Cryptanalysis of One Fair E-cash System

LiHua Liu[†] Zhengjun Cao[‡]

[†]Department of Mathematics, ShangHai JiaoTong University. Shanghai, P.R. China.

[‡]Institute of System Science, Chinese Academy of Sciences. Beijing, P.R. China. *

Abstract One fair e-cash system was proposed in [1]. In this paper, we show that the system is insecure. Besides, we point out that there are two drawbacks. One is that those integer intervals for $s_i (i = 1, \cdots, 9)$ are unappropriate. The other is that the datum s_3 in signature data is redundant. Moreover, we give a minute description of the technique to shun the challenge in the scheme. We think the method is a little interesting.

Keywords Group signature, forgeability.

1 Introduction

The concept of group signature was introduced by Chaum and Heyst [2], which allows individual members to make signatures on behalf of the group. More formally, a secure group signature scheme must satisfy the following properties: unforgeability, anonymity, traceability, coalition-resistance, unlinkability, exculpability (see [3, 4] for more details).

An important application of group signature is to construct fair e-cash systems. Loosely speaking, a fair electronic cash is a system that allows customers to make payments anonymously. Moreover, under circumstances, a trusted authority can revoke the anonymity of suspicious transactions. The fair e-cash system[5] do not realize coin tracing. In order to remend it, Canard et al. proposed a fair E-cash system based on one variant of ACJT group signature scheme. The fair e-cash scheme differs from the one of Maitland and Boyd[5]: in their system, the group is formed from the customers that spend the electronic coins, whereas in the new system the group is formed from the coins themselves. The authors claimed that their system ensures traceability of double-spenders, supports coin tracing and provides coins that are unforgeable and anonymous under standard assumptions.

In the paper, we show that the scheme is not secure. It is universally forgeable. Our attack is direct and simple without any extra assumptions. Besides, we explain our techniques to shun the challenge in the scheme at full length. We think the method is a little interesting.

*‡Corresponding author's address: Institute of System Science, Chinese Academy of Sciences, Beijing, P.R. China. postcode: 100080. Tel: +86-010-82682282 E-mail: zjcamss@hotmail.com

2 Review of the E-cash system

In the simplified model, four types of parties are involved: a bank **B**, a trusted authority **T**, shops **S** and customers **C**. The fair e-cash system consists of five basic protocols, three of which are the same as in anonymous e-cash, namely a withdrawal protocol with which **C** withdraws electronic coins from **B**, a payment protocol with which **C** pays **S** with the coins he has withdrawn, and a deposit protocol with which **S** deposits the coins to **B**. The two additional protocols are conducted between **B** and **T**, namely owner tracing and coin tracing. they work as follows:

—coin tracing protocol: the bank provides the trusted authority with the view of a withdrawal protocol and asks for the information that allows to identify the corresponding coin in the deposit phase.

—owner tracing protocol: the bank provides the trusted authority with the view of a (suspect) payment and asks for the identity of the withdrawer of the coins used in this (suspect) payment.

2.1 Setup

Let $\epsilon > 1, k, l_p$ be security parameters, $\lambda_1, \lambda_2, \gamma_1, \gamma_2$ denote lengths satisfying

$$\lambda_1 > \epsilon(\lambda_2 + k) + 2, \quad \lambda_2 > 4l_p, \quad \gamma_1 > \epsilon(\gamma_2 + k) + 2, \quad \gamma_2 > \lambda_1 + 2.$$

Define

$$\Lambda :=]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[, \qquad \Gamma :=]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[.$$

Finally, let H be a collision-resistant hash function $H : \{0,1\}^* \longrightarrow \{0,1\}^k$.

2.2 Bank's setup protocol (performed once by B)

—Select random secret l_p-bits primes p', q' such that $p = 2p' + 1, q = 2q' + 1$ are primes. Set the modulus $n = pq$.

—Choose random generators $a, a_0, g, h, m \in_R QR(n)$, where $QR(n)$ is the set of all quadratic residues modulo n.

2.3 T's setup protocol (performed once by T)

—Choose $y, Y \in_R Z^*_{p'q'}$ and publish $z = g^y \mod n, Z = g^Y \mod n$.

Finally, the public key of the system is $PK = (n, a, a_0, g, h, m, z, Z)$, the bank's private key is $SK_B = (p', q')$ and T's private key is $SK_T = (y, Y)$.

2.4 Withdrawal protocol

For the sake of simplicity, we assume that there is only one coin denomination in the system. So all coins will have the same monetary value (d$).

—The withdrawal protocol has some similarities with the **Join protocol** of Ateniese et al.[1]: each coin obtained by a customer can be seen as a (new) membership certificate of the group signature scheme of Atenises et al. At the end of the protocol, the customer C obtains a coin $(x, [A, e])$ s.t. $A^e = a_0 a^x \ (mod\ n)$. The value x is only known by C. The purpose of the pair

(A_1, A_2), which is an ElGamal encryption of the message m^x under T's private key, and the proof V is to ensure the possibility of "coin tracing". B stores a^x and (A_1, A_2) in the user's entry of the withdrawal database for possible later anonymity revocation.

C		**B**
$\tilde{x} \in_R]0, 2^{\lambda_2}[, \tilde{r} \in_R]0, n^2[$		
$C_1 = g^{\tilde{x}} h^{\tilde{r}} \ (mod\ n)$		
$U = PK(\alpha, \beta : C_1 = g^\alpha h^\beta)$	$\overset{C_1, U}{- - - \longrightarrow}$	Verifies $C_1 \in QR(n)$, Verifies U
	$\overset{\tilde{\alpha}, \tilde{\beta}}{\longleftarrow - - -}$	$\tilde{\alpha}, \tilde{\beta} \in_R Z^*_{2^{\lambda_2}} \times]0, 2^{\lambda_2}[$
$x = 2^{\lambda_1} + (\tilde{\alpha}\tilde{x} + \tilde{\beta} (mod\ 2^{\lambda_2}))$		
$r \in_R \{0,1\}^{2^{l_p}},\ C_2 = a^x (mod\ n)$		
$A_1 = m^x Z^r,\ A_2 = g^r \ (mod\ n)$		
$V = PK(\alpha, \beta : C_2 = a^\alpha \wedge$		
$A_1 = m^\alpha Z^\beta \wedge A_2 = g^\beta)$		
$W = PK(\alpha, \beta, \gamma : \alpha \in] - 2^{\lambda_2}, 2^{\lambda_2}[$		
$\wedge\ C_2 / a^{2^{\lambda_1}} = a^\alpha \wedge\ C_1^{\tilde{\alpha}} g^{\tilde{\beta}} = g^\alpha (g^{2^{\lambda_2}})^\beta h^\gamma)$	$\overset{C_2, A_1, A_2, V, W}{- - - \longrightarrow}$	Verifies $C_2 \in QR(n),\ V,\ W$ Debits C's account from d \$ $e \in_R \Gamma$ a prime $A = (a_0 C_2)^{1/e} \ (mod\ n)$
Verifies $A^e = a_0 a^x \ (mod\ n)$	$\overset{A, e}{\longleftarrow - - -}$	

2.5 Payment protocol

During the payment protocol, the payment transcript tr (where tr includes various information such as the identification number of the shop, the date and time of the transaction, etc.) is signed using the group (membership) certificate (A, e) and the secret key x (obtained during the withdrawal protocol). More precisely: the customer first chooses at random $\omega, \omega_1, \omega_2, \omega_3 \in_R I_{2l_p}$ (where $I_i = \pm\{0,1\}^i$) and then computes the following equations:

$$T_1 = a^x z^\omega \ (mod\ n), \quad T_2 = g^\omega \ (mod\ n), \quad T_3 = A h^{\omega_1} \ (mod\ n),$$

$$T_4 = m^x \ (mod\ n), \quad T_5 = g^{\omega_1} h^{\omega_2} \ (mod\ n), \quad T_6 = g^e h^{\omega_3} \ (mod\ n).$$

Noting the fact that the equation of T_3 can be rewritten $a_0 = T_3^e / (a^x h^{e\omega_1}) \ (mod\ n)$ using $A^e = a_0 a^x \ (mod\ n)$. Then, putting the equation of T_5 to e, we obtain that $1 = T_5^e / (g^{e\omega_1} h^{e\omega_2}) \ (mod\ n)$. The payment protocol is then the following interactive signature of knowledge between **C** and **S**:

C		S
$r_1 \in_R I_{\epsilon(\gamma_2+k)}, \quad r_2 \in_R I_{\epsilon(\lambda_2+k)}$		
$r_3, r_7, r_8 \in_R I_{\epsilon(\gamma_1+2l_p+k+1)}$		
$r_4, r_5, r_6, r_9 \in_R I_{\epsilon(2l_p+k)}$		
$d_1 = a^{r_2} z^{r_4},\ d_2 = g^{r_4},\ d_4 = m^{r_2}$		
$d_3 = T_3^{r_1}/(a^{r_2} h^{r_7}),\ d_5 = g^{r_5} h^{r_6},$		
$d_6 = T_5^{r_1}/(g^{r_7} h^{r_8}),\ d_7 = g^{r_1} h^{r_9}\ (mod\ n)$	$\overset{(d_1,\cdots,d_7)}{- - - \longrightarrow}$	$c = H(T_1 \parallel \cdots \parallel T_6 \parallel d_1 \parallel \cdots \parallel d_7 \parallel tr)$
$s_1 = r_1 - c(e - 2^{\gamma_1})$	$\longleftarrow \overset{c}{- - - -}$	
$s_2 = r_2 - c(x - 2^{\lambda_1})$		
$s_3 = r_3 - ce\omega, \quad s_4 = r_4 - c\omega$		
$s_5 = r_5 - c\omega_1, \quad s_6 = r_6 - c\omega_2$		
$s_7 = r_7 - ce\omega_1, \quad s_8 = r_8 - ce\omega_2$		
$s_9 = r_9 - c\omega_3$	$\overset{(s_1,\cdots,s_9)}{- - - \longrightarrow}$	$d'_1 = T_1^c a^{s_2 - c2^{\lambda_1}} z^{s_4}\ (mod\ n)$
		$d'_2 = T_2^c g^{s_4}\ (mod\ n)$
		$d'_3 = a_0^c T_3^{s_1 - c2^{\gamma_1}}/(a^{s_2 - c2^{\lambda_1}} h^{s_7})\ (mod\ n)$
		$d'_4 = T_4^c m^{s_2 - c2^{\lambda_1}}\ (mod\ n)$
		$d'_5 = T_5^c g^{s_5} h^{s_6}\ (mod\ n)$
		$d'_6 = T_5^{s_1 - c2^{\gamma_1}}/(g^{s_7} h^{s_8})\ (mod\ n)$
		$d'_7 = T_6^c g^{s_1 - c2^{\gamma_1}} h^{s_9}\ (mod\ n)$
		$c \overset{?}{=} H(T_1 \parallel \cdots \parallel T_6 \parallel d'_1 \parallel \cdots \parallel d'_7 \parallel tr)$
		Verifies $s_1 \in_R I_{\epsilon(\gamma_2+k)+1}, \quad s_2 \in_R I_{\epsilon(\lambda_2+k)+1}$
		Verifies $s_3, s_7, s_8 \in_R I_{\epsilon(\gamma_1+2l_p+k+1)+1}$
		Verifies $s_4, s_5, s_6, s_9 \in_R I_{\epsilon(2l_p+k)+1}$

2.6 Deposit and tracing protocol

To be credited of the value of this coin, the shop spends the transcript of the execution of the payment protocol to the bank, which verifies, exactly as the shop did, that the signature on tr is correct (namely the signature of knowledge U). If this is successful, the bank checks for double-spending by searching if T_4 is already in its deposit database. If this value is not found, T_4 is stored in the deposit database and the payment is accepted as valid.

If T_4 has been previously used, the bank sends both transcripts to the trusted authority T. From these transcripts, T can retrieve $a^x = T_1/T_2^y\ (mod\ n)$. With a^x, the bank can identify the withdrawal session in which this value has been used and consequently can also identify the fraudulent customer.

2.7 Coin tracing

T is given a withdrawal transcript. T decrypts the ElGamal ciphertext (A_1, A_2) to obtaion the value m^x. This value be put on a blacklist for recognizing it when it is spent.

2.8 Owner tracing

T is given the values T_1 and T_2 observed in a payment. T decrypts this ciphertext to obtain the value a^x. With this value, the bank can identify a withdrawal session and consequently a customer C.

3 On the notation $I_i = \pm\{0,1\}^i$

How to understand the notation $I_i = \pm\{0,1\}^i$? One may hold that $I_i =]-2^i, 2^i[$. If so, then there is no length restriction for those numbers $r_i (0 \leq i \leq 9)$ which are randomly picked by the customer. It's impractical. We know that the length of random numbers acts as a key role in public key cryptosystems. So, we hold that the notation I_i means

$$]2^{i-1}, 2^i[\quad \text{or} \quad]-2^i, -2^{i-1}[$$

From the verifying phase, we know that $s_4 \in I_{\epsilon(2l_p+k)+1}$, i.e.,

$$\begin{aligned} s_4 \quad &\in \quad]2^{\epsilon(2l_p+k)}, 2^{\epsilon(2l_p+k)+1}[\\ &\text{or} \quad]-2^{\epsilon(2l_p+k)+1}, -2^{\epsilon(2l_p+k)}[\end{aligned}$$

But, $r_4 \in I_{\epsilon(2l_p+k)}$, i.e.,

$$\begin{aligned} r_4 \quad &\in \quad]2^{\epsilon(2l_p+k)-1}, 2^{\epsilon(2l_p+k)}[\\ &\text{or} \quad]-2^{\epsilon(2l_p+k)}, -2^{\epsilon(2l_p+k)-1}[\end{aligned}$$

Since $c \in \{0,1\}^k$, i.e., $c \in]2^{k-1}, 2^k[$, $\omega \in I_{2l_p}$, i.e.,

$$\omega \in]2^{2l_p-1}, 2^{2l_p}[, \quad \text{or} \quad]-2^{2l_p}, -2^{2l_p-1}[$$

Therefore,

$$\omega c \in]2^{k+2l_p-2}, 2^{k+2l_p}[, \quad \text{or} \quad]-2^{k+2l_p}, -2^{k+2l_p-2}[$$

$$\begin{aligned} r_4 - \omega c \quad &\in \quad]2^{\epsilon(k+2l_p)-1} + 2^{k+2l_p-2}, 2^{\epsilon(k+2l_p)} + 2^{k+2l_p}[\\ &\text{or} \quad]-2^{k+2l_p} - 2^{2l_p}, -2^{k+2l_p-2} - 2^{2l_p-1}[\\ &\text{or} \quad]2^{\epsilon(k+2l_p)-1} - 2^{k+2l_p}, 2^{\epsilon(k+2l_p)} - 2^{k+2l_p-2}[\\ &\text{or} \quad]-2^{\epsilon(k+2l_p)} + 2^{k+2l_p-2}, -2^{\epsilon(k+2l_p)-1} + 2^{k+2l_p}[\end{aligned}$$

Thus,

$$\begin{aligned} r_4 - \omega c \quad &\in \quad]2^{\epsilon(k+2l_p)-1} - 2^{k+2l_p}, 2^{\epsilon(k+2l_p)} + 2^{k+2l_p}[\\ &\text{or} \quad]-2^{\epsilon(k+2l_p)} + 2^{k+2l_p-2}, -2^{k+2l_p-2} - 2^{2l_p-1}[\end{aligned}$$

Obviously,

$$\begin{aligned}]2^{\epsilon(k+2l_p)-1} - 2^{k+2l_p}, 2^{\epsilon(k+2l_p)} + 2^{k+2l_p}[\quad &\notin \quad]2^{\epsilon(2l_p+k)}, 2^{\epsilon(2l_p+k)+1}[\\ ]-2^{\epsilon(k+2l_p)} + 2^{k+2l_p-2}, -2^{k+2l_p-2} - 2^{2l_p-1}[\quad &\notin \quad]-2^{\epsilon(2l_p+k)+1}, 2^{\epsilon(2l_p+k)}[\end{aligned}$$

That means $r_4 - \omega c$ might not belong to $I_{\epsilon(2l_p+k)+1}$.

So do $s_1, s_2, s_3, s_5, s_6, s_7, s_8, s_9$. That is to say, a member might make a false signature even if he executes the protocol well. Of course, the drawback is easy to overcome. It only needs to adjust the intervals either for s_i $(i = 1, \cdots, 9)$ or for $\omega, \omega_1, \omega_2, \omega_3, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9$.

4 Forgeability

In this section, we show that the scheme is insecure.

The attacker($\mathbf{A}$) picks random numbers[1]

$$\omega, \quad \omega_1, \quad \omega_2, \quad \omega_3, \quad r_1, \quad r_2, \quad r_3, \quad r_4, \quad r_5, \quad r_6, \quad r_7, \quad r_8, \quad r_9.$$

Computes

$$\begin{aligned} T_1 &= a^{\omega_3} z^{\omega} \ (mod\ n), & T_2 &= g^{\omega} \ (mod\ n), \\ T_3 &= a_0 a^{\omega_3} h^{\omega_1} \ (mod\ n), & T_4 &= m^{\omega_3} \ (mod\ n), \\ T_5 &= g^{\omega_1} h^{\omega_2} \ (mod\ n), & T_6 &= g h^{\omega_1} \ (mod\ n). \end{aligned}$$

and

$$\begin{aligned} d_1 &= a^{r_2} z^{r_4} \ (mod\ n), & d_2 &= g^{r_4} \quad (mod\ n), \\ d_3 &= a_0^{r_1} a^{\omega_3 r_1 - r_2} / h^{r_7} \ (mod\ n), & & \\ d_4 &= m^{r_2} \ (mod\ n), & d_5 &= g^{r_5} h^{r_6} \quad (mod\ n), \\ d_6 &= 1/(g^{r_7} h^{r_8}) \ (mod\ n), & d_7 &= g^{r_1} h^{r_9} \quad (mod\ n). \end{aligned}$$

Send $(d_1, d_2, d_3, d_4, d_5, d_6, d_7)$ to $\mathbf{S}$.

$\mathbf{S}$ computes

$$c = H(T_1 \parallel T_2 \parallel T_3 \parallel T_4 \parallel T_5 \parallel T_6 \parallel d_1 \parallel d_2 \parallel d_3 \parallel d_4 \parallel d_5 \parallel d_6 \parallel d_7 \parallel tr)$$

and send the challenge value c to $\mathbf{A}$.

$\mathbf{A}$ calculates

$$\begin{aligned} s_1 &= r_1 - c + c2^{\gamma_1} \ (mod\ n), & s_2 &= r_2 - \omega_3 c + c2^{\lambda_1} \ (mod\ n), \\ s_3 &= r_3 \ (mod\ n), & s_4 &= r_4 - c\omega \ (mod\ n), \\ s_5 &= r_5 - \omega_1 c \ (mod\ n), & s_6 &= r_6 - \omega_2 c \ (mod\ n), \\ s_7 &= r_7 + \omega_1 (r_1 - c) \ (mod\ n), & s_8 &= r_8 + \omega_2 (r_1 - c) \ (mod\ n), \\ s_9 &= r_9 - \omega_1 c \ (mod\ n). & & \end{aligned}$$

[1]We don't list out the integer sets for these random numbers because those integer intervals for s_i $(i = 1, \cdots, 9)$ in original scheme are unappropriate. By the above analysis of s_4, we know that the process to pick those random numbers is tedious. In fact, the security of the scheme is based on the challenge (hash value), not on those intervals restriction for $\omega, \omega_1, \omega_2, \omega_3, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9$.

Finally, the group signature is

$$(c, T_1, T_2, T_3, T_4, T_5, T_6, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, tr)$$

Correctness:

$$\begin{aligned}
d'_1 &= T_1^c a^{s_2 - c2^{\lambda_1}} z^{s_4} = (a^{\omega_3} z^{\omega})^c a^{r_2 - c\omega_3} z^{r_4 - c\omega} = a^{r_2} z^{r_4} = d_1 \ (mod\ n),\\
d'_2 &= T_2^c g^{s_4} = g^{\omega c} g^{r_4 - c\omega} = g^{r_4} = d_2 \ (mod\ n),\\
d'_3 &= a_0^c T_3^{s_1 - c2^{\gamma_1}} / (a^{s_2 - c2^{\lambda_1}} h^{s_7}) = a_0^c (a_0 a^{\omega_3} h^{\omega_1})^{r_1 - c} / (a^{r_2 - \omega_3 c} h^{r_7 + \omega_1 (r_1 - c)})\\
&= a_0^{r_1} a^{\omega_3 r_1 - r_2} / h^{r_7} = d_3 \ (mod\ n),\\
d'_4 &= T_4^c m^{s_2 - c2^{\lambda_1}} = m^{\omega_3 c} m^{r_2 - \omega_3 c} = m^{r_2} = d_4 \ (mod\ n),\\
d'_5 &= T_5^c g^{s_5} h^{s_6} = (g^{\omega_1} h^{\omega_2})^c g^{r_5 - \omega_1 c} h^{r_6 - \omega_2 c}\\
&= g^{r_5} h^{r_6} = d_5 \ (mod\ n),\\
d'_6 &= T_5^{s_1 - c2^{\gamma_1}} / (g^{s_7} h^{s_8}) = (g^{\omega_1} h^{\omega_2})^{r_1 - c} / (g^{r_7 + \omega_1 (r_1 - c)} h^{r_8 + \omega_2 (r_1 - c)})\\
&= 1/(g^{r_7} h^{r_8}) = d_6 \ (mod\ n),\\
d'_7 &= T_6^c g^{s_1 - c2^{\gamma_1}} h^{s_9} = (g h^{\omega_1})^c g^{r_1 - c} h^{r_9 - \omega_1 c} = g^{r_1} h^{r_9} = d_7 \ (mod\ n).
\end{aligned}$$

As for checking

$$s_1 \in_R I_{\epsilon(\gamma_2 + k)+1},\ s_2 \in_R I_{\epsilon(\lambda_2 + k)+1},\ s_3, s_7, s_8 \in_R I_{\epsilon(\gamma_1 + l_p + k + 1)+1},\ s_4, s_5, s_6, s_9 \in_R I_{\epsilon(2l_p + k)+1}$$

we omit it (see the discussion in above section).

Remark 1 *Actually, the number s_3 is not used in verifying phase, it is redundant. This is another designing error.*

5 How to shun the challenge

First, by the form $d'_2 = T_2^c g^{s_4}$, we know that the challenge value c has to be counteracted in the expression. Therefore, we must assume that

$$T_2 = g^{\omega}$$

where ω is undetermined. Then we have $d'_2 = g^{\omega c} g^{s_4} = g^{\omega c + s_4}$. Set

$$r_4 := s_4 + \omega c$$

it implies $\boxed{d'_2 = g^{r_4}}$

Second, by the form of $d'_4 = T_4^c m^{s_2 - c2^{\lambda_1}}$, we know T_4 **must be of the form** m^{ω_3}, where ω_3 is undetermined. Hence, we have $d'_4 = m^{\omega_3 c + s_2 - c2^{\lambda_1}}$. Set

$$r_2 := \omega_3 c + s_2 - c2^{\lambda_1}$$

Then, we have $\boxed{d'_4 = m^{r_2}}$

By the form

$$d_1' = T_1^c a^{s_2 - c2^{\lambda_1}} z^{s_4} = T_1^c a^{r_2 - \omega_3 c} z^{r_4 - \omega c}$$

we have to set

$$T_1 = a^{\omega_3} z^{\omega}$$

Hence,

$$\boxed{d_1' = a^{r_2} z^{r_4}}$$

By the form $d_5' = T_5^c g^{s_5} h^{s_6}$, we can assume that

$$T_5 = g^{\omega_1} h^{\omega_2}$$

where ω_1, ω_2 are undetermined. Hence, we have

$$d_5' = T_5^c g^{s_5} h^{s_6} = (g^{\omega_1} h^{\omega_2})^c g^{s_5} h^{s_6} = g^{s_5 + \omega_1 c} h^{s_6 + \omega_2 c}$$

Set

$$r_5 := s_5 + \omega_1 c, \qquad r_6 := s_6 + \omega_2 c$$

we have

$$\boxed{d_5' = g^{r_5} h^{r_6}}$$

By the form $d_6' = T_5^{s_1 - c2^{\gamma_1}} / (g^{s_7} h^{s_8}) = g^{\omega_1 (s_1 - c2^{\gamma_1}) - s_7} h^{\omega_2 (s_1 - c2^{\gamma_1}) - s_8}$, we set

$$r_1 := s_1 - c2^{\gamma_1} + c, \quad r_7 := s_7 - \omega_1 (r_1 - c), \quad r_8 := s_8 - \omega_2 (r_1 - c)$$

Hence

$$\boxed{d_6' = 1/(g^{r_7} h^{r_8})}$$

By the form

$$\begin{aligned} d_3' = a_0^c T_3^{s_1 - c2^{\gamma_1}} / (a^{s_2 - c2^{\lambda_1}} h^{s_7}) &= a_0^c T_3^{r_1 - c} / (a^{r_2 - \omega_3 c} h^{r_7 + \omega_1 (r_1 - c)}) \\ &= (a_0 T_3^{-1} a^{\omega_3} h^{\omega_1})^c T_3^{r_1} a^{-r_2} h^{-r_7 - \omega_1 r_1} \end{aligned}$$

we set

$$T_3 = a_0 a^{\omega_3} h^{\omega_1}$$

Hence

$$\boxed{d_3' = a_0^{r_1} a^{\omega_3 r_1 - r_2} / h^{r_7}}$$

Finally, by the form $d_7' = T_6^c g^{s_1 - c2^{\gamma_1}} h^{s_9} = T_6^c g^{r_1 - c} h^{s_9}$, we set

$$T_6 = g h^{\omega_1}$$

Hence $d_7' = T_6^c g^{r_1 - c} h^{s_9} = (g h^{\omega_1})^c g^{r_1 - c} h^{s_9} = g^{r_1} h^{s_9 + \omega_1 c}$ Set

$$r_9 := s_9 + \omega_1 c$$

we have

$$\boxed{d_7' = g^{r_1} h^{r_9}}$$

Therefore, to shun the challenge in the scheme, it only needs to choose random numbers $\omega, \omega_1, \omega_2, \omega_3, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9$ such that $s_i (1 \leq i \leq 9)$ satisfy the corresponding restrictions in original scheme.

6 Conclusion

In this paper, we show that the fair e-cash system proposed in [1] is insecure. Besides, we point out there are two drawbacks. One is that those intervals for $s_i (i = 1, \cdots, 9)$ are unappropriate. The other is that the number s_3 in signature data is redundant. In fact, how to design secure and efficient fair e-cash systems is still a hot problem.

References

[1] S. Canard and J. Traore. One Fair E-cash Systems Based on Group Signature Schemes. In: Information Security and Privacy (ACISP'03), LNCS 2727, pp. 237-248. Berlin: Springer-Verlag, 2003.

[2] D. Chaum and E. van Heyst. Group signatures. In: Advances in Cryptology-EUROCRYPT'91, LNCS 950, 257-265. Springer-Verlag, 1992.

[3] G. Ateniese, J.Camenisch, M.Joye, and G.Tsudik. A practical and provably secure coalition-resistant group signature scheme. In: Advances in Cryptology-CRYPTO'2000, LNCS 1880, 255-270. Springer-Verlag, 2000.

[4] Mihir Bellare, Daniele Micciancio, Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. EUROCRYPT 2003, LNCS 2656 pp.614-629, 2003.

[5] Greg Maitland and Colin Boyd. Fair electronic cash based on a group signature scheme. In: Information and Communications Security'2001, LNCS 2229, 461-465, Springer-Verlag, 2001.