On Fair E-cash Systems based on Group Signture Schemes

Sébastien Canard and Jacques Traoré

France Telecom R&D
42, rue des Coutures, BP6243
14066 Caen Cedex, France
{sebastien.canard, jacques.traore}@francetelecom.com

Abstract. A fair electronic cash system is a system that allows customers to make payments anonymously. Moreover, under certain circumstances, a trusted authority can revoke the anonymity of suspicious transactions. Various fair e-cash systems using group signature schemes have been proposed [4, 15, 16, 18]. Unfortunately, they do not realize coin tracing [4, 15, 18] (the possibility to trace the coins withdrawn by a customer). In this paper, we describe several failures in the solution of [16] and we present a secure and efficient fair e-cash system based on a group signature scheme. Our system ensures traceability of *double-spenders*, supports coin tracing and provides coins that are unforgeable and anonymous under standard assumptions.

1 Introduction

Many anonymous electronic cash systems have been proposed in the recent years. In these systems, there is no mechanism for the bank, the merchants or any other party to identify the users involved in a transaction. If desirable from a user's point of view, this unconditional anonymity could however be misused for illegal purposes, such as money laundering or perfect blackmailing.
Fair electronic cash systems have been suggested independently by [3] and [17] as a solution to prevent such fraudulent activities. The main feature of these systems is the existence of a trusted authority that can revoke, under specific circumstances, the anonymity of the coins.

Brickell et al. in [3] proposed the first fair off-line electronic cash system. Unfortunately, their scheme requires the participation of the trustee in the withdrawals of coins, which is undesirable in practice. Camenisch, Maurer and Stadler [5] and independently Frankel et al. [11] proposed fair e-cash schemes with an off-line (passive) authority: the participation of the trustee is only required in the set-up of the system and for anonymity revocation. The efficiency and the security (anonymity) of these schemes [3, 5, 11] have been later improved [12, 14]. Unfortunately, the security for the bank (namely the unforgeability of the coins) relies, in these schemes, on non-standard assumptions.
Group signature schemes have been introduced in 1991 by Chaum and van Heyst

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

[6]. They allow members to sign a document on behalf of the group in such a way that the signatures remain anonymous and untraceable for everyone but a designated authority, who can recover the identity of the signer whenever needed (this procedure is called "signature opening"). Currently, the best group signature scheme is the one of Ateniese et al. [1].
In 1999, Traoré [18] proposed a solution that combine a group signature scheme and a blind signature scheme in order to design a privacy-protecting fair off-line electronic cash system. Unfortunately, his proposal does not realize coin tracing (the possibility to trace the coins withdrawn by a customer). In 2001, Maitland and Boyd [15] proposed a variant of this solution based on the group signature scheme of Ateniese et al. [1]. Very recently, Qiu et al. [16] designed a new electronic cash system, using again a combination of a group signature scheme and a blind signature scheme. However, their solution does not work for various reasons (owing to space limitations, the cryptanalysis of [16] will appear in the full paper). Camenisch and Lysyanskaya [4] proposed a fair electronic cash system where blind signatures are not used (named one-show credentials) but they don't achieve coin tracing.

In this paper, we investigate the same way of using a group signature scheme for designing a fair off-line electronic cash system as [4] do. In fact in [15], [16] and [18], each customer is a member of a group whereas in this paper, a group certificate corresponds to a coin delivered by the bank. This implies a relatively efficient solution which is completely secure and that does not need the use of a blind signature such as other proposals [15, 16, 18]. Our way of realizing tracing after a double-spending is also different from the solution of [4].
Our paper is organized as follows. In Section 2, we describe our solution and in Section 3, we analyse the security of our proposal.

2 A New Electronic Cash System

In this section, we describe a new fair off-line e-cash scheme based on the group signature scheme of Ateniese et al. [1]. Our fair e-cash scheme however differs from the one of Maitland and Boyd [15] which is based on the same group signature scheme: in their system, the group is formed from the customers that spend the electronic coins, whereas in our system the group is formed from the coins themselves. This difference will allow us, as we will see, to easily incorporate a coin tracing mechanism.
In the simplified model of fair electronic cash that we use, four types of parties are involved: a bank B, a trusted authority T, shops S and customers C. A 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:

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

– **Coin tracing protocol:** the bank provides the trusted authority with the view of a withdrawal protocol and asks for the information that allows it 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.

The security of our scheme relies on the Strong-RSA (S-RSA) assumption (see [13]), and on the Decision Diffie-Hellman (DDH) assumption in groups of unknown order [2].

2.1 Setup

Let $\epsilon > 1$, k and l_p be security parameters (the parameter ϵ controls the tightness of the statistical zero-knowledgeness and the parameter l_p sets the size of the modulus to use). Let λ_1, λ_2, γ_1 and γ_2 denote lengths satisfying $\lambda_2 > 4l_p$, $\lambda_1 > \epsilon(\lambda_2 + k) + 2$, $\gamma_2 > \lambda_1 + 2$ and $\gamma_1 > \epsilon(\gamma_2 + k) + 2$. Let us define $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$. Finally, let H be a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$.

Bank's Setup Protocol (performed once by B):

– Select random secret l_p-bits primes p', q' such that $p = 2p'+1$ and $q = 2q'+1$ are primes. Set the modulus $n = pq$.
– Choose random generators[1] a, a_0, g, h, m of $QR(n)$ (the set of all quadratic residues modulo n).

T's Setup Protocol (performed once by T):

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

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 one is $SK_T = (y, Y)$.

2.2 Withdrawal Protocol

For the sake of simplicity, we assume that there is only one coin denomination in the system (extension to multiple denominations will be described in the full paper). So all coins will have the same monetary value (d \$).
The withdrawal protocol[2] (Fig. 1) 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 Ateniese et al. At the end

[1] It is assumed that the discrete log of these elements w.r.t. the others is unknown.
[2] In the sequel, $PK(\alpha : f(\alpha, \ldots))(M)$ will be a *signature of knowledge* on message M of a value α that verifies the predicate f. Signatures of knowledge are signatures derived from zero-knowledge proofs of knowledge using the Fiat-Shamir heuristic [10].

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

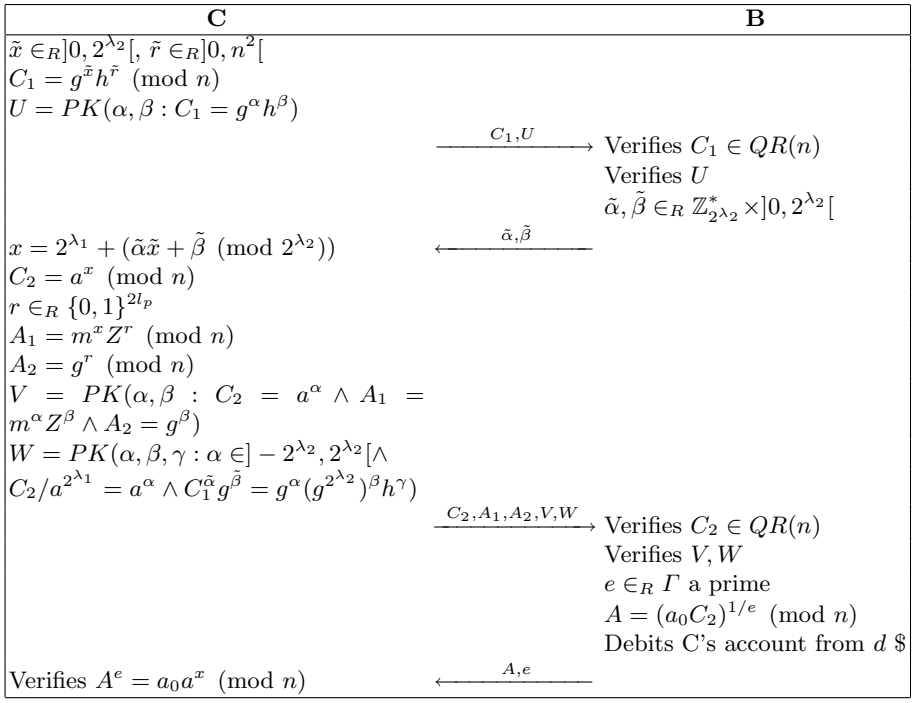

Fig. 1. The Withdrawal Protocol.

of the protocol, the customer C obtains a coin $(x, [A, e])$ s.t. $A^e = a_0 a^x \pmod n$. The value x is only known by C. The purpose of the pair (A_1, A_2), which is an El Gamal encryption [9] 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.

2.3 Payment Protocol

During the payment protocol (Fig. 2), 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 $w, w_1, w_2, w_3 \in_R I_{2l_p}$ (where $I_d = \pm\{0,1\}^d$) and then computes the following equations:

$$T_1 = a^x z^w \pmod n \quad T_2 = g^w \pmod n \qquad T_3 = Ah^{w_1} \pmod n$$
$$T_4 = m^x \pmod n \quad T_5 = g^{w_1} h^{w_2} \pmod n \quad T_6 = g^e h^{w_3} \pmod n$$

Noting the fact that the equation of T_3 can be rewritten $a_0 = T_3^e/(a^x h^{ew_1}) \pmod n$ using $A^e = a_0 a^x \pmod n$. Then, putting the equation of T_5 to e, we

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

C		S
$r_1 \in_R I_{\epsilon(\gamma_2+k)}$, $r_2 \in_R I_{\epsilon(\lambda_2+k)}$		
$r_3, r_7, r_8 \in_R I_{\epsilon(\gamma_1+2l_p+k+1)}$		
$r_4, r_5, r_6, r_9 \in_R I_{\epsilon(2l_p+k)}$		
$d_1 = a^{r_2} z^{r_4} \pmod n$		
$d_2 = g^{r_4} \pmod n$		
$d_3 = T_3^{r_1}/(a^{r_2} h^{r_7}) \pmod n$		
$d_4 = m^{r_2} \pmod n$		
$d_5 = g^{r_5} h^{r_6} \pmod n$		
$d_6 = T_5^{r_1}/(g^{r_7} h^{r_8}) \pmod n$		
$d_7 = g^{r_1} h^{r_9} \pmod n$		
	$\xrightarrow{d_1,\ldots,d_7}$	$c = H(T_1 \| \ldots \| T_6 \| d_1 \| \ldots \| d_7 \| tr)$
	$\xleftarrow{c}$	
$s_1 = r_1 - c(e - 2^{\gamma_1})$		
$s_2 = r_2 - c(x - 2^{\lambda_1})$		
$s_3 = r_3 - cew$		
$s_4 = r_4 - cw$		
$s_5 = r_5 - cw_1$		
$s_6 = r_6 - cw_2$		
$s_7 = r_7 - cew_1$		
$s_8 = r_8 - cew_2$		
$s_9 = r_9 - cw_3$ (all in $\mathbb{Z}$)		
	$\xrightarrow{s_1,\ldots,s_9}$	Verifies $c = H(T_1 \| \ldots \| T_6 \| T_1^c a^{s_2 - c2^{\lambda_1}} z^{s_4} \|$
		$T_2^c g^{s_4} \| a_0^c T_3^{s_1 - c2^{\gamma_1}}/(a^{s_2 - c2^{\lambda_1}} h^{s_7}) \|$
		$T_4^c m^{s_2 - c2^{\lambda_1}} \| T_5^c g^{s_5} h^{s_6} \|$
		$T_5^{s_1 - c2^{\gamma_1}}/(g^{s_7} h^{s_8}) \| T_6^c g^{s_1 - c2^{\gamma_1}} h^{s_9} \| tr)$
		Verifies $s_1 \in I_{\epsilon(\gamma_2+k)+1}$, $s_2 \in I_{\epsilon(\lambda_2+k)+1}$
		Verifies $s_3, s_7, s_8 \in I_{\epsilon(\gamma_1+2l_p+k+1)+1}$
		Verifies $s_4, s_5, s_6, s_9 \in I_{\epsilon(2l_p+k)+1}$

Fig. 2. The Payment Protocol.

obtain that $1 = T_5^e/(g^{ew_1} h^{ew_2}) \pmod n$. The payment protocol is then the following interactive signature of knowledge between C and S (see Fig. 2):

$$U = PK(\ \alpha, \beta, \gamma, \delta, \zeta, \eta, \theta, \iota, \kappa : T_1 = a^\beta z^\delta \wedge T_2 = g^\delta \wedge a_0 = T_3^\alpha/(a^\beta h^\theta) \wedge$$
$$T_4 = m^\beta \wedge T_5 = g^\zeta h^\eta \wedge 1 = T_5^\alpha/(g^\theta h^\iota) \wedge T_6 = g^\alpha h^\kappa).$$

2.4 Deposit and Tracing Protocols

To be credited of the value of this coin, the shop sends 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[3] by searching if T_4

[3] i.e., using the same coin in two different transactions. In other words, the bank tries to determine whether the group certificate (A, e) and the secret key x, underlying this payment transaction, have already been used or not.

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

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[4].
If T_4 has been previously used, the bank sends both transcripts to the trusted authority T. From these transcripts, T can retrieve $a^x = T_1/T_2^y \pmod n$. 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.

Coin Tracing. T is given a withdrawal transcript. T decrypts the El Gamal ciphertext (A_1, A_2) to obtain the value m^x. This value can be put on a blacklist for recognizing it when it is spent.

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 Security Analysis

We focus on the main security requirements of an electronic cash system: one-more unforgeability[5] and anonymity (the inability for anyone, except T, to match a transaction with a user).

3.1 Unforgeability

Theorem 1. *Under the S-RSA assumption, a probabilistic polynomial-time (PPT) adversary cannot, after initiating polynomially many withdrawal sessions, output, with non-negligible probability (in l_p), a coin $(x, [A, e])$ with $x \in \Lambda$ and $e \in \Gamma$ that is different from all the coins obtained in the withdrawal sessions (where the withdrawal sessions can be performed in an adaptative and arbitrary interleaving manner).*

Proof. Let $\mathcal{M}$ be an attacker who can adaptively run the withdrawal protocol so as to obtain the coins $(x_j, [A_j, e_j]), j = 1, \ldots, l$ with $x_j \in \Lambda$, $e_j \in \Gamma$ and $A_j = (a_0 a^{x_j})^{1/e_j} \pmod n$ and then can output $(\hat{x}, [\hat{A}, \hat{e}])$ with $\hat{x} \in \Lambda$, $\hat{e} \in \Gamma$, $\hat{A} = (a_0 a^{\hat{x}})^{1/\hat{e}} \pmod n$ and $(\hat{x}, \hat{e}) \neq (x_j, e_j)$ for all $1 \leq j \leq l$ with a non negligible probability.
Given a pair (n, v), we randomly repeat one of the two following algorithms with $\mathcal{M}$ and we hope to succeed in computing a pair $(u, d) \in \mathbb{Z}_n^* \times \mathbb{Z}_{>1}$ such that $u^d = v \pmod n$ from $\mathcal{M}$'s answers.

- First algorithm:
  1. Choose $x_1, \ldots, x_l \in_R \Lambda$ and $e_1, \ldots, e_l \in_R \Gamma$.
  2. Compute $a = v^{\prod_{1 \leq k \leq l} e_k} \pmod n$.

[4] This technique has been first introduced in [18] and subsequently used by Camenisch and Lysyanskaya [4] for their one-show credentials scheme.
[5] which means that is must be infeasible to create more than l coins from l withdrawals.

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

3. Choose $r \in_R \Lambda$ and compute $a_0 = a^r \pmod n$.
4. $\forall 1 \leq i \leq l$, compute $A_i = v^{(x_i+r)\prod_{k\neq i} e_k} \pmod n$.
5. Choose $g, h, m \in_R QR(n)$ and $y, Y \in_R \{1, \ldots, n^2\}$ and compute $z = g^y \pmod n$ and $Z = g^Y \pmod n$.
6. Run the withdrawal protocol l times with $\mathcal{M}$ and with $(n, a, a_0, g, h, m, z, Z)$ as input. At the i-th run, we receive C_1 and U from $\mathcal{M}$. Use the proof of knowledge U to extract $\tilde{x}_i$ and $\tilde{r}_i$ such that $C_1 = g^{\tilde{x}_i} h^{\tilde{r}_i} \pmod n$ (rewinding $\mathcal{M}$ twice for a similar commitment). Choose $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ such that the prepared x_i (see step 1.) is $x_i = 2^{\lambda_1} + (\tilde{\alpha}_i \tilde{x}_i + \tilde{\beta}_i \pmod{2^{\lambda_2}})$ Then, send $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ to $\mathcal{M}$. Follow the protocol and then send to $\mathcal{M}$ the couple $[A_i, e_i]$.
After the l withdrawals, $\mathcal{M}$ outputs $(\hat{x}, [\hat{A}, \hat{e}])$ with $\hat{x} \in \Lambda$, $\hat{e} \in \Gamma$, $\hat{A} = (a_0 a^{\hat{x}})^{1/\hat{e}} \pmod n$ and $(\hat{x}, \hat{e}) \neq (x_j, e_j)$ for all $1 \leq j \leq l$.
7. If there exists $1 \leq j \leq l$ such that $gcd(\hat{e}, e_j) \neq 1$, then output $\perp$ and quit. Else, let $\tilde{e} = (\hat{x} + r) \prod_{1 \leq k \leq l} e_k$ (and then $\hat{A}^{\hat{e}} = v^{\tilde{e}} \pmod n$). Since $gcd(\hat{e}, e_j) = 1$ for all $1 \leq j \leq l$, then $gcd(\hat{e}, \tilde{e}) = gcd(\hat{e}, (\hat{x} + r))$. Hence, by the Bezout's theorem, it exists $\alpha, \beta \in \mathbb{Z}$ such that $\alpha\hat{e} + \beta\tilde{e} = gcd(\hat{e}, (\hat{x} + r))$. Let $u = v^\alpha \hat{A}^\beta \pmod n$ and $d = \hat{e}/gcd(\hat{e}, (\hat{x} + r))$ ($\gamma_2 > \lambda_1 + 2 \Longrightarrow \hat{e} > (\hat{x} + r) \Longrightarrow d > 1$) and then $u^d = v \pmod n$. Output (u, d).

This algorithm only succeeds if $\mathcal{M}$ outputs a coin $(\hat{x}, [\hat{A}, \hat{e}])$ such that $gcd(\hat{e}, e_j) = 1$ for all $1 \leq j \leq l$. The next algorithm can find a couple (u, d) if $gcd(\hat{e}, e_j) \neq 1$ for some $1 \leq j \leq l$ (since e_j is prime, $gcd(\hat{e}, e_j) \neq 1 \Longrightarrow gcd(\hat{e}, e_j) = e_j$).

– Second algorithm:
1. Choose $x_1, \ldots, x_l \in_R \Lambda$ and $e_1, \ldots, e_l \in_R \Gamma$.
2. Choose $j \in_R \{1, \ldots, l\}$ and compute $a = v^{\prod_{k\neq j} e_k} \pmod n$.
3. Choose $r \in_R \Lambda$ and compute $A_j = a^r \pmod n$ and $a_0 = A_j^{e_j}/a^{x_j} \pmod n$.
4. $\forall 1 \leq i \leq l, i \neq j$, compute $A_i = v^{(x_i + e_j r - x_j)\prod_{k\neq i,j} e_k} \pmod n$.
5. Choose $g, h, m \in_R QR(n)$ and $y, Y \in_R \{1, \ldots, n^2\}$ and compute $z = g^y \pmod n$ and $Z = g^Y \pmod n$.
6. Similar to the step 6. of the first algorithm.
7. If $gcd(\hat{e}, e_j) \neq e_j$, then output $\perp$ and quit. Else, $\exists t / \hat{e} = t e_j$ and we can define $B = \hat{A}^t / A_j \pmod n$ if $\hat{x} \geq x_j$ and $B = A_j / \hat{A}^t \pmod n$ otherwise. Then $B = (a^{|\hat{x} - x_j|})^{1/e_j} = (v^{|\tilde{e}|})^{1/e_j} \pmod n$ with $\tilde{e} = (\hat{x} - x_j) \prod_{k\neq j} e_k$. Since $gcd(e_j, \prod_{k\neq j} e_k) = 1$, then $gcd(e_j, |\tilde{e}|) = gcd(e_j, |\hat{x} - x_j|)$. Hence, by the Bezout's theorem, it exists $\alpha, \beta \in \mathbb{Z}$ such that $\alpha e_j + \beta|\tilde{e}| = gcd(e_j, |\hat{x} - x_j|)$. Let $u = v^\alpha B^\beta \pmod n$ and $d = e_j / gcd(e_j, |\hat{x} - x_j|)$ ($\gamma_2 > \lambda_1 + 2 \Longrightarrow e_j > |\hat{x} - x_j| \Longrightarrow d > 1$) and then $u^d = v \pmod n$. Output (u, d).

Consequently, randomly running one of the two algorithms until the output is not $\perp$ permits an attacker getting access to the machine $\mathcal{M}$ to solve the S-RSA problem in expected running-time polynomial in l. As the S-RSA problem is assumed to be infeasible, we can conclude that no one can create more than l coins from l withdrawals (where l is polynomial in l_p). □

We will now prove that if S *accepts* a payment, then this implies that C necessarily knows a coin $(x, [A, e])$, with $x \in \Lambda$ and $e \in \Gamma$ s.t. $A^e = a_0 a^x \pmod n$.

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

Theorem 2. *Under the S-RSA assumption, the interactive payment protocol is a proof of knowledge of a withdrawal coin* $(x, [A, e])$.

Proof. We have to show that a knowledge extractor is able to recover the coin $(x, [A, e])$ from two accepting signatures. Let $(c, s_1, \ldots, s_9, d_1, \ldots, d_7, T_1, \ldots, T_6)$ and $(\tilde{c}, \tilde{s}_1, \ldots, \tilde{s}_9, d_1, \ldots, d_7, T_1, \ldots, T_6)$ be these two accepting tuples.
Using Lemma 1 (see below), we can show that for all $i = 1, \ldots, 9$, there exists $\theta_i \in \mathbb{Z}$ such that $s_i - \tilde{s}_i = \theta_i(\tilde{c} - c)$.
As $T_5^{\tilde{c}-c} = g^{s_5-\tilde{s}_5} h^{s_6-\tilde{s}_6} \pmod{n}$, it follows (since $\tilde{c} - c$ can be either even or odd) that there exists some v such that $T_5 = v g^{\theta_5} h^{\theta_6} \pmod{n}$ with $v^2 = 1$. Moreover, the value v must be either 1 or -1 as otherwise $gcd(v \pm 1, n)$ is a non trivial factor of n. Using d_6 and the result above, it comes:

$$\begin{aligned}
(T_5^{-2^{\gamma_1}})^{\tilde{c}-c} &= T_5^{s_1-\tilde{s}_1}/(g^{s_7-\tilde{s}_7} h^{s_8-\tilde{s}_8}) \pmod{n} \\
((v g^{\theta_5} h^{\theta_6})^{-2^{\gamma_1}})^{\tilde{c}-c} &= (v g^{\theta_5} h^{\theta_6})^{s_1-\tilde{s}_1}/(g^{s_7-\tilde{s}_7} h^{s_8-\tilde{s}_8}) \pmod{n} \\
1 &= \tilde{v} v^{\theta_1+2^{\gamma_1}} g^{\theta_5(\theta_1+2^{\gamma_1})-\theta_7} h^{\theta_6(\theta_1+2^{\gamma_1})-\theta_8} \pmod{n}
\end{aligned}$$

where $\tilde{v}^2 = 1$. Since 1, g and h are in $QR(n)$ and $v = \pm 1$, it is necessary that $\tilde{v} v^{\theta_1+2^{\gamma_1}} = 1$ (since $-1 \notin QR(n)$) and, under the fact that the discrete logarithm of g in base h is unknown, that $\theta_5(\theta_1 + 2^{\gamma_1}) = \theta_7 \pmod{p'q'}$ (as g is of order $p'q'$).
From d_3, we obtain, using similar arguments as for T_5 and this last result, that:

$$\begin{aligned}
(a_0 T_3^{-2^{\gamma_1}}/a^{-2^{\lambda_1}})^{\tilde{c}-c} &= T_3^{s_1-\tilde{s}_1}/(a^{s_2-\tilde{s}_2} h^{s_7-\tilde{s}_7}) \pmod{n} \\
a_0 T_3^{-2^{\gamma_1}}/a^{-2^{\lambda_1}} &= u T_3^{\theta_1}/(a^{\theta_2} h^{\theta_7}) \pmod{n} \\
a_0 &= u T_3^{\theta_1+2^{\gamma_1}} \left(\frac{1}{a}\right)^{\theta_2+2^{\lambda_1}} \left(\frac{1}{h^{\theta_5}}\right)^{\theta_1+2^{\gamma_1}} \pmod{n} \\
a_0 &= u \left(\frac{T_3}{h^{\theta_5}}\right)^{\theta_1+2^{\gamma_1}} \left(\frac{1}{a}\right)^{\theta_2+2^{\lambda_1}} \pmod{n}
\end{aligned}$$

where u is such that $u^2 = 1$. Again, $u = \pm 1$ as otherwise $gcd(u \pm 1, n)$ is a non trivial factor of n. Let us note $\pi_1 = \theta_1 + 2^{\gamma_1}$, $\pi_2 = \theta_2 + 2^{\lambda_1}$ and $s = 1$ if $\pi_1 > 0$ and -1 otherwise (and consequently $\pi_1 = s|\pi_1|$). Then we have:

$$A^{|\pi_1|} = a_0 a^{\pi_2} \pmod{n} \text{ with } A = \begin{cases} \left(\frac{u T_3}{h^{\theta_5}}\right)^s \text{ if } \pi_1 \text{ is odd} \\ \left(\frac{T_3}{h^{\theta_5}}\right)^s \text{ if } \pi_1 \text{ is even} \end{cases}$$

The case "π_1 even" implies that $\left(\frac{T_3}{h^{\theta_5}}\right)^{\pi_1}$ is a quadratic residue modulo n: as a_0 and a are in $QR(n)$, it is then necessary that $u = 1$ since $-1 \notin QR(n)$ (and $QR(n)$ is a group).
Since $\pi_1 = \theta_1 + 2^{\gamma_1}$, $\theta_1 = \frac{s_1-\tilde{s}_1}{\tilde{c}-c}$ and $s_1, \tilde{s}_1 \in I_{\epsilon(\gamma_2+k)+1}$, we have $s_1 - \tilde{s}_1 \in I_{\epsilon(\gamma_2+k)+2}$ and since the smallest value that $\tilde{c} - c$ can take is 1 the integer π_1 must lie in $[2^{\gamma_1} - 2^{\epsilon(\gamma_2+k)+2}, 2^{\gamma_1} + 2^{\epsilon(\gamma_2+k)+2}]$. Similarly, we can prove that π_2 must lie in $[2^{\lambda_1} - 2^{\epsilon(\lambda_2+k)+2}, 2^{\lambda_1} + 2^{\epsilon(\lambda_2+k)+2}]$ which is in accordance with what is expected with a signature of knowledge that proves that a discrete logarithm

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

lies in an interval (see [1]).
Consequently, by putting $x = \pi_2$, $A = \left(\frac{T_3}{h^{\theta_5}}\right)^s$ and $e = |\pi_1|$, we obtain that $(x, [A, e])$ is a valid certificate such that $A^e = a_0 a^x \pmod n$ and hence, this is a valid proof of knowledge. □

Lemma 1. *Given two accepting payment protocols (c, s_1, ..., s_9, d_1, ..., d_7, T_1, ..., T_6) and ($\tilde{c}$, $\tilde{s}_1$, ..., $\tilde{s}_9$, d_1, ..., d_7, T_1, ..., T_6) it is necessary that, for all $i = 1, \ldots, 9$, there exists $\theta_i \in \mathbb{Z}$ such that $s_i - \tilde{s}_i = \theta_i(\tilde{c} - c)$.*

Proof. From the two representations of $d_2 = T_2^c g^{s_4} = T_2^{\tilde{c}} g^{\tilde{s}_4} \pmod n$ we can write that $g^{s_4 - \tilde{s}_4} = T_2^{\tilde{c}-c} \pmod n$. Let δ_4 be the greatest common divisor (gcd) of $s_4 - \tilde{s}_4$ and $\tilde{c} - c$. By the Bezout's theorem there exists $\alpha_4, \beta_4 \in \mathbb{Z}$ such that $\alpha_4(s_4 - \tilde{s}_4) + \beta_4(\tilde{c} - c) = \delta_4$. As a consequence, we can write g as $g = g^{(\alpha_4(s_4 - \tilde{s}_4) + \beta_4(\tilde{c}-c))/\delta_4} = (T_2^{\alpha_4} g^{\beta_4})^{\frac{\tilde{c}-c}{\delta_4}} \pmod n$. If $\tilde{c} - c \neq \delta_4$ we have found a $(\frac{\tilde{c}-c}{\delta_4})^{th}$ root of g, which contradicts the S-RSA assumption. Then, $\tilde{c} - c = \delta_4 = gcd(s_4 - \tilde{s}_4, \tilde{c} - c)$ and consequently:

$$\exists \theta_4 \in \mathbb{Z} / s_4 - \tilde{s}_4 = \theta_4(\tilde{c} - c).$$

From the two representations of $d_1 = (T_1 a^{-2^{\lambda_1}})^c a^{s_2} z^{s_4} = (T_1 a^{-2^{\lambda_1}})^{\tilde{c}} a^{\tilde{s}_2} z^{\tilde{s}_4} \pmod n$ we can write that $a^{s_2 - \tilde{s}_2} = (T_1 a^{-2^{\lambda_1}})^{\tilde{c}-c} z^{\tilde{s}_4 - s_4} = (T_1 a^{-2^{\lambda_1}} z^{-\theta_4})^{\tilde{c}-c} \pmod n$. Let $\delta_2 = gcd(s_2 - \tilde{s}_2, \tilde{c} - c)$ and $\alpha_2, \beta_2 \in \mathbb{Z}$ such that $\alpha_2(s_2 - \tilde{s}_2) + \beta_2(\tilde{c} - c) = \delta_2$. Hence, we can write a as $a = a^{(\alpha_2(s_2 - \tilde{s}_2) + \beta_2(\tilde{c}-c))/\delta_2} = ((T_1 a^{-2^{\lambda_1}} z^{-\theta_4})^{\alpha_2} a^{\beta_2})^{\frac{\tilde{c}-c}{\delta_2}} \pmod n$. If $\tilde{c} - c \neq \delta_2$ we have found a $(\frac{\tilde{c}-c}{\delta_2})^{th}$ root of g, which contradicts the S-RSA assumption. Then, $\tilde{c} - c = \delta_2 = gcd(s_2 - \tilde{s}_2, \tilde{c} - c)$ and consequently:

$$\exists \theta_2 \in \mathbb{Z} / s_2 - \tilde{s}_2 = \theta_2(\tilde{c} - c).$$

Then, using the two representations of $d_5 = T_5^c g^{s_5} h^{s_6} = T_5^{\tilde{c}} g^{\tilde{s}_5} h^{\tilde{s}_6} \pmod n$, we can write that $T_5^{\tilde{c}-c} = g^{s_5 - \tilde{s}_5} h^{s_6 - \tilde{s}_6} \pmod n$. We can show (see Lemma 2 below) that it is necessary that $\tilde{c} - c$ divides both $s_5 - \tilde{s}_5$ and $s_6 - \tilde{s}_6$. As a consequence:

$$\forall i \in \{5, 6\}, \exists \theta_i \in \mathbb{Z} / s_i - \tilde{s}_i = \theta_i(\tilde{c} - c).$$

We can do the same things for d_3, d_6 and d_7 to conclude that:

$$\forall i \in \{1, 3, 7, 8, 9\}, \exists \theta_i \in \mathbb{Z} / s_i - \tilde{s}_i = \theta_i(\tilde{c} - c).$$ □

Lemma 2. *Under the S-RSA assumption, given two representations of $d = T^c g^s h^t = T^{\tilde{c}} g^{\tilde{s}} h^{\tilde{t}} \pmod n$, it is necessary that $\tilde{c} - c$ divides both $s - \tilde{s}$ and $t - \tilde{t}$.*

Proof. In fact, if $\tilde{c} - c$ does not divide both $s - \tilde{s}$ and $t - \tilde{t}$ then, there are three cases. Suppose first that $\tilde{c} - c$ divides $s - \tilde{s}$ and not $t - \tilde{t}$. Then, there exists $\theta \in \mathbb{Z}$ such that $s - \tilde{s} = \theta(\tilde{c} - c)$. From d, we can write that $h^{t - \tilde{t}} = T^{\tilde{c}-c} g^{\tilde{s}-s} = (T g^{-\theta})^{\tilde{c}-c}$. Let δ be the greatest common divisor of $t - \tilde{t}$ and $\tilde{c} - c$. By the Bezout's theorem there exists $\alpha, \beta \in \mathbb{Z}$ such that $\alpha(t - \tilde{t}) + \beta(\tilde{c} - c) = \delta$. As a consequence, we can write h as $h = h^{(\alpha(t-\tilde{t}) + \beta(\tilde{c}-c))/\delta} = ((T g^{-\theta})^{\alpha} h^{\beta})^{\frac{\tilde{c}-c}{\delta}} \pmod$

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

n). As $\tilde{c} - c \neq \delta$ ($\tilde{c} - c$ does not divide $t - \tilde{t}$) we have found a $(\frac{\tilde{c}-c}{\delta})^{th}$ root of h, which contradicts the S-RSA assumption.
Suppose then that $\tilde{c}-c$ divides $t-\tilde{t}$ and not $s-\tilde{s}$, we can do the same argument to contradict the S-RSA assumption. Finally, if $\tilde{c}-c$ does not divide $s-\tilde{s}$ nor $t-\tilde{t}$, it is possible to construct an algorithm that can also break the S-RSA assumption (see [7] for such an algorithm). □

We can now conclude from Theorem 1 and 2 that from l withdrawals with the bank, the customer can at most obtain l coins that a shop S will accept.

3.2 Anonymity

The following theorem proves that the bank cannot know who is involved during the payment protocol: the identity of the customer is kept secret even from the bank (except from the trusted authority for obvious reasons).

Theorem 3. *Under the DDH assumption and in the random oracle model, given a bank's view $W(C)$ of a withdrawal with a customer C and the view of a payment P, no PPT machine (apart from T) can decide whether the coin underlying the payment P comes from $W(C)$ or not with probability non-negligibly better than random guessing (in l_p).*

Proof. (sketch) Suppose we have a PPT machine $\mathcal{M}$ that can, on input $W(C)$ (a bank's view of withdrawal) and P (a payment transcript) decide, with probability non-negligibly better than random guessing, whether the coin used in P comes from $W(C)$. We will show that the bank can use this machine as an oracle to break the DDH assumption.
Let $n = pq$ be the product of two distinct safe primes of length l_p (where l_p is a security parameter). Let a be a random generator of $QR(n)$, m a random element of $QR(n)$, a^x a random element of $\langle a \rangle$ (the subgroup of $QR(n)$ generated by a) and $m^{x'}$ a random element of $\langle m \rangle$ (n, a, m, a^x and $m^{x'}$ will be the target instance of the DDH problem.
We will show that given $\mathcal{M}$, the bank can decide non-negligibly better than random guessing if $m^x = m^{x'} \pmod n$ [6]. We will first construct a (polynomial) converting algorithm $\mathcal{AL}$ which will transform the target instance of the DDH problem into a valid bank's view of a withdrawal and a correct payment transcript of our fair e-cash scheme.

Construction of $\mathcal{AL}$:

- Initialisation:
  The bank first chooses $r \in_R \Lambda$ and $e \in_R \Gamma$. B then computes $A = a^r \pmod n$ and $a_0 = A^e/a^x \pmod n$. Finally, B chooses two random generators g and h of $QR(n)$ and two random elements z and Z of $QR(n)$. The public key of the fair e-cash scheme becomes $PK = (n, a, a_0, g, h, m, z, Z)$ (where n, a and m are the values defined in the target instance of the DDH problem).

[6] In fact, this is a straightforwardly equivalent formulation of the DDH problem.

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

– Simulation of the withdrawal session:
1. B chooses at random $\tilde{\alpha} \in \mathbb{Z}^*_{2^{\lambda_2}}$ and $\tilde{\beta} \in]0, 2^{\lambda_2}[$ and defines $C_2 = a^x \pmod n$. Note that for a given triplet $(x, \tilde{\alpha}, \tilde{\beta})$ (with $x \in \Lambda$), there always exists a $\tilde{x}$ such that $x = 2^{\lambda_1} + (\tilde{\alpha}\tilde{x} + \tilde{\beta} \pmod{2^{\lambda_2}})$ (since $\tilde{\alpha}$ is an inversible element of $\mathbb{Z}_{2^{\lambda_2}}$).
2. B chooses $C_1 \in_R QR(n)$. Notice that $\forall \tilde{x} \in]0, 2^{\lambda_2}[, \exists \tilde{r} \in]0, n^2[$ such that $C_1 = g^{\tilde{x}} h^{\tilde{r}} \pmod n$.
3. B simulates the proof U which is possible in the random oracle model. For this purpose, B chooses $c \in_R I_k$, $s_1, s_2 \in_R I_{\epsilon(2l_p+k)+1}$, computes $t = C_1^c g^{s_1} h^{s_2} \pmod n$, defines $c = H(g \| h \| C_1 \| t)$ and returns $U = (c, s_1, s_2)$ as the signature of knowledge.
4. B chooses two random values in $QR(n)$ and defines these values as A_1 and A_2 (recall that B does not know $m^x \pmod n$). Notice that the machine $\mathcal{M}$ will not be able to distinguish this random pair (A_1, A_2) from a "correct" El Gamal encryption of $m^x \pmod n$. Otherwise this would imply that $\mathcal{M}$ can break the El Gamal encryption in the sense of indistinguishability, i.e. break the DDH assumption (see [2]). Notice then that the "correctness" of the ("wrong") ciphertext (A_1, A_2) (i.e., the proof V) can still be simulated in the random oracle model using standard techniques. The fact that the statement being "proved" is false is irrelevant since $\mathcal{M}$ will not be able to discern it.
5. B then simulates in the random oracle model, using standard techniques, the proof W.

$W(C) = (C_1, U, \tilde{\alpha}, \tilde{\beta}, C_2, V, W, A, e)$ is then a valid bank's view of a withdrawal protocol.

– Simulation of the payment:
1. B chooses two random values in $QR(n)$ and defines these values as T_1 and T_2 (recall that B does not know $a^{x'} \pmod n$). See the remarks in step 4 of the simulation of the withdrawal session.
2. B defines $T_4 = m^{x'} \pmod n$.
3. B chooses $e' \in_R \Gamma$ and $w_3 \in_R I_{2l_p}$ and computes $T_6 = g^{e'} h^{w_3} \pmod n$. Notice that $\forall (x', e') \in \Lambda \times \Gamma, \exists A' \in QR(n)$ such that $A'^{e'} = a_0 a^{x'} \pmod n$. Notice also that $\forall T_3 \in QR(n), \exists w_1 \in I_{2l_p}$ such that $T_3 = A' h^{w_1} \pmod n$ (in our case, B does not know (and cannot compute) the value A', since B does not know x').
4. B chooses at random $T_3 \in QR(n)$ and $T_5 \in QR(n)$. Notice that $\exists w_2 \in I_{2l_p}$ such that $T_5 = g^{w_1} h^{w_2} \pmod n$.
5. B simulates U in the random oracle model, using standard techniques.

$P = (T_1, \ldots, T_6, U)$ is then a valid payment transcript.
$W(C)$ and P are then feed to $\mathcal{M}$ which returns a bit b (where $b = 0$ if P is linked to $W(C)$ and 1 otherwise). If $b = 0$, B concludes that $m^x = m^{x'} \pmod n$ and that $m^x \neq m^{x'} \pmod n$ otherwise.
We thus have constructed a polynomial-time algorithm which can break the DDH

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.

assumption. As this is assumed to be infeasible, we can conclude that no one but T can match a transaction with a user. □

References

1. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. Crypto'2000, volume 1880 of LNCS, pages 255-270. Springer-Verlag, 2000.
2. D. Boneh. The Decision Diffie-Hellman Problem. 3rd Algorithmic Number Theory Symposium, volume 1423 of LNCS, pages 48-63. Springer-Verlag, 1998.
3. E. Brickell, P. Gemmel, D. Kravitz. Trustee-Based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change. 6th ACM-SIAM, pages 457-466. ACM Press, 1995.
4. J. Camenisch, A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. Eurocrypt 2001, volume 2045 of LNCS, pages 93-118. Springer-Verlag, 2001.
5. J. Camenisch, U.M. Maurer, M. Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trustees. Esorics'96, pages 33-43. Springer-Verlag, 1996.
6. D. Chaum, E. van Heyst. Group Signatures. Eurocrypt'91, volume 547 of LNCS, pages 257-265. Springer-Verlag, 1991.
7. I. Damgård, E. Fujisaki. A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order. Asiacrypt 2002, volume 2501 of LNCS, pages 143-159. Springer-Verlag, 2002.
8. G. Davida, Y. Frankel, Y. Tsiounis, M. Yung. Anonymity Control in E-Cash Systems. Financial Crypto'97, volume 1318 of LNCS, pages 1-16. Springer-Verlag, 1997.
9. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Trans. Inform. Theory, 31, pages 469-472. 1985.
10. A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. Crypto'86, volume 263 of LNCS, pages 186-194. Springer-Verlag, 1987.
11. Y. Frankel, Y. Tsiounis, M. Yung. Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash. Asiacrypt'96, volume 1163 of LNCS, pages 286-300. Springer-Verlag, 1996.
12. Y. Frankel, Y. Tsiounis, M. Young. Fair Off-Line e-cash Made Easy, Asiacrypt'98, volume 1514 of LNCS, pages 257-270. Springer-Verlag, 1998.
13. E. Fujisaki, T. Okamoto. Statistical Zero-Knowledge Protocols Solution to Identification and Signature Problems. Crypto'97, volume 1294 of LNCS, pages 16-30. Springer-Verlag, 1997.
14. M. Gaud, J. Traoré. On the Anonymity of Fair Off-Line e-Cash Systems, Financial Crypto'03 (to appear).
15. G. Maitland, C. Boyd. Fair Electronic Cash Based on a Group Signature Scheme. ICICS 2001, volume 2229 of LNCS, pages 461-465. Springer-Verlag, 2001.
16. W. Qiu, K. Chen, D. Gu. A New Off-line Privacy Protecting E-Cash System with Revokable Anonymity. ISC 2002. 2002.
17. M. Stadler, J.M. Piveteau, J. Camenisch. Fair Blind Signatures, Eurocrypt'95, volume 921 of LNCS. pages 209-219. Springer-Verlag, 1995.
18. J. Traoré. Group Signatures and Their Relevance to Privacy-Protecting Off-Line Electronic Cash Systems. ACISP'99, volume 1587 of LNCS, pages 228-243. Springer-Verlag, 1999.

Appeared in R. Safavi-Naini, J. Seberry (Eds.): ACISP 2003, LNCS 2727,pp. 237–248, 2003.