SCIS 2007 The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

An Unlinkable Off-line E-Cash System

Yosuke Tomii * Yoshifumi Manabe Tatsuaki Okamoto †

Abstract— This paper presents an off-line anonymous e-cash scheme, that is secure under the strong RSA assumption and the strong Diffie-Hellman (SDH) assumption. A user can withdraw a wallet containing 2^l coins, each of which she can spend unlinkably. The complexity of the withdrawal operation is $\mathcal{O}(k^4)$, the spend operation is $\mathcal{O}(k^3)$, where k is security parameter. The user's wallet can be stored using $\mathcal{O}(k)$ bits. Our scheme also offers exculpability of users, that is, the bank can prove to third parties that a user has double-spent. Our scheme is secure in the random oracle model.

Keywords: electronic cash, anonymity, unlinkability, traceability

1 Introduction

1.1 Background

Electronic cash was proposed by Chaum [2][3], and has been extensively studied [4][5][6][7][8][9][10][11] [12][13].

As a coin is represented by data, and it is easy to duplicate data, an electronic cash scheme requires a mechanism that prevents a user from spending the same coin twice (double-spending). There are two scenarios. In the *on-line* scenario, the bank is on-line in each transaction to ensure that no coin is spent twice, and each merchant must consult the bank before accepting a payment. In the *off-line* scenario, the merchant accepts a payment autonomously, and later submits the payment to the bank; the merchant is guaranteed that such a payment will be either honored by the bank, or will lead to the identification (and therefore punishment) of the double-spender.

In this paper, we give an off-line 2^l-spendable unlinkable electronic cash scheme. Our framework is based on [15] by Camenisch.

1.2 Our Result

This paper proposes a new efficient unlinkable off-line electronic cash scheme secure in the random oracle model. The security proof of our scheme depends on the strong RSA assumption and the strong SDH assumption.

2 Preliminaries

2.1 Definition of Off-line E-Cash System

Our electronic cash scenario consists of three usual players: the user:$\mathcal{U}$, the bank:$\mathcal{B}$, and the merchant:$\mathcal{M}$; together with the algorithms: BKeygen,UKeygen, MKeygen, Withdraw,Spend,Deposit, Identify,Trace and VerifyOwnership.

- BKeygen is a key generation algorithm for the bank $\mathcal{B}$. It takes as input k bit security parameter, and outputs the key pair, $(\mathsf{pk_B}, \mathsf{sk_B})$.
- UKeygen is a key generation algorithm for the user $\mathcal{U}$. It takes as input k bit security parameter, and outputs the key pair, $(\mathsf{pk_U}, \mathsf{sk_U})$.
- Withdraw is a protocol between $\mathcal{U}$ and $\mathcal{B}$. $\mathcal{U}$ withdraws a 2^l unit wallet:$\mathcal{W}$ with serial number S. $\mathcal{U}$ sends signature $\mathcal{Q}$ to $\mathcal{B}$. $\mathcal{B}$ records $\mathcal{Q}$ in database:$\mathcal{D}$ to trace users double spending some coin. $\mathcal{U}$ receives $\mathcal{B}$'s signature.
- Spend is a protocol between $\mathcal{U}$ and $\mathcal{M}$. $\mathcal{U}$ sends zero-knowledge proof of knowledge of $\mathcal{W}$:Φ to $\mathcal{M}$.
- Deposit is a protocol between $\mathcal{M}$ and $\mathcal{B}$. $\mathcal{M}$ sends Φ to $\mathcal{B}$. $\mathcal{B}$ verifies Φ. If the coin has been received already, $\mathcal{B}$ rejects Φ. Otherwise, $\mathcal{B}$ accepts it.
- Identify is an algorithm to find double-spender $\mathcal{U}'$ from double spent coin Φ_1,Φ_2.
- Trace is an algorithm to output evidence:Π which $\mathcal{B}$ computes from Φ_1,Φ_2 and $\mathcal{D}$ to be used in the VerifyOwnership step.
- VerifyOwnership is an algorithm to confirm that $\mathcal{U}'$ certainly spent coin Φ_1,Φ_2. Anyone can verify double spent coin with serial number S using Π.

2.2 Definition of Security

2.2.1 Balance

Adversary $\mathcal{A}$ plays the following game:

$\mathcal{A}$ executes the Withdraw and Deposit protocols with the bank as many times as desired. (It can simulate running the Spend protocol with itself.)

$\mathcal{A}$ wins the game if the honest bank accepts a coin which differs from any coin got through the Withdraw protocol.

No probabilistic polynominal-time adversary succeeds in this game with non-negligible probability.

* Kyoto University
† Kyoto University, NTT Labotatories, NTT Corporation, 1-1 Hiakri-no-oka, Yokosuka, Kanagawa, 239-0847 Japan

2.2.2 Identification of double-spenders

Adversary $\mathcal{A}$ plays the following game:

$\mathcal{A}$ executes the Withdraw and Spend protocols with the bank as many times as desired.

$\mathcal{A}$ wins the game if the honest merchant cannot output $\mathcal{A}$'s secret key when $\mathcal{A}$ uses multiple coins with the same serial number.

No probabilistic polynominal-time adversary succeeds in this game with non-negligible probability.

2.2.3 Trace of double-spenders

Adversary $\mathcal{A}$ plays the following game:

$\mathcal{A}$ executes the Withdraw and Spend protocols with the bank as many times as desired.

$\mathcal{A}$ executes Spend protocols, the honest merchant accepts double spent coins (S, Φ_1),(S, Φ_2). The bank outputs (S', Π) by Trace.

$\mathcal{A}$ wins the game if $S \neq S'$ or VerifyOwnership(S, Π) returns reject.

No probabilistic polynominal-time adversary succeeds in this game with non-negligible probability.

2.2.4 Anonymity of users

Adversary $\mathcal{A}$ plays the following game:

$\mathcal{A}$ sets $\mathsf{pk_B}, \mathsf{sk_B}$. Honest users $\mathcal{U}_0$, $\mathcal{U}_1$ execute the withdraw protcol, and get wallet $\mathcal{W}_0$, $\mathcal{W}_1$, respectively.

One of $\mathcal{U}_0$ and $\mathcal{U}_1$ is now selected randomly, say $\mathcal{U}_b$. $\mathcal{U}_b$ executes the spend protocol. $\mathcal{A}$ outputs $b' = 0$ or 1.

$$\mathsf{Adv}_{\mathcal{A}}^{Anonymity} := 2Pr[b = b'] - 1$$

No probabilistic polynominal-time adversary's $\mathsf{Adv}_{\mathcal{A}}^{Anonymity}$ is non-negligible probability.

2.2.5 Exculpability

Exculpability guarantees that only users who really are guilty of double spending are convicted of double spending.

Adversary $\mathcal{A}$ plays the following game:

$\mathcal{A}$ sets $\mathsf{pk_B}, \mathsf{sk_B}$. An honest $\mathcal{U}$ executes withdraw and spend protocols as many times as $\mathcal{A}$ wishes.

$\mathcal{A}$ wins the game if $\mathcal{A}$ outputs (S, Π) of user $\mathcal{U}$ such that
VerifyOwnership(S, Π) returns accept.

No probabilistic polynominal-time adversary succeeds in this game with non-negligible probability.

2.3 Bilinear Maps

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be two cyclic groups of prime order p, where possibly $\mathbb{G}_1 = \mathbb{G}_2$. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$. ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2)$. e is a non-degenerate bilinear map. $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_3| = p$, i.e.,

- (Bilinear): for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, for all $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$
- (Non-degenerate): $e(g_1, g_2) \neq 1$ $(i.e., e(g_1, g_2)$ is a generator of $\mathbb{G}_T)$.
- (Efficient): e, ψ and the group in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ can be computed efficiently.

2.4 Verifiable Encryption

In Section 4.2, we apply a technique by Camenisch and Damgard [14] for turning any semantically secure encryption scheme into a verifiable encryption scheme. A verifiable encryption scheme is a two-party protocol between a prover and encryptor $\mathcal{U}$ and a verifier and receiver $\mathcal{B}$.

In the follwing, a verifiable encryption of a committed value is shown, in which ElGamal encryption is applied for keys using bilinear maps.

2.4.1 Encryption and Decryption

$\tilde{g} \in \mathbb{G}_1$ and $g, f, h \in \mathbb{G}_2$ are public data. $\mathcal{U}$ randomly chooses $u \in \mathbb{Z}_p^*$ and computes $e(\tilde{g}, g^u) = e(\tilde{g}, g)^u$. $(\mathsf{p_k}, \mathsf{s_k}) := (e(\tilde{g}, g)^u, g^u)$. Let m be the plaintext and c the cyphertext.

$$\begin{aligned} \text{Encrypt :} \quad & \mathcal{U} \text{ randomly chooses } k \in \mathbb{Z}_p^*. \\ & c := (c_1, c_2) = (\tilde{g}^k, {\mathsf{p_k}}^k m). \\ \text{Decrypt :} \quad & m = \frac{c_2}{e(c_1, g^u)} \end{aligned}$$

2.4.2 A Verifiable Encryption Scheme

$A := \tilde{g}^u \tilde{f}^t \tilde{h}^s$ is a commitment to s. $E(s) := (e_{\bar{a}}, c_{a1}||c_{a2}||c_{a3})$ is an encryption of s. $\mathcal{U}$ randomly chooses $r_1, r_2, r_3, k_1, k_2 \in \mathbb{Z}_p^*$. $\mathcal{U}$ computes

$$\begin{aligned} X &= \tilde{g}^{r_1} \tilde{f}^{r_2} \tilde{h}^{r_3} \\ c_{11} &= r_1 + u \bmod p \\ c_{12} &= r_2 + t \bmod p \\ c_{13} &= r_3 + s \bmod p \\ c_{21} &= r_1 + 2u \bmod p \\ c_{22} &= r_2 + 2t \bmod p \\ c_{23} &= r_3 + 2s \bmod p \\ e_1 &= e(\tilde{g}^{k_1}, {\mathsf{p_k}}^{k_1}(c_{11}||c_{12}||c_{13})) \\ e_2 &= e(\tilde{g}^{k_2}, {\mathsf{p_k}}^{k_2}(c_{21}||c_{22}||c_{23}))\ , \end{aligned}$$

and sends (X, e_1, e_2) to $\mathcal{B}$.

$\mathcal{B}$ returns to $a = \{1 \text{ or } 2\}$ randomly.

$\mathcal{U}$ sends (c_{a1}, c_{a2}, k_a) to $\mathcal{V}$. Let $\bar{a} = \{1 \text{ if } a = 2, 2 \text{ if } a = 1\}$.

$\mathcal{B}$ verifies

$$\begin{aligned} e_a &= (\tilde{g}^{k_a}, {\mathsf{p_k}}^{k_a}(c_{a1}||c_{a2}||c_{a3})) \\ \tilde{g}^{c_{a1}} \tilde{f}^{c_{a2}} \tilde{h}^{c_{a3}} &= XA^a \\ E(s) &= (e_{\bar{a}}, c_{a1}||c_{a2}||c_{a3}) \end{aligned}$$

By decripting $e_{\bar{a}}$, $\mathcal{B}$ obtains $c_{\bar{a}_1}$, $c_{\bar{a}_2}$, and $c_{\bar{a}_3}$ and calculates s by $s := c_{23} - c_{13} \bmod p$.

This protocol is repeated k times, $\mathcal{U}$ succeds in cheating $\mathcal{B}$ with probability $\frac{1}{2^k}$.

2.5 Committed Number Lies in an Interval

In Section 4.3, we apply a technique proposed by Boudot [1] to prove Committed Number:J belongs to $[0, 2^l)$. This requires the strong RSA assumption.

2.6 Signature Scheme

In Section 4.2, 4.3, 4.7, we apply a signature scheme proposed by Okamoto [16] to achieve anonymity and traceability.

2.6.1 Key generation

Randomly select generators $g_2, u_2, v_2 \in \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$ and $v_1 \leftarrow \psi(v_2)$. Randomly select $x \in \mathbb{Z}_p^*$ and compute $w_2 \leftarrow {g_2}^x \in \mathbb{G}_2$. The public and secret keys are:
Public key : $g_1, g_2, w_2, u_2, v_2,$
Secret key : x

2.6.2 Signature generation

Let $m \in \mathbb{Z}_p^*$ be the message to be signed. Signer $\mathcal{S}$ randomly selects r and s from $\mathbb{Z}_p^*$ and computes

$$\sigma \leftarrow ({g_1}^m u_1 {v_1}^s)^{\frac{1}{x+r}} .$$

(σ, r, s) is the signature of m.

2.6.3 Signature verification

Given public-key $(g_1, g_2, w_2, u_2, v_2)$, message m, and signature (σ, r, s), check that $m, r, s \in \mathbb{Z}_p^*$, $\sigma \in \mathbb{G}_1$, $\sigma \neq 1$, and

$$e(\sigma, w_2 {g_2}^r) = e(g_1, {g_2}^m u_2 {v_2}^s) .$$

If they hold, the verification result is **valid**; otherwise the result is **invalid**.

2.6.4 Definition of Secure Signature Schemes

In this section we recall the standard notion of security, existential unforgeability against chosen message attacks [17] as well as a slightly stronger notion of security for a signature scheme, strong existential unforgeability against chosen message attacks [18]. To define existential unforgeability, we introduce the folloing game among adversary $\mathcal{A}$ and honest signer $\mathcal{S}$.

1. Key setup:
   Run key generation algorithm $\mathcal{G}(1^n)$ to obtain a pair of public-key and secret-key $(\mathsf{pk}, \mathsf{sk})$. pk is given to adversary $\mathcal{A}$, and $(\mathsf{pk}, \mathsf{sk})$ is given to signer $\mathcal{S}$.

2. Queries to signing oracle:
   $\mathcal{A}$ adaptively requests $\mathcal{S}$ (or signing oracle) to sign on at most q_s message of his choice $m_1, \ldots, m_{q_s}$, $\mathcal{S}$ responds to m_i with a signature $\sigma_i = \mathcal{S}(\mathsf{sk}, m_i)$

3. Output:
   Eventually, $\mathcal{A}$ outputs pair $(m.\sigma)$. $\mathcal{A}$ wins the game if m is not any of $m_i (i = 1, \ldots, q_s)$ and $\mathcal{V}(\mathsf{pk}, m, \sigma) = \mathsf{accept}$. We define $\mathsf{Adv}_{\mathcal{S}}^{unforge}$ to be the probability that $\mathcal{A}$ wins the above game, taken over the coin tosses made by $\mathcal{A}$, $\mathcal{G}$ and $\mathcal{S}$.

Definition: (Existential Unforgeability) Adversary $\mathcal{A}(t, q_s, \epsilon)$-$forges$ a signature scheme if $\mathcal{A}$ runs in time at most t. $\mathcal{A}$ makes at most q_s queries to $\mathcal{S}$, and $\mathsf{Adv}_{\mathcal{S}}^{unforge}$ is at least ϵ. A signature scheme is (t, q_s, ϵ)– *existentially-unforgeable* under adaptive chosen message attacks if no adversary $\mathcal{A}(t, q_s, \epsilon)$-$forges$ the scheme.

3 Assumptions

3.1 Strong RSA Assumption:

Given an RSA module $\mathbf{n}$ and a random element $\mathbf{g} \in \mathbb{Z}_n^*$, it is hard to compute $\mathbf{h} \in \mathbb{Z}_n^*$ and integer $e > 1$ such that $\mathbf{h}^e \equiv \mathbf{g} \bmod \mathbf{n}$. The module $\mathbf{n}$ is of special form $\mathbf{pq}$, where $\mathbf{p} = 2\mathbf{p}' + 1$ and $\mathbf{q} = 2\mathbf{q}' + 1$ are safe primes.

3.2 Strong Diffie-Hellman (SDH) Assumption:

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups. The q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given the $(q+2)$-tuple $(g_1, g_2, g_2^x, \ldots, g_2^{x^q})$ as input, output pair $(g_1^{\frac{1}{x+c}}, c)$ where $c \in \mathbb{Z}_p^*$. Algorithm $\mathcal{A}$ has advantage, $\mathsf{Adv}_{SDH}(q)$, in solving q-SDH in $(g_1^{\frac{1}{x+c}}, c)$ if

$$\mathsf{Adv}_{SDH}(q) \leftarrow Pr[\mathcal{A}(g_1, g_2, g_2^x, \ldots, g_2^{x^q}) = (g_1^{\frac{1}{x+c}}, c)]$$

Adversary $\mathcal{A}$ (t, ϵ)-breaks the q-SDH problem if $\mathcal{A}$ runs in time at most t and $\mathsf{Adv}_{SDH}(q)$ is at least ϵ. The (q, t, ϵ)-SDH assumption holds if no adversary $\mathcal{A}$ (t, ϵ)-breaks the q-SDH problem.

4 Proposed E-cash System

4.1 Key Generation

$H(x)$ is a collision-resistant hash function.

Bank: Upon input of security parameter. $\mathcal{B}$ randomly generates
$\{g, f, h, v_b, w_b\} \in \mathbb{G}_2$ and set $\tilde{g} \leftarrow \psi(g)$, $\tilde{f} \leftarrow \psi(f)$, $\tilde{h} \leftarrow \psi(h)$, $\tilde{v_b} \leftarrow \psi(v_b)$, $\tilde{w_b} \leftarrow \psi(w_b)$. Randomly selects $b \in \mathbb{Z}_p^*$ and computes $x_b \leftarrow g^b, y_b \leftarrow f^b, z_b \leftarrow h^b$. $\mathcal{B}$'s public key $\mathsf{p_{kB}}$ and secrets key $\mathsf{s_{kB}}$ are:
$\mathsf{p_{kB}} = \{\tilde{g}, g, \tilde{f}, f, \tilde{h}, h, \tilde{v_b}, v_b, \tilde{w_b}, w_b, x_b, y_b, z_b\}$, $\mathsf{s_{kB}} = \{b\}$.

User: $\mathcal{U}$ randomly selects $\{v_u, w_u\} \in \mathbb{G}_2, u \in \mathbb{Z}_p^*$ and computes $x_u \leftarrow h^u$, $\tilde{v}_u \leftarrow v^u$ and $\tilde{w}_u \leftarrow w^u$. $\mathcal{U}$'s public key $\mathsf{p_{kU}}$ and secret key $\mathsf{s_{kU}}$ are:
$\mathsf{p_{kU}} = \{\tilde{g}, g, \tilde{f}, f, \tilde{h}, h, \tilde{v_u}, v_u, \tilde{w_u}, w_u, x_u, e(\tilde{g}, g)^u\}$,
$\mathsf{s_{kU}} = \{u, g^u\}$.

4.2 Withdraw

1. $\mathcal{U}$ randomly selects $s', t \in \mathbb{Z}_p^*$. $\mathcal{U}$ sends $A' = \tilde{g}^u \tilde{f}^t \tilde{h}^{s'}$ to $\mathcal{B}$. $\mathcal{U}$ exeutes proof of knowledge for u. $PK[u, t, s; x_u = h^u \wedge A' = \tilde{g}^u \tilde{f}^t \tilde{h}^{s'}]$

   $\mathcal{U}$ randomly chooses $R_a, R_b, R_c \in \mathbb{Z}_p^*$. $\mathcal{U}$ computes

   $$Z_u = h^{R_a}, Z_A = g^{R_a} f^{R_b} h^{R_c}$$

   and sends to $\mathcal{B}$. $\mathcal{B}$ returns $d \in \mathbb{Z}_p^*$ randomly. $\mathcal{U}$ computes

   $$D_u = R_a + du$$
   $$D_t = R_b + dt$$
   $$D_s = R_c + ds$$

   and sends to $\mathcal{B}$. $\mathcal{B}$ verifies by

   - $h^{D_u} = Z_u {x_u}^d$
   - $g^{D_u} f^{D_t} h^{D_s} = Z_A (A')^d$

$\mathcal{B}$ randomly selects $r' \in Z_p^*$, and sends it to $\mathcal{U}$. $\mathcal{U}$ sets $s = r' + s'$. $\mathcal{U}$ and $\mathcal{B}$ locally compute $A = \tilde{g}^u \tilde{f}^t \tilde{h}^s = A' \tilde{h}^{r'}$ each other.

2. $\mathcal{U}$ and $\mathcal{B}$ execute the verifiable encryption protocol k times. $\mathcal{U}$ randomly selects $s_{i1}, s_{i2} \in Z_p^*$. $\mathcal{U}$ computes signature $\tau_{\mathcal{U}i} = (\tilde{g}^{H(E(s)_i)} \tilde{v_u} \tilde{w_u}^{s_{i1}})^{\frac{1}{u+s_{i2}}}$ for $E(s)_i := (e_{\tilde{a}}^{(i)}, c_{a_1}^{(i)} || c_{a_2}^{(i)} || c_{a_3}^{(i)})$. $\mathcal{B}$ verifies signature $\tau_{\mathcal{U}i}$ by

$$e\big(\tau_{\mathcal{U}i}, x_u h^{s_{i2}}\big) = e(\tilde{h}, g^{H(E(s)_i)} v_u {w_u}^{s_{i1}}) \ .$$

$\mathcal{B}$ accepts

$$Q = (Q_1, \ldots, Q_k) \ .$$
$$\Big(Q_i = \big(E(s)_i, \tau_i := (\tau_{\mathcal{U}i}, s_{i1}, s_{i2})\big)\Big)$$

3. $\mathcal{B}$ randomly selects $r_1, r_2 \in Z_p^*$. $\mathcal{B}$ computes $\sigma_{\mathcal{B}} = (A \tilde{v_b} \tilde{w_b}^{r_1})^{\frac{1}{b+r_2}}$, and sends $\sigma := \{\sigma_{\mathcal{B}}, r_1, r_2\}$ to $\mathcal{U}$. $\mathcal{B}$ records the entry $(\mathsf{pk_U}, Q, \sigma)$ in his database $\mathcal{D}$. $\mathcal{U}$ verifies signature σ by

$$e\big(\sigma_{\mathcal{B}}, x_b y_b z_b (gfh)^{r_2}\big) = e(\tilde{g}\tilde{f}\tilde{h}, g^u f^t h^s v_b {w_b}^{r_1}) \ .$$

4. $\mathcal{U}$ saves the wallet $\mathcal{W} = (s, t, \sigma, J)$, where J is an l-bit counter initially set to zero.

4.3 Spend

1. $\mathcal{U}$ receives spending data I including merchant infomation. $\mathcal{U}$ computes $R = H(I)$.

2. $\mathcal{U}$ sends

$$S = g^{\frac{1}{s+J}}$$
$$T = g^{u+\frac{R}{t+J}}$$

to $\mathcal{M}$.

3. $\mathcal{U}$ chooses $R_1, \ldots, R_{13} \in Z_p^*$ randomly. $\mathcal{U}$ executes below zero knowledge proof of knowledge protocols.

$$PK[(J, R_J') : \mathbf{Y_J} = \mathbf{g}^J \mathbf{h}^{R_{J'}} \mathrm{mod}\ \mathbf{n}$$
$$\wedge\ Y_J = g^J h^{R_J} \wedge\ 0 \leq J < 2^l] \ \ [1]$$
$$PK[s, R_s; Y_s = h^s g^{R_s}]$$
$$PK[t, R_t; Y_t = f^t h^{R_t}]$$
$$PK[u, R_u; Y_u = g^u f^{R_u}]$$
$$PK[J, R_J; Y_J = g^J f^{R_J}]$$
$$PK[R_9, R_{10}; X_\alpha = x_b {y_b}^{R_9} (gfh)^{R_{10}}]$$
$$PK[R_2, R_4, R_6, R_{11}, R_{12}, R_{13}; X_{\beta_1} = g^{R_{11}}$$
$$\wedge\ X_{\beta_2} = g^{(-R_6 R_{11} + R_4 R_{11} + R_2 R_{11})} v_b^{R_{12}} w_b^{R_{13}}$$
$$\wedge\ X_{\beta_3} = g^{R_2 + R_4 + R_6}]$$
$$PK[J, s; S = g^{\frac{1}{s+J}}]$$
$$PK[u, t, J; T = g^{u+\frac{R}{t+J}}]$$

$\mathcal{U}$ computes

$$\begin{aligned}
{\sigma_B}' &= {\sigma_B}^\eta \\
\alpha &= \{x_b y_b (gfh)^{r_2}\}^{\frac{\theta}{\eta}} \\
\beta &= \{g^u f^t h^s v_b {w_b}^{r_1}\}^\theta \\
X_s &= h^{R_1} g^{R_2} \\
X_t &= f^{R_3} h^{R_4} \\
X_u &= g^{R_5} f^{R_6} \\
X_J &= g^{R_7} f^{R_8} \\
X_\alpha &= (x_b y_b)^{R_9} (gfh)^{R_{10}} \\
X_{\beta_1} &= (gfh)^{R_{11}} \\
X_{\beta_2} &= g^{-R_5 R_{11}} f^{-R_3 R_{11}} h^{-R_1 R_{11}} v_b^{R_{12}} w_b^{R_{13}} \\
X_{\beta_3} &= g^{R_5} f^{R_3} h^{R_1} \\
X_S &= S^{R_3 + R_7} \\
X_{T_1} &= T^{R_3 + R_7} \\
X_{T_2} &= g^{R_5} \\
X_{T_3} &= g^{R_7 + R_3} \\
X_{T_4} &= g^{R_5 (R_3 + R_7)} \\
Y_s &= h^s g^{R_s} \\
Y_t &= f^t h^{R_t} \\
Y_u &= g^u f^{R_u} \\
Y_J &= g^J f^{R_J} \\
\gamma &= H(I || X_s || X_t || X_u || X_J || X_\alpha || X_{\beta_1} || X_{\beta_2} || X_{\beta_3} \\
&\quad || X_S || X_{T_1} || X_{T_2} || X_{T_3} || X_{T_4} || Y_s || Y_t || Y_u || Y_J) \\
C_s &= R_1 + \gamma s \bmod p \\
\tilde{C_s} &= R_2 + \gamma R_s \bmod p \\
C_t &= R_3 + \gamma t \bmod p \\
\tilde{C_t} &= R_4 + \gamma R_t \bmod p \\
C_u &= R_5 + \gamma u \bmod p \\
\tilde{C_u} &= R_6 + \gamma R_u \bmod p \\
C_J &= R_7 + \gamma J \bmod p \\
\tilde{C_J} &= R_8 + \gamma R_J \bmod p \\
C_\eta &= R_9 + \gamma \frac{\theta}{\eta} \bmod p \\
\tilde{C_\eta} &= R_{10} + \gamma r_2 \frac{\theta}{\eta} \bmod p \\
C_{\theta_1} &= R_{11} + \gamma \theta \bmod p \\
C_{\theta_2} &= R_{12} + \gamma^2 \theta \bmod p \\
C_{\theta_3} &= R_{13} + \gamma^2 r_1 \theta \bmod p
\end{aligned}$$

$\mathcal{U}$ sends zero knowledge proof of knowledge Φ: $({\sigma_B}', \alpha, \beta, X_s, X_t, X_u, X_J, X_\alpha, X_{\beta_1}, X_{\beta_2}, X_{\beta_3}, X_S, X_{T_1}, X_{T_2}, X_{T_3}, X_{T_4}, Y_s, Y_J, Y_t, Y_u, \gamma, C_s, \tilde{C_s}, C_t, \tilde{C_t}, C_u, \tilde{C_u}, C_J, \tilde{C_J}, C_{\theta_1}, C_{\theta_2}, C_{\theta_3})$ to $\mathcal{M}$.

4. $\mathcal{M}$ verifies Φ.

   - $X_s {Y_s}^\gamma = h^{C_s} g^{\tilde{C_s}}$
   - $X_t {Y_t}^\gamma = f^{C_t} h^{\tilde{C_t}}$
   - $X_u {Y_u}^\gamma = g^{C_u} f^{\tilde{C_u}}$
   - $X_J {Y_J}^\gamma = g^{C_J} f^{\tilde{C_J}}$
   - $e({\sigma_{\mathcal{B}}}', \alpha) = e(gfh, \beta)$
   - $X_\alpha \alpha^\gamma = (x_b y_b)^{C_\eta} (gfh)^{\tilde{C_\eta}}$

- $X_{\beta_2}\beta^{\gamma^2} X_{\beta_3}^{C_{\theta_1}}$
  $= g^{C_{\theta_1} C_u} f^{C_{\theta_1} C_t} h^{C_{\theta_1} C_s} X_{\beta_1}^{-(C_u+C_t+C_s)} {v_b}^{C_{\theta_2}} {w_b}^{C_{\theta_3}}$
- $S^{\gamma(C_t+C_J)} = X_S \; g^{\gamma}$
- $T^{\;\gamma(C_t+C_J)} {X_{T_1}}^{-\gamma}$
  $= g^{C_u(C_t+C_J)} {X_{T_2}}^{-(C_t+C_J)} \; {X_{T_3}}^{-C_u} X_{T_4} \, g^{R\gamma^2}$

$\mathcal{M}$ accepts the coin $\{S, T, \Phi, R, I\}$.

5. If $J > 2^l - 1, \mathcal{U}$ sets $J = J + 1$.

4.4 Deposit

1. $\mathcal{M}$ sends the coin $\{S, T, \Phi, R, I\}$ to $\mathcal{B}$.
2. $\mathcal{B}$ verifies Φ, and accepts the coin if the (S, R) pair hasn't been spent.

4.5 Identify

From the two coins that have the same S and different R, $\mathcal{B}$ computes $\mathsf{s_{kU}}$.

$$\left(\frac{{T_2}^{R_1}}{{T_1}^{R_2}}\right)^{(R_1-R_2)^{-1}} = g^u.$$

4.6 Trace

$\mathcal{B}$ finds $\mathsf{p_{kU}} = e(\tilde{g}, g)^u = e(\tilde{g}, g^u)$. $\mathcal{B}$ recovers double spent coin $s, J_j, S_j = g^{\frac{1}{s+J_j}}$ from $\mathcal{D}$. $\mathcal{B}$ outputs $\Pi := (s, J_j, g^u, \mathsf{p_{kU}}, Q_i)$.

4.7 Verify Ownership

Anyone can check that the user with $\mathsf{p_{kU}}$ is the owner of a coin with serial number s by

- $S = g^{\frac{1}{J_j+s}}$
- $E(s)_i = (e_{\bar{a}}^{(i)}, c_{a_1}^{(i)} || c_{a_2}^{(i)} || c_{a_3}^{(i)})$
- $e(\tau_{\mathcal{U}i}, x_u g^{s_{i2}}) = e(\tilde{g}, g^{k_{i\bar{a}}} v_u {w_u}^{s_{i1}})$

5 Sketch of Security Proof

5.1 Balance

Let us assume that there is an adversary $\mathcal{A}$ that succeeds the balance game with non-negligible probability. From the proof of knowledge protocol, it means that $\mathcal{A}$ can generate a signature $\sigma_{\mathcal{B}}$ such that verification returns accept but $\mathcal{B}$ did not sent to $\mathcal{A}$. Using $\mathcal{A}$, we can obtain a forger of the signature scheme in [16].

5.2 Identification of double-spenders

Let us assume that there is an adversary $\mathcal{A}$ that succeeds the identification game with non-negligible probability. $\mathcal{A}$ outputs two coins C_1, C_2 with the same serial number which are accepted by honest bank. Since marchant information I_i differs in C_1 and C_2, $T_1 \neq T_2$ with a high probability. Thus, because of the correctness of the algorithm, we obtain $\mathsf{pk_U} = g^u$ from the equation in 4.5.

5.3 Trace of double-spenders

When adversary $\mathcal{A}$ spends two coins C_0, C_1 with the same serial number, these are valid coins because of balancde property. Thus the bank outputs $\mathsf{pk_U} = g^u$ from the equation in 4.5. Thus $\mathcal{A}$ wins the game only if the entry Q in Bank $\mathcal{B}$ is not correct one. It contradicts the security of ElGamal encryption or verifiable encryption.

5.4 Anonymity of users

For a honest user $\mathcal{U}_j$, we can constract a simulator $\mathcal{S}$ who does not know privte keys for $\mathcal{U}_j$ but the output is computationally indistinguishable from the output of $\mathcal{U}_j$ to adversary $\mathcal{A}$.

5.5 Exculpability

Adversary $\mathcal{A}$ wins the exculpability game if (1) $\mathcal{A}$ can forge Φ accepted by verifyownership or (2) $\mathcal{A}$ outputs two varid coins with the same serial number by two different user $\mathcal{U}_1$ and $\mathcal{U}_2$. For case (1), accepted by verifyownership includes obtaining a signature acepted by verification. It means that the signature scheme in [16] is not existentially-unforgeable and contradicts the assumption. For case (2), it is impossible to forge a coin, thus these two coins are really generated by honest $\mathcal{U}_1$ and $\mathcal{U}_2$. However, in this case, $\mathsf{pk_U}$ cannot obtained from these coins, thus verifyguilt will return reject.

6 Conclusion

References

[1] Fabrice Boudot, "Efficient Proofs that a Committed Number Lies in an Interval", Eurocrypt 2000, 2000.

[2] David Chaum, "Blind signature for untraceable payments", Crypto '82, 1982.

[3] David Chaum, "Blind signature systems", Crypto '83, 1983.

[4] David Chaum, Amos Fiat, Moni Naor "Untraceable electronic cash", Crypto '88, 1988.

[5] Matthew Franklin, Moti Yung, "Towards provably secure efficient electronic cash", Asiacrypt '96, 1996.

[6] David Chaum, Torben Pryds Pedersen "Transferred cash grows in size", Eurocrypt '92, 1992.

[7] Stefan Brands, "An efficient off-line electronic cash system based on the representation problem", CS-R9323, 1993.

[8] Jan L. Camenisch, Jean-Marc Piveteau, Markus A. Stadler, "Blind signature based on the discrete logaritm problem", Eurocrypt '94, 1994.

[9] Stefan Brands, "Rapid demonstration of linear rations connected by boolean operators", Crypto '93, 1993.

[10] Markus A. Stadler, Jean-Marc Piveteau, Jan L. Camenisch, "Fair blind signature", Eurocrypt '95, 1995.

[11] Yair Frankel, Yiannis Tsiounis, Moti Yung, "Indirect discourse proofs", Asiacrypt '96, 1996.

[12] Yiannis S. Tsiounis, "Efficient Electronic Cash", PhD thesis, 1997.

[13] Mihir Bellare, Adriana Palacio, "GQ and Schnorr Identification Schemes:Proofs of Security against Impersonation under Active and Concurrent Attacks", Crypto '02, 2002.

[14] Jan Camennish, Ivan Damgard, "Verifiable Encryption, Group Encryption, and Their Applications to Separable Group Signatures and Signature Sharing Schemes", Asiacrypt 2000, 2000.

[15] Jan Camennish, Susan Hohenberger, Anna Lysyanskaya, "Compact E-Cash", Eurocrypt 2005, 2006.

[16] Tatsuaki Okamoto, "Efficient Blind and Partially Blind Signatures Without Random Oracle", TCC 2006, 2006.

[17] Dan Boneh, X. Boyen, "Short signatures without random oracles", Springer 2004, 2004.

[18] Shafi Goldwasser, Silvio Micali, Ron L. Rivest, "A digital signature scheme secure against adaptive chosen message attackes", SIAM Journal Computing, 1988.