}-\sigma e_{2j}+\theta e_{1i}+\sigma e_{2i})\prod_{1\le l\le K;\ l\neq i,j} e_l} \pmod n$. (so, $A_i^{e_i}=a_0a^{x_i}a_1^{e_{1i}}a_2^{e_{2i}} \pmod n$, $1 \le i \le K$, $i\neq j$).

5. Run the Withdrawal Protocol K times with M on input $(n,a,a_0,a_1,a_2,g,g_1,g_2,h,h_1,h_2PK_C)$. At the i-th run, receive the commitment C_{1i},C'_{1i},C''_{1i} and U_i from M. Use the proof of knowledge U_i to extract $\tilde{x}_i$, $\tilde{x}_{1i}$, $\tilde{x}_{2i}$, $\tilde{r}_i$, $\tilde{r}_{1i}$, $\tilde{r}_{2i}$ such that $C_{1i}=g^{\tilde{x}_i}h^{\tilde{r}_i} \pmod n$, $C'_{1i}=g_1^{\tilde{x}_{1i}}h_1^{\tilde{r}_{1i}} \pmod n$, $C''_{1i}=g_2^{\tilde{x}_{2i}}h_2^{\tilde{r}_{2i}} \pmod n$ (this involves rewinding of M). Choose $\tilde{\alpha}_i$, $\tilde{\beta}_i$, $\tilde{\alpha}_{1i}$, $\tilde{\beta}_{1i}$, $\tilde{\alpha}_{2i}$, $\tilde{\beta}_{2i}$ such that the prepared $x_i = 2^{\lambda_1} + (\tilde{\alpha}_i\tilde{x}_i + \tilde{\beta}_i \pmod{2^{\lambda_2}})$, $e_{1i}=2^{\lambda_1} + (\tilde{\alpha}_{1i}\tilde{x}_{1i}+ \tilde{\beta}_{1i} \pmod{2^{\lambda_2}})$, $e_{2i}=2^{\lambda_1} + (\tilde{\alpha}_{2i}\tilde{x}_{2i}+ \tilde{\beta}_{2i} \pmod{2^{\lambda_2}})$ and send $\tilde{\alpha}_i$, $\tilde{\beta}_i$, $\tilde{\alpha}_{1i}$, $\tilde{\beta}_{1i}$, $\tilde{\alpha}_{2i}$, $\tilde{\beta}_{2i}$ to M. Follow the protocol and then send M the prepared $[A_i, e_i]$.

   After these K withdrawals, M outputs $(\hat{e}_1,\hat{e}_2,\hat{x},[\hat{A},\hat{e}])$, *s.t.* $\hat{A}^{\hat{e}}=a_0a^{\hat{x}}a_1^{\hat{e}_1}a_2^{\hat{e}_2} \pmod n$ with $\hat{e}_1,\hat{e}_2,\hat{x}\in\Lambda$ *and* $\hat{e}\in\Gamma$, and $(\hat{e}_1,\hat{e}_2,\hat{x},\hat{A},\hat{e}) \neq (e_{1j},e_{2j},x_j,A_j,e_j)$ for all $1\le j \le K$.

6. If $\gcd(\hat{e}, e_j) = 1$ for all j ($1 \le j \le K$) then output $\perp$ and quit. Otherwise, $\gcd(\hat{e}, e_j) = e_j$ for at least one j ($1 \le j \le K$). $\exists t\in\mathbb{Z}$ s.t $\hat{e}=te_j$, $B := \hat{A}^t/A_j \pmod n$ if $\hat{x} \ge x_j$, and $B := A_j/\hat{A}^t \pmod n$ otherwise. Hence $B \equiv (a^{|\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|})^{1/e_j} \equiv v^{|\tilde{e}|/e_j} \pmod n$ with $\tilde{e} := (\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2)\prod_{1\le l\le K;\ l\neq j} e_l$. Since $\gcd(e_j, \prod_{1\le l\le K;\ l\neq j} e_l,) = 1$ then $\gcd(e_j, |\tilde{e}|) = \gcd(e_j, |\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|)$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta, \in \mathbb{Z}$ s.t. $\alpha e_j + \beta|\tilde{e}| = \gcd(e_j, |\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|)$. Let $u := v^{\alpha}B^{\beta} \pmod n$ and $d := e_j/\gcd(e_j, |\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|) > 1$ ($\because \hat{x}, x_j, \theta, e_{1j}, \hat{e}_1, \sigma, e_{2j}, \hat{e}_2 \in \Lambda$, $e_j \in \Gamma$, $2^{\gamma_1} - 2^{\gamma_2} > 2^{\lambda_1+1}+2^{\lambda_2+1}+2^{2\lambda_1+2}+2^{2\lambda_2+2}+2^{\lambda_1+\lambda_2+3}$ $\therefore e_j > |\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|$), $\because v^{|\tilde{e}|/e_j} \equiv B \pmod n$ $\therefore u^d = v^{\alpha d}B^{\beta d} = v^{(\alpha e_j+\beta|\tilde{e}|)/\gcd(e_j,|\hat{x}-x_j-\theta e_{1j}-\sigma e_{2j}+\theta\hat{e}_1+\sigma\hat{e}_2|)} = v \pmod n$. Output (u, d) .

So, by randomly running one of the two algorithms and cooperating with M, the S-RSA problem can be solved in expected running-time polynomial in K. As solving S-RSA problem is considered infeasible, we can conclude that no one can create more than K coins from K withdrawals without B (where K is polynomial in l_p).

5.2. Authenticity of Payment

Theorem 3. Under the S-RSA assumption, the payment protocol underlying the E-cash scheme is a proof of knowledge of a coin (e_1,e_2,x,[A,e]).

Proof. We show that the knowledge extractor is able to recover the coin (e_1,e_2,x,[A,e]) once it has found two accepting tuples: (T_1, T_2, d_1, d_2, d_2', c_1, s_1, s_2, s_σ, s_3, s_4) and (T_1, T_2, d_1, d_2, d_2', $\tilde{c}_1$, $\tilde{s}_1$, $\tilde{s}_2$, $\tilde{s}_\sigma$, $\tilde{s}_3$, $\tilde{s}_4$).

Because $d_1 = a_0^{c_1} T_1^{s_1 - c_1 2^{\gamma_1}} / (a^{s_2 - c_1 2^{\lambda_1}} a_1^{-s_\sigma} a_2^{s_3 - c_1 2^{\lambda_1}} g^{s_4}) = a_0^{\tilde{c}_1} T_1^{\tilde{s}_1 - \tilde{c}_1 2^{\gamma_1}} / (a^{\tilde{s}_2 - \tilde{c}_1 2^{\lambda_1}} a_1^{-\tilde{s}_\sigma} a_2^{\tilde{s}_3 - \tilde{c}_1 2^{\lambda_1}} g^{\tilde{s}_4})$ (mod n). Under the Discrete Log assumption, considering difficulty of representation problem[13], we have the same (r_1~r_4,w) in both equation sets (∗) due to the extention of Corollary 1. It follows that $[T_1^e / (a_0 a^x a_1^{e_1} a_2^{e_2} g^{ew})]^{\tilde{c}_1 - c_1} = 1$ (mod n), then $T_1^{s_1 - \tilde{s}_1} = (T_1^{-2^{\gamma_1}} a_0 a^x a_1^{e_1} a_2^{e_2} g^{ew})^{\tilde{c}_1 - c_1}$ (mod n). Letting $\sigma_1 = \gcd(s_1 - \tilde{s}_1, \tilde{c}_1 - c_1)$, by the extended Euclidean algorithm, there exist $\alpha_1, \beta_1 \in \mathbb{Z}$ s.t. $\alpha_1(s_1 - \tilde{s}_1) + \beta_1(\tilde{c}_1 - c_1) = \sigma_1$. Hence, $T_1 \equiv T_1^{[\alpha_1(s_1 - \tilde{s}_1) + \beta_1(\tilde{c}_1 - c_1)]/\sigma_1} \equiv [(T_1^{-2^{\gamma_1}} a_0 a^x a_1^{e_1} a_2^{e_2} g^{ew})^{\alpha_1} T_1^{\beta_1}]^{(\tilde{c}_1 - c_1)/\sigma_1}$ (mod n). Note that $(\tilde{c}_1 - c_1) \geq \sigma_1$ as $\sigma_1 = \gcd(s_1 - \tilde{s}_1, \tilde{c}_1 - c_1)$ and if $(\tilde{c}_1 - c_1) > \sigma_1$, $[(T_1^{-2^{\gamma_1}} a_0 a^x a_1^{e_1} a_2^{e_2} g^{ew})^{\alpha_1} T_1^{\beta_1}]$ is $[(\tilde{c}_1 - c_1)/\sigma_1]^{\text{th}}$ root which contradicts the S-RSA assumption, so we have $(\tilde{c}_1 - c_1) = \sigma_1$, that is $\exists\ \kappa_1 \in \mathbb{Z}$ s.t $s_1 - \tilde{s}_1 = \kappa_1 (\tilde{c}_1 - c_1)$. As d_1 in two accepting tuples are equal, for the same reason[13], $d_1 = T_1^{r_1} / (a^{r_2} a_1^{e_2} a_2^{r_3} g^{r_4})$ (mod n), then $r_1 = s_1 + c_1(e - 2^{\gamma_1}) = \tilde{s}_1 + \tilde{c}_1(e - 2^{\gamma_1})$, $r_4 = s_4 + c_1 ew = \tilde{s}_4 + \tilde{c}_1 ew$, so we have:

$$e = \kappa_1 + 2^{\gamma_1} = (s_1 - \tilde{s}_1)/(\tilde{c}_1 - c_1) + 2^{\gamma_1}$$

and $w = (s_4 - \tilde{s}_4)/(\tilde{c}_1 - c_1)e$. Thus we obtain:

$$A = T_1 / g^w = T_1 / g^{[(s_4 - \tilde{s}_4)/(\tilde{c}_1 - c_1)][(s_1 - \tilde{s}_1)/(\tilde{c}_1 - c_1) + 2^{\gamma_1}]}$$

Similarly, the knowledge extractor can also recovered the coin secret (e_1,e_2,x). So it concludes the proof.

5.3. Anonymity of Customer

Theorem 4. When a coin is spent only once, the coin (e_1,e_2,x,[A,e]) can not be figured out from (∗).

Proof. For achieving "repeat-spending owner tracing" without T, our payment protocol doesn't use the classical mode of the signature of knowledge. So we have to show that though the underlying interactive protocol is not statistical zero-knowledge but "knowledge-partly-leak" is under control. That is to say, we have to analyze the solvability of (∗), more exactly, whether any one of (e_1,e_2,x,[A,e]) can be figured out from (∗).

$$\left.\begin{array}{l} \left.\begin{array}{l} s_1 = r_1 - c_1(e - 2^{\gamma_1}) \\ s_2 = r_2 - c_1(x - 2^{\lambda_1}) \\ s_\sigma = -e_2 + c_1 e_1 \end{array}\right\} (**) \\ s_3 = r_3 - c_1(e_2 - 2^{\lambda_1}) \\ s_4 = r_4 - c_1 ew \quad \ldots\ldots\ldots\ldots (1) \end{array}\right\} (*)$$

For simplifying the problem, we can focus on the frontal four equations of (∗), *i.e.*, (∗∗). That is because (∗) has integer solutions according to Theorem 3, and equation *(1)* always has infinite many integer solutions as r_4 and w can be random selected. So *(1)* doesn't change solvability of (∗).

We transform (**) into (**1).

$$(**1)\begin{cases} c_1 e - r_1 = c_1 \cdot 2^{\gamma_1} - s_1 \\ c_1 x - r_2 = c_1 \cdot 2^{\lambda_1} - s_2 \\ c_1 e_1 - e_2 = s_\sigma \\ c_1 e_2 - r_3 = c_1 2^{\lambda_1} - s_3 \end{cases}$$

Then the coefficient matrix M_1 of (**1) is:

$$\begin{array}{c} e\ \ x\ e_1\ e_2\ r_1\ r_2\ r_3 \\ \begin{pmatrix} c_1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & c_1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1 & 0 & 0 & -1 \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} e\ \ x\ e_1\ e_2\ r_1\ r_2\ r_3 \\ \begin{pmatrix} c_1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & c_1 & 0 & 0 & 0 & -1/c_1 \\ 0 & 0 & 0 & c_1 & 0 & 0 & -1 \end{pmatrix} \end{array}$$

Note that rank of matrix M_1, *i.e.*, $r(M_1) = 4 < 7$ implies (**1) has infinite many integer solutions, but it does not equate to that any one of $(e_1,e_2,x,[A,e])$ can not be figured out from (**1). The row vectors of the matrix M_1 is:

$$\begin{aligned} \alpha_1 &= (\ c_1\ \ 0\ \ 0\ \ 0\ -1\ \ 0\ \ 0\ ), \\ \alpha_2 &= (\ 0\ \ c_1\ \ 0\ \ 0\ \ 0\ -1\ \ 0\ ), \\ \alpha_3 &= (\ 0\ \ 0\ \ c_1\ \ 0\ \ 0\ \ 0\ -1/c_1), \\ \alpha_4 &= (\ 0\ \ 0\ \ 0\ \ c_1\ \ 0\ \ 0\ -1\ ). \end{aligned}$$

For figuring out the coin component, take e_2 for example, need to get the coefficient matrix like below (the constant $\Delta \neq 0$), but we will prove it is infeasible.

$$\begin{pmatrix} c_1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & c_1 & 0 & 0 & 0 & -1/c_1 \\ 0 & 0 & 0 & \Delta \cdot c_1 & 0 & 0 & 0 \end{pmatrix}$$

That is we have (k_1,k_2,k_3,k_4) satisfy $\alpha'_4 = (0\ 0\ 0\ \Delta \cdot c_1\ 0\ 0\ 0) = k_1 \cdot \alpha_1 + k_2 \cdot \alpha_2 + k_3 \cdot \alpha_3 + k_4 \cdot \alpha_4$. Because $\Delta \neq 0$, $c_1 \neq 0$ (see payment protocol), that implies $(k_1,k_2,k_3,k_4) \neq (0,0,0,0)$. That is

$$(***)\begin{cases} k_1 c_1 + 0 + 0 + 0 = 0 \\ 0 + k_2 c_1 + 0 + 0 = 0 \\ 0 + 0 + k_3 c_1 + 0 = 0 \\ 0 + 0 + 0 + k_4 c_1 = \Delta c_1 \\ -k_1 + 0 + 0 + 0 = 0 \\ 0 - k_2 + 0 + 0 = 0 \\ 0 + 0 - k_3/c_1 - k_4 = 0 \end{cases}$$

Then the augmented matrix X of (***) is (X denotes the coefficient matrix of (***)):

$$\begin{array}{c} k_1\ \ k_2\ \ k_3\ \ k_4 \\ \begin{pmatrix} c_1 & 0 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 \\ 0 & 0 & c_1 & 0 & 0 \\ 0 & 0 & 0 & c_1 & \Delta c_1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1/c_1 & -1 & 0 \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} k_1\ \ k_2\ \ k_3\ \ k_4 \\ \begin{pmatrix} c_1 & 0 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 & 0 \\ 0 & 0 & c_1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & \Delta c_1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

As $c_1 \neq 0$, if $\Delta \neq 0$, then $r(X) \neq r(X)$, that is (***) has no solutions, but Theorem 3 shows it is not the case. So $\Delta=0$, then (***) is homogeneous equation set. And as $r(X)=4$, so (***) has unique zero solution. These all contradict prior suppose, so we conclude that e_2 can not be figured out from (**), nor from (*).Similarly, we also can prove (e_1,x,A,e) can not be figured out from (*).

Theorem 5. Under the Discrete Logarithm assumption and the DDH assumption, if e-coins are spent only once, given all views of withdrawal and payment, no PPT machine (but **T**) can determine the coin owner with non-negligible probability better than random guessing (in l_{p-1}).

Proof . First, the symbols connected with **C'** identity are in several circumstances:

$\tilde{x}$, $\tilde{x}_1$, $\tilde{x}_2$, A, e_2 are information-theoretically hiding in C_1, C'_1, C''_1, T_1, d_1.

x, e_1, e_2, h^{e_1} are computationally hard from C_2, C'_2, T_2, C''_2, d_2, A_1, A_2 under Discrete Logarithm assumption.

e_1, e_2, x, e are computationally infeasible from s_1, s_2, s_σ, s_3, s_4 due to Theorem 4.

Now we analyze if any adversary excluding T can distinguish $C'_2=a_1^{e_1} \pmod n$ from $T_2=h^{e_1} \pmod n$ or distinguish $C''_2=a_2^{e_2} \pmod n$ from $d_2=h^{e_2} \pmod n$. Because a_1, K_1, K_2 is randomly chosen by T, and $h=a_1^{K_1} \pmod n$ and $a_2=h^{K_2} \pmod n$. That is $h=a_1^{\psi} \pmod n$, $a_2=h^{\sigma} \pmod n$ with ψ, $\sigma \in_R Z^*_{p'q'}$ to adversaries. Then if adversaries can identify the customer (the coin corresponding the determinate withdrawal) implies with non-negligible probability they can distinguish $(a_1,a_1^{\psi},a_1^{e_1},a_1^{\psi e_1})$ from $(a_1,a_1^{\psi},a_1^{e_1},a_1^{\upsilon})$ and $(h,h^{\sigma},h^{e_2},h^{\sigma e_2})$ from $(h,h^{\sigma},h^{e_2},h^{\tau})$ with $\psi,\upsilon,\sigma,\tau \in_R Z^*_{p'q'}$, note $T_2=h^{e_1}=a_1^{\psi e_1} \pmod n$, $C''_2=a_2^{e_2}=h^{\sigma e_2} \pmod n$. It contradicts the DDH assumption.

So only given a valid payment tuple (T_1, T_2, d_1, d_2, d_2', c_1, s_1, s_2, s_σ, s_3, s_4), identifying the actual customer is computationally infeasible for everyone but T.

5.4. Strong Exculpability of Customer

Theorem 6. Only **C** can spend the e-coin which is withdrawn by himself, and *no one else can collude to counterfeit the proof of* C*'s spending.*

Proof . The proofs consist of two parts: the proof of system setup and the proof of system protocol.

Proof of system setup

For resisting some existent attack [34] on group signature [28], our parameter (a,a_0,a_1) are random chosen, that can be verified (see in system setup of section 4). As [35] point out, the essence of the attack presented in [34] is that some unanticipated knowledge also can be used in the structure of knowledge signature so as to fabricate signature of other knowledge. And the system setup of our scheme can frustrate this attack.

Proof of system protocol

First in withdrawal protocol, no one else get any information about x apart from C_2 in withdrawal protocol, *i.e.*, x is computationally hidden from everyone but C under the Discrete Logarithm assumption. Because if anyone excluding C knows x, it means he can calculate $x = \log_a C_2 \pmod n$, or he can calculate $x = 2^{\lambda_1} + (\tilde{\alpha}\tilde{x} + \tilde{\beta} \pmod{2^{\lambda_2}})) \pmod n$ where $\tilde{x}$ is information-theoretically hiding in C_1. And W guarantee C_2 is generated correctly, *i.e.*, no one can decide x beforehand.

Next in the first step of payment protocol, T_1 is an unconditionally binding commitment to x. Then calculating $(s_1,s_2,s_\sigma,s_3,s_4)$ in response to challenge c_1 will not reveal x due to Theorem 4.

Even when customer spends E-cash more than once, that is (T_1, T_2, d_1, d_2, d_2', c_1, s_1, s_2, s_σ, s_3, s_4), ($\tilde{\tilde{T}}_1$, T_2, $\tilde{d}_1$, d_2, $\tilde{d}_2'$, $\tilde{c}_1$, $\tilde{s}_1$, $\tilde{s}_2$, $\tilde{s}_\sigma$, $\tilde{s}_3$, $\tilde{s}_4$)⋯are got, though (e_1,e_2)will be figured out that is presented in section 4, but x will not be recovered. It is because $s_2=r_2-c_1(x-2^{\lambda_1})$, and r_2 will be updated every time. Moreover, even if the factorization of n is publicly known, and (e_1,e_2,A,e) are published, the interactive proof underlying payment protocol is still a proof of knowledge of x due to Theorem 3. So no one except the e-coin owner can finish payment unless $x=\log_a C_2$, or $x=\log_a(A^e/a_0a_1^{e_1}a_2^{e_2})$ or $w=\log_g(T_1/A)$, $x=\log_a(T_1{}^e/a_0a_1^{e_1}a_2^{e_2}g^{ew})$ can be computed (provided that l_p is larger than twice to output length of the hash function/size of the challenges).

In addition, it is impossible to perform the whole protocol with the absence of C, as $J=SPK(\alpha:PK_C=g^\alpha)(H(r_C \| A_1 \| A_2 \| timestamp))$ can not be fabricated unless $SK_C=\log_g PK_C$ can be computed. So customer only needs to be responsible for his own behavior, and nobody can collude to frame him up.

6. Efficiency Analysis

6.1. Storage Space of the E-cash Systems

We summarize the storage space of several recent E-cash schemes. To achieve equivalent secure level, the scheme of [10] has a point P of 160 bits and q of 160 bits; The system of [16](single spend of scheme 1) takes p=170 bits; The off-line e-cash system proposed by [36] has a point P of 160 bits and 160 bits prime q; The system of [37] has 160 bits prime q and 321 bits prime p; In [9, 15](system one),[38-40],and our scheme, the group G would be the group of quadratic residue modulus a safe-prime product n, which would be of 1024 bits. For convenience of compare, table 1 gives the total storage space of customer, bank and shop.

Table 1. Storage Space of the E-cash Schemes

	[9]	[10]	[15]	[16]	[36]	[37]	[38]	[39]	[40]	Our scheme
Withdrawal [bit]	4416	1120	5632	1704	1824	800	6784	8160	6420	5908
Payment [bit]	7732	480	15200	4768	1282	1304	6836	5188	30740	2996
Deposit [bit]	7732	960	15200 +2400x	4768 +512x	3232	1656	6836	5164	27648	2996

x: is equal to the quantity of deposit coins

6.2. Computation Cost of the E-cash Systems

We list the main computation of the protocols of each scheme in a unified way, *i.e.*, neglecting addition, modular addition, hash function, *etc.*, L=10 and t=40 as a moderate value in [39, 40].

Table 2. Computation Cost of the E-cash Schemes

		[9]	[10]	[15]	[16]	[36]	[37]	[38]	[39]	[40]	Our scheme
Withdrawal	$\mathbf{F_1}$	18	8	12	9	16	15	44	2156	8	32
	$\mathbf{F_2}$	0	0	0	2	0	0	0	22	5+*x*	0
Payment	$\mathbf{F_1}$	20	2	40	37	11	9	29	34	1673	8
	$\mathbf{F_2}$	0	3	0	10	0	0	0	14	0	0
Deposit	$\mathbf{F_1}$	7	0	11	10	5	7	6	10	14	3
	$\mathbf{F_2}$	0	3	0	4	0	0	0	0	0	0

F_1: multi-based exponentiation, F_2: bilinear pairing,
x: is related to system parameters

First, in our opinion, it is only for reference that various E-cash schemes compare with each other in efficiency. Because each scheme uses different technology, more importantly, each scheme has different features such as our complete fair tracing. In spite of that, we think our scheme is efficient and give a brief illustration of it. See from above tables, the most efficient scheme is [10], but we don't think it has an obvious efficiency advantage over our scheme for 4 reasons: (1). In [10], when tracing a dishonest customer, TTP is indispensable, and [10] achieve only one kind of tracing. In contrast, our scheme achieve practical complete tracing; (2). [10] Is designed based on secure communication channel which leads to costly computation and those computations are not included in table 1 and table 2. In contrast, our scheme is not based on secure channel; (3). Some burdensome processes such as the registration protocol of [10] are not counted in table 1 and table 2, and their registration protocol is necessary for every executing protocols. In contrast, our scheme needs no registration protocol; (4). The tracing method of [10] is TTP searches in customer information list which have to be stored in TTP beforehand. In contrast, the tracing method of our scheme is a simple computation without storing any information of customer in TTP.

In general, similar schemes constructed over cyclic group using exponentiation have larger storage size than schemes using bilinear group which has short representations of group elements. But for parameters yielding comparable security, the former is faster than the latter [41]. And in a good implementation [41], the computation of the multi-based exponentiation does not take far more time than a single exponentiation, so for comparing clearly, we treat single exponentiation as multi-based exponentiation uniformly. Note that if considering precomputation performed by customer, in our payment stage, the computation of customer is 5 additions, and the computation of shop is 3 multi-based exponentiations and 2 hash operations. And payment always has request of efficiency in application.

7. Conclusion

Our nested structure of signature of knowledge solves the difficulty in designing fair E-cash efficiently. And we present the first complete fair tracing E-cash scheme with provable security. Considering the reality, in our system, bank performs repeat-spending owner/coin tracing and loss-coin tracing without TTP; TTP performs unconditional owner/coin tracing when necessary. Based on the security proofs of our scheme, each part in the system can be protected to a high degree, and all security features of the system are proved under standard assumptions. According to the efficiency analysis, our method of constructing nested structure of signature of

knowledge may be useful to other cryptographic applications of knowledge-controllable-leak.

Acknowledgements

This work is supported by Major Project of Chinese National Programs for Fundamental Research and Development (973 Program, No: 2010CB731403), Security Management of Information Content Innovation Base (No: TS0010303001), National Engineering Laboratory for Information Content Analysis Technology (No: GT036001), The National Natural Science Fund (No: 61271220) and Natural Science Foundation of Ningbo (No: 2012A610064).

References

[1] D. Chaum, "Blind signatures for untraceable payments", CRYPTO'82, Plenum Press, New York, **(1983)**, pp. 199-203.
[2] S. von Solms and D. Naccache, "On blind signatures and perfect crimes", Computers & Security, vol. 11, **(1992)**, pp. 581-583.
[3] M. Stadler, J. Piveteau and J. Camenisch, "Fair blind signatures", Advances in Cryptology Eurocrypt'95, **(1995)**, pp. 209-219.
[4] E. Brickell, P. Gemmell and D. Kravitz, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change", Proceedings of the 6th annual ACM-SIAM symposium on Discrete algorithms, **(1995)**, pp. 457-466.
[5] A. Lysyanskaya and Z. Ramzan, "Group blind digital signatures: A scalable solution to electronic cash", FC'98, **(1998)**, pp. 197-238.
[6] J. Zhang, L. Ma, and Y. Wang, "Fair E-Cash System without Trustees for Multiple Banks", CISW 2007, **(2007)**, pp. 585-587.
[7] G. Maitland and C. Boyd, "Fair electronic cash based on a group signature scheme", Information and Communications Security, **(2001)**, pp. 461-465.
[8] W. Qiu and K. Chen, "A new offline privacy protecting e-cash system with revokable anonymity", Information Security, **(2002)**, pp. 177.
[9] S. Canard and J. Traoré, "On fair e-cash systems based on group signature schemes", ACISP2003, **(2003)**, pp. 237-248.
[10] H. Oros and C. Popescu, "A Secure and Efficient Off-line Electronic Payment System for Wireless Networks", Intl. J. of Computers, Comm. and Control, Suppl. Issue, vol. V, no. 4, **(2010)**, pp. 551-557.
[11] S. Canard, C. Delerablée, A. Gouget, E. Hufschmitt, F. Laguillaumie, H. Sibert, J. Traoré and D. Vergnaud, "Fair E-Cash: Be Compact, Spend Faster", Information Security, **(2009)**, pp. 294-309.
[12] Y. Chen, J. S. Chou, H. M. Sun and M. H. Cho, "A novel electronic cash system with trustee-based anonymity revocation from pairing", Electronic Commerce Research and Applications, **(2011)**.
[13] S. Brands and C. v. W. e. Informatica, "An efficient off-line electronic cash system based on the representation problem", Citeseer, **(1993)**.
[14] S. Brands, "Untraceable off-line cash in wallet with observers", Advances in Cryptology—CRYPTO'93, **(1994)**, pp. 302-318.
[15] J. Camenisch, S. Hohenberger and A. Lysyanskaya, "Compact e-cash", Advances in Cryptology–EUROCRYPT 2005, **(2005)**, pp. 302-321.
[16] M. Au, W. Susilo and Y. Mu, "Practical compact e-cash", Proceedings of the 12th Australasian conference on Information security and privacy, **(2007)**, pp. 431-445.
[17] M. Belenkiy, M. Chase, M. Kohlweiss and A. Lysyanskaya, "Compact e-cash and simulatable VRFs revisited", Pairing-Based Cryptography–Pairing 2009, **(2009)**, pp. 114-131.
[18] Z. Eslami and M. Talebi, "A new untraceable off-line electronic cash system", Electronic Commerce Research and Applications, vol. 10, **(2011)**, pp. 59-66.
[19] Z. Tan, "An Off-line Electronic Cash Scheme Based on Proxy Blind Signature", The Computer Journal, vol. 54, **(2011)**, pp. 505-512.
[20] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups", Advances in Cryptology—CRYPTO'97, **(1997)**, pp. 410-424.
[21] B. Schoenmakers, "Security aspects of the E-cash™ payment system", State of the Art in Applied Cryptography. LNCS 1528. Springer-Verlag, **(1998)**, pp. 338-352.

[22] W. S. Juang, "RO-cash: An efficient and practical recoverable pre-paid offline e-cash scheme using bilinear pairings", Journal of Systems and Software, vol. 83, **(2010)**, pp. 638-645.
[23] J. Liu, P. Tsang and D. Wong, "Recoverable and untraceable e-cash", Public Key Infrastructure, **(2005)**, pp. 206-214.
[24] N. Bari and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees", Advances in Cryptology-EUROCRYPT'97, vol. 1233 of LNCS, **(1997)**, pp. 480-494.
[25] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations", Advances in Cryptology-CRYPTO'97, **(1997)**, pp. 16-30.
[26] W. Diffie and M. Hellman, "New directions in cryptography", Information Theory, IEEE Transactions on, vol. 22, **(1976)**, pp. 644-654.
[27] D. Boneh, "The decision diffie-hellman problem", Algorithmic Number Theory, **(1998)**, pp. 48-63.
[28] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme", Advances in Cryptology-CRYPTO 2000, **(2000)**, pp. 255-270.
[29] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols", Proceedings of the 1st ACM conference on Computer and communications security, **(1993)**, pp. 62-73.
[30] J. Camenisch, R. Chaabouni and A. Shelat, "Efficient protocols for set membership and range proofs", Advances in Cryptology-ASIACRYPT 2008, **(2008)**, pp. 234-252.
[31] J. Camenisch, A. Kiayias and M. Yung, "On the portability of generalized schnorr proofs", Advances in Cryptology-EUROCRYPT 2009, **(2009)**, pp. 425-442.
[32] J. Camenisch, "Group signature schemes and payment systems based on the discrete logarithm problem", PhD thesis, vol. 2 of ETH Series in Information Security an Cryptography, Hartung-Gorre Verlag, Konstanz, ISBN 3-89649-286-1, **(1998)**.
[33] J. Camenisch and M. Michels, "Proving in zero-knowledge that a number is the product of two safe primes", Advances in Cryptology-EUROCRYPT'99, **(1999)**, pp. 107-122.
[34] Z. Cao, "Analysis of one popular group signature scheme", Advances in Cryptology–ASIACRYPT 2006, **(2006)**, pp. 460-466.
[35] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, "Remarks on Analysis of one popular group signature scheme in Asiacrypt 2006", International Journal of Applied Cryptography, vol. 1, **(2009)**, pp. 320-322.
[36] H. Wang, J. Cao and Y. Zhang, "A flexible payment scheme and its role-based access control", IEEE Transactions on Knowledge and Data Engineering, **(2005)**, pp. 425-436.
[37] M. Lee, G. Ahn, J. Kim, J. Park, B. Lee, K. Kim and H. Lee, "Design and implementation of an efficient fair off-line e-cash system based on elliptic curve discrete logarithm problem", Journal of communication and networks, vol. 4, **(2002)**, pp. 81-89.
[38] B. Lian, G. L. Chen and J. H. Li, "A Provably Secure and Practical Fair E-cash Scheme", 2010 IEEE International Conference on Information Theory and Information Security, **(2010)**.
[39] M. Au, W. Susilo and Y. Mu, "Practical anonymous divisible e-cash from bounded accumulators", Financial Cryptography and Data Security, pp. 287-301, **(2008)**.
[40] S. Canard and A. Gouget, "Divisible e-cash systems can be truly anonymous", Advances in Cryptology-EUROCRYPT 2007, **(2007)**, pp. 482-497.
[41] J. Camenisch and J. Groth, "Group signatures: Better efficiency and new theoretical aspects", Security in Communication Networks, **(2005)**, pp. 120-133.
[42] K. O. Elaalim and S. Yang, "Fair Electronic Cash System with Identity-Based Group Signature Scheme", Journal of Information Security, **(2012)**, pp. 177-183.

Authors

Bin Lian received his M.S. degree in cryptography from Southwest Jiaotong University in 2005, Chengdu. Now he works at Ningbo Institute of Technology, Zhejiang University. Currently, he is a Ph.D. candidate in the School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai. His research interests lie in cryptography and E-commerce security.

Gongliang Chen received his B.S. in Peking University, and M.S. degree in Chinese Academy of Science. In 1993, he received his Ph.D. degree in Université de Saint Etienne, France. He is also a visiting scholar of Université Paris VI, France. He is currently a professor at the School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai. His main research area includes cryptographic theory, technology of network security.

Jianhua Li received his Ph.D. degree in Shanghai Jiao Tong University, Shanghai in 1998**.** He is now a professor and doctoral supervisor at the School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai. He is also the director of National Engineering Laboratory for Information Content Analysis Technology. His recent research directions are cryptography and provable security in distributed and embedded systems.