POLYPUBLIE

Secure untraceable off-line electronic cash system

Yaser Baseri, Benyamin Takhtaei, Javad Mohajeri

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2013
Type	Article de revue / Article
Version	Version officielle de l'éditeur / Published version
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/14281/
Évaluation par les pairs Peer reviewed	Révisé par les pairs / Refereed
Conditions d'utilisation Terms of use	Creative Commons Attribution 4.0 International (CC BY)
Référence Citation	Baseri, Y., Takhtaei, B., & Mohajeri, J. (2013). Secure untraceable off-line electronic cash system. Scientia Iranica D, 20(3), 637-646. https://doi.org/10.1016/j.scient.2013.05.002

Publication officielle
Official publication

Titre de la revue Journal title	Scientia Iranica D (vol. 20, no. 3)
Maison d'édition Publisher	Elsevier
URL officielle Official URL	https://doi.org/10.1016/j.scient.2013.05.002
Mention légale Legal notice	© 2013 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY license.

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

Scientia Iranica D (2013) 20 (3), 637–646

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.sciencedirect.com

Secure untraceable off-line electronic cash system

Y. Baseri [a,c,*], B. Takhtaei [b], J. Mohajeri [a]

[a] *Institute of Electronics Research, Sharif University of Technology, Tehran, Iran*

[b] *Data and Network Security Lab, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*

[c] *Department of Computer and Software Engineering, École Polytechnique de Montréal, Montréal, Québec, Canada*

Received 25 May 2011; revised 23 June 2012; accepted 21 November 2012

KEYWORDS

Electronic cash;
Payment systems;
Untraceable;
Date attachable.

Abstract Eslami and Talebi (2011) [25] proposed an untraceable electronic cash scheme and claimed that their scheme protects the anonymity of customers, detects the identity of double spenders and provides the date attachability of coins to manage the bank database. In this paper, illustrating Eslami and Talebi's scheme, as one of the latest untraceable electronic cash schemes, and showing its weaknesses (in fulfilling the properties of perceptibility of double spender, unforgeability and date attainability of coins) and its faults (related to exchange protocol), we propose a new untraceable electronic cash scheme which is immune to the weaknesses of the former. Our scheme contains anonymity, double-spending detection, unforgeability and date attachability properties and prevents forging. To do this, we described a special construction which injects the expiration date and the identity of the customer onto the coin and detects the identity in the case of double spending. Lastly, we show that the efficiency of our scheme is comparable with other schemes.

© 2013 Sharif University of Technology. Production and hosting by Elsevier B.V.
Open access under CC BY license.

1. Introduction

Nowadays, thanks to the progressing technology of computer networks and the Internet, information technology is used in many aspects of human life. One aspect is the use of information technology in electronic commerce. Since the appearance of electronic commerce, people have been able to carry out their commercial activities by the use of electronic money in their payment transactions. The first electronic money has been proposed by the David Chaum to have similar properties to paper cash [1]. There are a number of features considered for an electronic cash system. Some of them are listed:

- Anonymity: There should not be any relationship between the cash and its owner.
- Unforgeability: The digital cash should not be produced by anyone except authorized parties.
- Double spender perceptibility: The identity of malicious spenders, who spend the cash twice or more, should be revealed.
- Date attachability: Digital cash should contain the dates of withdrawing, paying and depositing. These dates are used to check the expiration date and charge for interest.
- Divisibility: Digital cash could be divided into smaller amounts.
- Transferability: Digital coins can be circulated among people.
- Anonymity revocation: While misusing or undertaking illegal activities, the coin or its owner could be traced.
- Portability: The security and the use of digital cash is not dependent on any physical location, but it could be transferred through computer networks into storage devices and vice versa.

In prevalent electronic cash systems, the bank, the customer and the merchant are three participants involved in the transaction cycle of the system. A customer opens an account in a bank, withdraws cash from his account and then pays it to a merchant. The merchant takes the cash, checks its validity, accepts it and deposits it with the bank.

* Corresponding author at: Institute of Electronics Research, Sharif University of Technology, Tehran, Iran. Tel.: +1 514 559 2996; fax: +1 514 340 4657.
E-mail addresses: yaser.baseri@polymtl.ca (Y. Baseri), b_takhtaei@ce.sharif.edu (B. Takhtaei), mohajer@sharif.edu (J. Mohajeri).

Peer review under responsibility of Sharif University of Technology.

Production and hosting by Elsevier

1026-3098 © 2013 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY license.

http://dx.doi.org/10.1016/j.scient.2013.05.002

Considering the relationship between the bank and the merchant, electronic cash systems could be divided into two categories: online and off-line. In online category, while paying the coin to the merchant, the bank should attend the transaction, validate the coin and check its double spending [2–4]. In contrast, in off-line payments, the validation of the coin is done partially by the merchant while paying. After connecting to the bank in the next phase, the validation will be completed. While efficiency is improved, double spending can only be detected after connecting with the bank [5–7]. Recently, some efforts have been made to integrate online and off-line electronic cash systems [8].

The main controversial issue in off-line electronic cash systems is simultaneously fulfilling untraceability and double spending detection. After the Chaum scheme which used a blind signature to achieve untraceability [1], numerous untraceable electronic cash schemes have been proposed based on this structure [9–14]. In blind signature-based schemes, the customer could get the signature of the bank on the coin without disclosing any information about the coin, and spend it without revealing his identity to the merchant. The other issue in off-line electronic cash is related to the nature of electronic cash. Since electronic cash is inherently digital, it could easily be copied and reused. So, a malicious customer could spend it twice or more. To address this problem, the identity of the customer should be revealed after double spending. One approach to doing this is the use of the *Cut and choose* technique. Although this technique is used in some schemes [14–16], due to computational and communicational overhead, it is highly inefficient. The other approach for detecting the identity of a double spender is the use of a restrictive blind signature, which is introduced by Bands [5]. In restrictive blind signature, the customer can blind the outside of the message, m, but not its internal structure. After double spending, the bank would be able to clear the structure in a polynomial time. Although Brands' scheme suffers from some weaknesses in misrepresenting the identity of the customer [17], some solutions have been proposed to prevent these weaknesses [10, 17]. Afterward, some schemes have been presented, which use a similar method to Bands' restrictive blind signature, to detect the identity of a double spender [18–20].

The other feature considered for an electronic cash system is adding the withdrawal date, transaction date (or effective date) and depositing date to it, in order to charge for interest and check the expiration date of the coin. Since the electronic cash systems are prepaid systems, the withdrawal and transaction dates are important to the customers and merchants if e-cash interest is considered. Using the transaction date, when the merchant deposits the e-cash, he can charge the interest of the e-cash during the transaction date and deposit date from the bank. To attach the date, several date attachment schemes have been proposed that let customers attach transaction dates to e-cash [21–24], and some other date attachment e-cash schemes let the merchant attach the date to the e-cash [5,25, 26]. In addition, to give the bank the ability of managing its own database, the expiration date should be attached to the coin. To detect the identity of a malicious spender while double spending, the banks should store the information related to the withdrawn coin in its own database. Regarding the expiration date, the bank could remove the information of outdated coins from its own database and control the size of it. Attaching an expiration date to coins, should be considered a procedure to exchange the outdated coins with new coins. This affair is done by considering an additional phase in the electronic cash protocol (i.e. exchange phase) [25,27,28]. In addition, to attach the expiration date, most schemes use partially blind signatures [23,29–32]. In partially blind signatures, the part of the information which contains pre-agreed information is clear to the signer and verifier (e.g. date and time), while the other parts of the massage and the signature are blinded to the signer.

Recently, Eslami and Talebi proposed an off-line electronic cash scheme and claimed that their scheme satisfies the requirements of the anonymity of customer, perceptibility of a double spender, date attachability and portability of cash [25]. Their scheme is based on cryptographic primitives, such as the ElGamal cryptosystem and blind signatures. However, as we show in the paper, their scheme suffers from some weaknesses in fulfilling the property of double spender perceptibility, date attachability, and some other weaknesses related to their exchange protocol.

In the rest of this paper, first, we review Eslami and Talebi's scheme and its weaknesses in Section 2. In Section 3, we propose a new scheme, which is immune to the weaknesses mentioned in the previous section. In the proposed scheme, we use a construction to provide the anonymity of a customer and to detect the identity of a double spender, which hides the identity of customer in the construction and reveals it after double spending. Furthermore, for attaching time to the coin, we use a method that injects the expiration date of the coin into the represented construction. Attempting to resist chosen plaintext attacks, which are based on homomorphic property, we insert an extra item, considering the addition of other items talented to the attacks. In Section 4, we show our scheme is immune to the weaknesses of Eslami and Talebi's scheme. Next, the security analysis of our scheme is presented in Section 5. We also compare its performance with some other related schemes in Section 6.

2. Eslami and Talebi's scheme and its failures

In this section, first, we review Eslami and Talebi's scheme. Then, we show its weaknesses to satisfy the claimed properties.

2.1. Eslami and Talebi's scheme

There are four participants in the scheme: a Central Authority (CA), the Bank (B), the Spender (S) and the Merchant (M). The protocol is done in five phases: initialization, withdrawal, payment, deposit and exchange.

2.1.1. Initialization

In this phase, the central authority fixes some parameters and certifies the public keys corresponding to the bank, spender and merchant. In this phase, the following steps are performed:

Step 1. The central authority, CA, selects a large prime, p, such that $q = (p-1)/2$ is also prime, α as the square of a primitive root (mod p) and three public hash functions, $\mathcal{H}$, $\mathcal{H}_0$ and $\mathcal{H}_1$. Then, it publishes p, α, $\mathcal{H}$, $\mathcal{H}_0$, $\mathcal{H}_1$.

Step 2. The bank selects its RSA parameters as (p_B, q_B, n_B, e_B, d_B), such that $n_B > p$, chooses a secret identity number, x, computes $z = \alpha^x \pmod p$ and publishes it.

Step 3. The spender selects its RSA parameters as (p_S, q_S, n_S, e_S, d_S), where $n_S > p$, m is an identity number and r_m is a random number. Then, it computes $I = (\mathcal{H}_1(m \parallel \alpha^{r_m}), m)^{e_B} \pmod{n_B}$ and sends it besides $\alpha^{r_m} \pmod p$ to the bank.

Step 4. The bank computes $I^{d_B} \pmod{n_B}$ to obtain m, and stores m and $\alpha^{r_m} \pmod p$, along with the identity information of the spender (e.g., name, address, etc.) in

its database. Then, it chooses a random number, k, and calculates the following numbers:

- $s = (m \parallel k) \pmod p$,
- $v = \alpha^s \pmod p$,
- $R = v^x \pmod p$.

It also stores s, k, v, R in its database and sends (v^{e_S}, R^{e_S}) to the spender.

Step 5. The merchant chooses an identification number ID_M and registers it with the bank.

2.1.2. Withdrawal

Giving a coin to the customer, the bank requires a proof of identity (i.e. the digital certificate issued by CA), just as when someone is withdrawing classical cash from an account. All coins in the scheme have the same value. A coin will be represented by a 6-tuple (u, g, A, r, A'', t) of numbers that are generated through the following steps:

Step 1. The spender decrypts v^{e_S}, R^{e_S} with his private key, d_S, to obtain the numbers v and R. Then, he chooses random numbers, e, l, β_1, β_2 and y, such that $gcd(y, p-1) = 1$, $gcd(l, n_B) = 1$, and $gcd(\beta_1, q) = 1$. He computes:

- $u = \alpha^y \pmod p$,
- $w = (R \parallel e)$,
- $g = \alpha^w \pmod p$,
- $A = v^{\beta_1}\alpha^{\beta_2} \pmod p$,
- $c = \beta_1^{-1}\mathcal{H}(u, g, A) \pmod q$,
- $a = Al^{e_B} \pmod{n_B}$,

and sends (a, c) to the bank.

Step 2. The bank selects $t = (Date \parallel Time)$ as the expiration date of the coin and computes:

- $c' = cx + s \pmod q$,
- $A' = (a\mathcal{H}_1(t))^{d_B} \pmod{n_B} = l(A\mathcal{H}_1(t))^{d_B} \pmod{n_B}$.

Then, it sends (A', c', t) to the spender.

Step 3. The spender computes:

- $r = \beta_1 c' + \beta_2 \pmod q$,
- $A'' = l^{-1}A' \pmod{n_B}$.

The coin (u, g, A, r, A'', t) is now complete.

2.1.3. Payment

This phase includes the following steps:

Step 1. Spender sends (u, g, A, r, A', t) to the merchant.

Step 2. The merchant checks the expiration date of the coin and verifies the equations $\alpha^r \stackrel{?}{=} Az^{\mathcal{H}(u,g,A)} \pmod p$ and $A\mathcal{H}_1(t) \stackrel{?}{=} A''^{e_B} \pmod{n_B}$ to ensure the validity of the coin. Then, he computes $d = \mathcal{H}_0(u, g, ID_M, Date \parallel Time)$, where $\mathcal{H}_0$ is the hash function in the initialization phase and *Date* and *Time* represent the date and time of the transaction. Finally, he sends d to the spender.

Step 3. The spender utilizes ElGamels scheme to compute γ, such that $wu + y\gamma = d \pmod{p-1}$, and sends γ to the merchant.

Step 4. The merchant accepts the coin if $g^u u^\gamma = \alpha^d \pmod p$.

2.1.4. Deposit

In this phase, the following steps are performed:

Step 1. The merchant sends $((u, g, A, r, A'', t), d, \gamma)$ to the bank.

Step 2. The bank checks whether the coin (u, g, A, r, A'', t) exists in either the deposit table or the exchange table, and skips to the Fraud Control procedure. Otherwise, it checks if $\alpha^r \stackrel{?}{=} Az^{\mathcal{H}(u,g,A)} \pmod p$ and $A\mathcal{H}_1(t) \stackrel{?}{=} A''^{e_B} \pmod{n_B}$.

If so, the coin is valid and the bank stores $((u, g, A, r, A'', t), d, \gamma)$ into the deposit table and transfers money to the merchants account.

In this phase, when the malicious spender spends the coin twice, the bank would be able to detect the identity of the malicious spender. Suppose that, first, merchant M deposits the coin $((u, g, A, r, A'', t)$ with the parameters $d, \gamma)$. When the second merchant, V, wants to deposit the same coin $((u, g, A, r, A'', t)$ with the extra parameters $d', \gamma')$ for the second time, the bank finds out that the coin has already been spent. The bank can use the property of the ElGamal signature to identify the spender who has done this. Since $y(\gamma - \gamma') = (d - d') \pmod{p-1}$, the bank would be able to compute y. Now, by the equation $wu + y = d \pmod{p-1}$, the bank can obtain w and identify the malicious spender.

2.1.5. Exchange

In this phase, the bank exchanges only outdated coins which are not in the deposit table or the exchange table. The owner of such coins can present the coin to the bank and receive a new coin with an updated expiration date. The details are as follows.

Step 1. The owner presents his/her outdated coin, together with I, to the bank, which checks (using a zero-knowledge technique) if the owner knows the corresponding r_m and if the coin is valid. Now, a new coin can be generated.

Step 2. The owner chooses random numbers and y', such that $gcd(y', p-1) = 1, gcd(l', n_B) = 1$, and $gcd(\beta', q) = 1$. Then, he computes u', w', g', A_1, c', a', as in Step 1.3 of the withdrawal protocol and sends a', c' to the bank.

Step 3. The bank computes c_1', A_1', as done in the withdrawal phase, and sends these numbers, along with t', to the owner.

Step 4. The owner computes r' and A'', as done in the withdrawal phase of the protocol.

The new coin is now complete. The bank then updates the exchange table. Note that when a coin enters this table, it is considered invalid and no further transaction on it can be performed.

2.2. Weaknesses of the scheme

In this subsection, we present some weaknesses of Eslami and Talebi's scheme and show that the scheme is vulnerable to some claimed properties. The first fault is in detecting the double spender's identity. The second one is in validating the expiration date of the coin, which results in violation of the unforgeability of the coin. The third fault is related to the exchange protocol.

2.2.1. First fault: attacking double spender detection

In this attack, misbehaving in the withdrawal protocol, the customer can forge his identity in such a way that the bank would not be able to identify him after double spending. As mentioned in the setup phase, for each customer, the bank stores the chosen parameters, m, α^{r_m}, s, k, v, R, besides his identity. After double spending, the bank could detect the value of w for the malicious customer. Finding the value of w, the bank

The malicious customer chooses a forged value $(R)'$ in $\mathbb{Z}_q$ instead of R and computes the forge value $(w)'$, $(g)'$, $(c)'$ instead of w, g, c (the other parameters remaind unchange):
$(w)' = ((R)'||e)$,
$(g)' = \alpha^{(w')} \pmod p$,
$(c)' = \beta_1^{-1}\mathcal{H}(u, (g)', A) \pmod q$,
The bank computes the forged value $(c')'$ (instead of and c') and sends A,$(c')'$ and t to the customer:
$(c')' = (c)'x + s \pmod q$,
The customer calculate the forged value $(r)'$ correspond to forged value of c' (i.e. $(c')'$):
$(r)' = \beta_1(c)' + \beta_2 \pmod q$,
The forged coin would be $(u, (g)', A, (r)', A'', t)$.

Figure 1: Attack 1, withdrawal phase.

could only detect the value of R from stored values. Referring to its database, the bank finds the identity of that customer. So, when the value of w is forged, the bank would not be able to detect the identity of the malicious spender. Since the correctness of the value of $w = (R \parallel e)$ has not been checkd anywhere, the customer would be able to change the value of R with arbitrary forged value R' in $\mathbb{Z}_q$, and, consequently, change the values of w, g and r with proper values w', g' and r' in the withdrawal phase (Figure 1).

Accordingly, in the payment phase, the values of d and r would be computed corresponding to the new forged values and the validation of the Elgamal signature employed in the payment phase will be satisfied (Figure 2).

By these changes, the bank could not verify the identity of the malicious spender. This coin passes through all validations in the withdrawal and payment phases. In addition, since the value of $(w)'$ is imitative, when a malicious customer spends the coin more than one time, the bank calculates $(w)'$ instead of his real parameter of w. So, it could not calculate his real identity.

2.2.2. Second fault: forging the expiration date

In this attack, withdrawing the coin, a malicious customer can manipulate the expiration date of the received coin. Since in the coin (u, g, A, r, A'', t), the expiration date is authorized only by A'', fixing the value of A and modifying the values of A'' and t in a proper way, the coin can be changed in such a way that it remains valid in the time, except time t. Suppose that the customer follows the withdrawal protocol two times with the same parameter, A. By that, the customer can withdraw two eligible coins $(u_1, g_1, A, r_1, A''_1, t_1)$ and $(u_2, g_2, A, r_2, A''_2, t_2)$ from his account in which $t_2 > t_1$. Replacing the values of A''_1, t_1 with corresponding values of the second coin (i.e. A''_2, t_2), the customer can forge the first coin in such a way that it is valid for time t_2 (Figure 3). This coin passes through the validation of payment phase (Figure 4).

2.2.3. Third fault: frauds on exchange protocol

Wallet scatting and thieving problem is one of the important problems of electronic wallets. Some solutions have been offered to solve this problem [20,33]. In Eslami and Talebi's protocol, without considering the exchange protocol, it is not possible to pay a coin without having information about the secret parameters, w and y. However, regarding the exchange protocol, a malicious customer can refer to the bank with his own real identity, exchange the thieved coin with a new coin and spend it. This problem is caused by the fact that in the first step of the exchange protocol, the bank only checks the identity of the possessor of the coin and validity of the coin, yet does not check the dependency of the coin on the possessor.

Also, in most cases, the loser does not have the information of the coin and, consequently, could not file a lawsuit in court. Even by knowing the information of the coin, he should prove his ownership of the coin, which is not mentioned in Eslami and Talebi's protocol.

The other problem of the exchange phase in Eslami and Talebi's protocol is related to the size of database. They claimed that by the proposed exchange protocol, they would be able to manage the size of database. However, since their scheme stores the information of all exchanged outdated coins in an exchange table, it could not avoid database size increase. As mentioned in [29,34], the main aim of inserting validation time into the coin is discarding the information of outdated coins and controlling the size of the database, while they did not consider this note.

3. The proposed scheme

There are four participants in the scheme: a Central Authority (CA), the Bank (B), the Spender (S) and the Merchant (M). The scheme contains five phases: initialization, withdrawal, payment, deposit and exchange.

Note that in the scheme, we use the RSA cryptosystem, which is based on the difficulty of the computation of e'th root of numbers in $\mathbb{Z}_n^*$, such that $n = p * q$ and p, q are two large prime numbers. The public key of the system is e, a reasonably large prime, and the corresponding private key is $1/e \pmod{\phi(n)}$. Ciphertexts can be computed as the e'th exponent of plaintexts (encryption) and the e'th root of ciphertexts can be computed as the planetexts (decryption). Everyone who knows the factorization of n is able to compute the e'th root of numbers in $\mathbb{Z}_n^*$ and, consequently, is able to decrypt ciphertexts. This type of RSA deployment has been used in some other protocols such as Ferguson's protocol [35].

The other note which should be mentioned here is that for attaching time to the coin, we use a method that injects the expiration date of the coin into the power of some parameters. Although similar RSA-based methods have been presented to attach the time to the structure of the signature, such as the schemes proposed by Abe and Fujisaki [29] and by Cao et al.'s [30], it is shown that their schemes are vulnerable to some homomorphic-based attacks ([36–38], respectively). To be immune to to these attacks, we insert extra items into the coin, considering the addition of the other two essential parameters. By that, we protect our protocol against the mentioned attacks and enhance the security of the scheme.

3.1. Initialization

In this phase, which could be enumerated as the set up phase, the central authority should set some public parameters.

The malicious customer spends the forged coin $(u, (g)', A, (r)', A'', t)$ to the merchant. This coin passes through all of the following validations checked by the merchant:
$A\mathcal{H}_1(t) \stackrel{?}{=} A''^{e_B} \pmod{n_B}$,
$\alpha^{(r)'} \stackrel{?}{=} Az^{\mathcal{H}(u,(g)',A)} \pmod{p}$,
In response to $(d)' = \mathcal{H}_0(u, (g)', ID_M, Date||Time)$ sent by the merchant, the spender computes $(\gamma)'$ in a way that $(w)'u + y(\gamma)' = (d)' \pmod{p-1}$. The merchant accepts the forged coin because the following equation is held:
$(g)'^u u^{(\gamma)'} = \alpha^{(d)'} \pmod{p}$.

Figure 2: Attack 1, payment phase.

The malicious customer runs withdrawal phase of the protocol in two different times t_1 and t_2 (such that $t_2 >> t_1$) and gets two eligible coins with the same parameter A from his account legally.
The customer spends the second coin $(u_2, g_2, A, r_2, A''_2, t_2)$ legally.
He replaces the values A''_1, t_1 of the first coin with the corresponding values of the second coin (A''_2, t_2).
The resulted coin would be $(u_1, g_1, A, r_1, A''_1, t_1)$.

Figure 3: Attack 2, withdrawal phase.

The malicious customer sends the forged coin $(u_1, g_1, A, r_1, A''_2, t_2)$ instead of $(u_1, g_1, A, r_1, A''_1, t_1)$ to a merchant. This coin passes through all of the following validations checked by the merchant:
$A\mathcal{H}_1(t_2) \stackrel{?}{=} A''^{e_B}_2 \pmod{n_B}$,
$\alpha^{r_1} \stackrel{?}{=} Az^{\mathcal{H}(u_1,g_1,A)} \pmod{p}$,
In response to $d_1 = \mathcal{H}_0(u_1, g_1, ID_M, Date||Time)$ sent by the merchant, the spender computes γ_1 in a way that $w_1u_1 + y_1\gamma_1 = d_1 \pmod{p-1}$. The merchant accepts the forged coin because the following equation holds:
$g_1^{u_1} u_1^{\gamma_1} = \alpha^{d_1} \pmod{p}$.

Figure 4: Attack 2, payment phase.

These parameters include two publicly known elements, g_1, g_2, of the same large prime order, l in $\mathbb{Z}_n^*$, and a one-way hash function, $\mathcal{H}$. In addition, each authenticated participant involved in the system should determine his own parameters and get a certificate for its own public key from certification authority. The required parameters of the bank are two RSA public/private key pairs (i.e. $((e_B, n), 1/e_B)$ and $((e'_B, n), 1/e'_B)$ such that $e_B > e'_B$). Using two RSA key pairs for the bank, we enhance the security of the system and prevent some security attacks based on the homomorphic property.

3.2. Opening an account

To open an account, the customer should identify himself to the bank. Authenticating the customer, the bank stores his identity information in its account database. By that, while double spending occurs, the bank would be able to determine the u parameter of the offender, compute g_1^u, refer to its database and reveal the identity of the offender. This process is done in the following steps:

Step 1. The customer:
- (a) Identifies himself by means of official documents, like a passport or some other identification.
- (b) Generates a random number, $u \in_R \mathbb{Z}_n^*$, and keeps it as his own secret identity information which is unknown to any other, unless he spends a coin more that one time.
- (c) Computes:
  - $ID_C = g_1^u \pmod{n}$ as his identity, such that $g_1^u g_2 \neq 1 \pmod{n}$.
- (d) Sends ID_C to the bank.
- (e) Provides a zero knowledge proof that he knows the discrete logarithm of ID_C, with respect to g_1.

Step 2. The bank B:
- (a) Checks the identity and the zero knowledge proof offered by the customer.
- (b) Stores the identity information of the customer in the account database.
- (c) Computes A and O_1 as its own signature on A:
  - $A = ID_C g_2 \pmod{n}$,
  - $O_1 = A^{1/e_B} \pmod{n}$.
- (d) Sends A and O_1 to the customer.

3.3. Withdrawal

Before withdrawing and asking for a coin, the spender should prove his/her ownership of the account to the bank. The spender should prove his identity in a similar way to the withdrawal of classical cash from an account (i.e. by offering his passport or driving license). In addition, he should refer to a bulletin board in which the bank periodically publishes the fresh time by two parameters, t and $e_B * t \pmod{\phi(n)}$. Time t is constant during the period and used to synchronize customers and the bank in the withdrawing process and to determine the validation time of coins. Note that $t * e_B$ plays the role of a public key for the bank and is chosen in such a way that its reverse (i.e. $1/(e_B * t) \pmod{\varphi(n)}$) exists. (Abe and Fujisaki [29] introduced a method to choose constant t such that $((1/(e_B * t)) \pmod{\phi(n)})$ exists.) The coin is represented by a five-tuple (A', B, s_1, s_2, s_3) constructed in the following steps (Figure 5):

Step 1. The spender S:
- (a) Chooses three random numbers, $x_1, x_2 \in_R \mathbb{Z}_{e'_B}^*$ and $s \in_R \mathbb{Z}_n^*$, and two blinding factors, $b_1, b_2 \in \mathbb{Z}_n^*$.

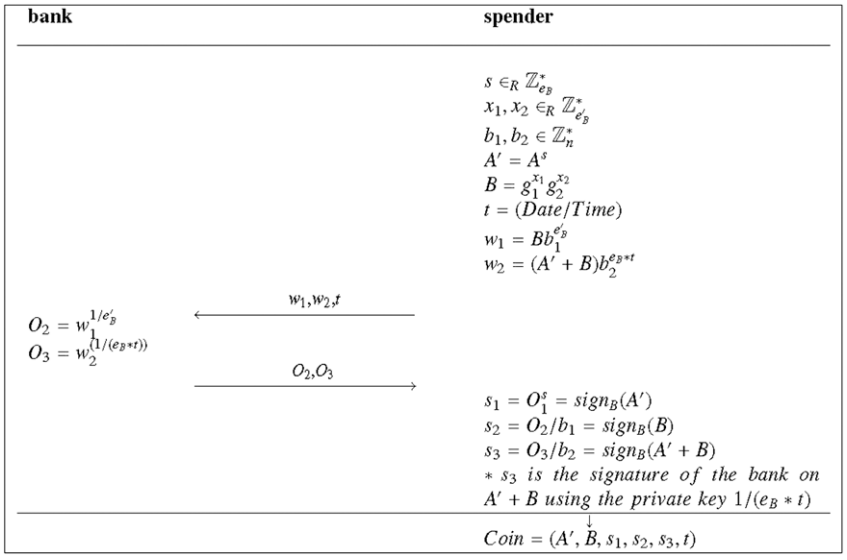


Figure 5: Withdraw protocol.

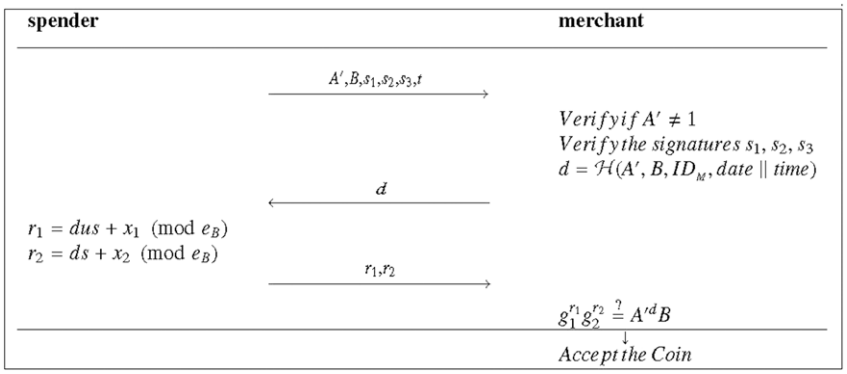


Figure 6: Payment protocol.

(b) Computes:
- $A' = A^s \pmod{n}$,
- $B = g_1^{x_1} g_2^{x_2} \pmod{n}$,
- $w_1 = Bb_1^{e'_B} \pmod{n}$,
- $w_2 = (A' + B)b_2^{(e_B * t)} \pmod{n}$.

(c) Sends w_1, w_2, t to the bank.

Step 2. The bank B:

(a) Checks the validity of the Date/Time slip.
(b) Signs w_1, and w_2 by computing:
- $O_2 = w_1^{1/e'_B} \pmod{n}$,
- $O_3 = w_2^{1/(e_B * t)} \pmod{n}$.

(c) Sends O_2 and O_3 to the spender.

Step 3. The spender S:

(a) Verifies the signatures of the bank on A, w_1, w_2.
(b) Obtains the signatures of the bank on A', B and $A' + B$, which are signed with private keys $1/e_B$, $1/e'_B$ and $(1/(e_B * t))$, respectively:
- $s_1 = O_1^s \pmod{n} = \text{sign}_B(A')$,
- $s_2 = O_2/b_1 \pmod{n} = \text{sign}_B(B)$,
- $s_3 = O_3/b_2 \pmod{n} = \text{sign}_B(A' + B)$.

The *Coin* is $(A', B, s_1, s_2, s_3, t)$.

3.4. Payment

When the customer wants to spend his coin at the shop, the following steps are done (Figure 6):

Step 1. The spender S:

(a) Sends A', B, s_1, s_2, s_3, t to the merchant M.

Step 2. The merchant M:

(a) Verifies if $A' \neq 0$.
(b) Checks the expiration date of the coin.
(c) Verifies the signatures, s_1, using the public key, e_B, s_2 using the public key, e'_B and s_3, and using the public key $(e_B * t)$.
(d) Computes:
- The challenge $d = \mathcal{H}(A', B, ID_M, date \parallel time)$ in which $\mathcal{H}$ is the hash function determined in the initialization phase, ID_M is the identity of the merchant and *date* $\parallel$ *time* represents the date and time of the transaction.

(e) Sends d to the spender.

Step 3. The spender S:

(a) Computes:
- $r_1 = dus + x_1 \pmod{e_B}$,
- $r_2 = ds + x_2 \pmod{e_B}$.

(b) Sends r_1 and r_2 to the merchant.

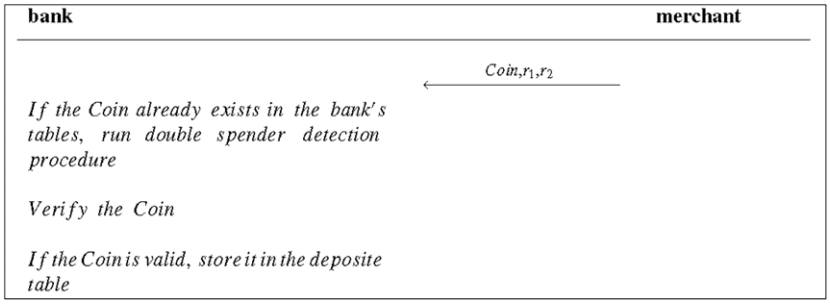


Figure 7: Deposit protocol.

Step 4. The merchant M:
(a) Accepts the coin if $g_1^{r_1} g_2^{r_2} = A'^{dB}$.

3.5. Deposit

In this phase, the following process is done between the bank B and the merchant M (Figure 7):

Step 1. The merchant M:
(a) Sends the transcript of each electronic coin (i.e. $Coin, r_1, r_2$) to the bank.

Step 2. The bank B:
(a) Checks the authenticity of the merchant and verifies the transcript of the received coin.
(b) Checks whether the coin exists in its deposit or exchange tables or not.
(i) If the coin exists, it runs the double spender detection procedure,
(ii) Else, accepts the coin, stores it in the deposit table and transfers money to the merchant.

3.5.1. Double spender detection procedure

Suppose that a malicious spender spends the same coin twice or more. Suppose that the malicious spender first spends the coin, along with d, r_1 and r_2. When the customer spends that coin for the second time (with the same parameters A' and B) along with d', r_1' and r_2', the bank finds out that the coin already exists in its tables. At that time, using the relation between r_1, r_2, d and consequently between r_1', r_2', d', it computes the identity of the malicious spender by the following equations:

- $u = \frac{r_1 - r_1'}{r_2 - r_2'} \pmod{e_B}$,
- $ID_c = g_1^u \pmod{n_B}$.

3.6. Exchange

In this phase, referring to the bank, the customer can exchange his old coin (which is not outdated) with new coins and update the expiration date of his own coin. To control the size of its database, the bank should remove the information of outdated coins from its database. This affair is undertaken by the following procedure (Figure 8):

Step 1. The Customer:
(a) Offers his coin, besides his identity, to the bank.

Step 2. The Bank B:
(a) Checks deposit and exchange tables to ensure that the coin has not already been exchanged or spent.
(b) Checks the authenticity of the customer and verification of the coin similar to the validation checking of the payment phase.
(c) Runs the withdrawal phase of the protocol.
(d) Updates the exchange table by inserting the information of the customer and the old coin.

4. Immunity to the proposed attacks

As shown in Section 2.2, we enumarate some weaknesses and faults of Eslami and Talebi's scheme, which are related to detecting the identity of a double spender, validating the expiration date and the fault related to the exchange protocol. In this section, we show how we immune our proposed scheme to these weaknesses and faults.

The first fault is related to detecting the double spender's identity. In our scheme, to spend the coin, $Coin = (A', B, s_1, s_2, s_3, t)$, more than one time, a malicious spender has two options: Using his own identity, or using a forged identity.

1. If he uses his own identity (the A or A^k), he should compute r_1 and r_2 parameters in the payment phase in such a way that equation $g_1^{r_1} g_2^{r_2} = A'^d B$ holds. Since:

$$\begin{aligned} A'^d B &= ((A^k)^s)^d B = A^{dks} B \\ &= (g_1^u g_2)^{dks} (g_1^{x_1} g_2^{x_2}) \\ &= g_1^{duks+x_1} g_2^{dks+x_2} \pmod{n}, \end{aligned}$$

then:
- $r_1 = duks + x_1 \pmod{e_B}$,
- $r_2 = dks + x_2 \pmod{e_B}$.

So, in the case of double spending, the values of r_1' and r_2' (the related values of r_1 and r_2 for the second time) should be:
- $r_1' = d'uks + x_1 \pmod{e_B}$,
- $r_2' = d'ks + x_2 \pmod{e_B}$.

Now, the bank would be able to detect the identity of the double spender in the way described in Section 3.5.
2. If the malicious spender, misbehaving in the withdrawal protocol, chooses a forged identity in the form of $A_{forged} \neq A^k$ for himself, he should provide the signature of the bank on A_{forged}^s, as the first parameter of the coin (i.e. s_1). So, he could provide the RSA signature of the bank on new values (i.e. A_{forged}^s) and this will contradict the security of the RSA cryptosystem.

The second fault is on validating the expiration date of the coin. Since the structure of our coin(its parameters) is different from the former, this attack is not applicable to our scheme.

The third fault is related to the exchange protocol. One part of the problem is caused by the fact that in the first step of the exchange protocol, the bank only checks the identity of the possessor of the coin and validity of the coin, yet does not check

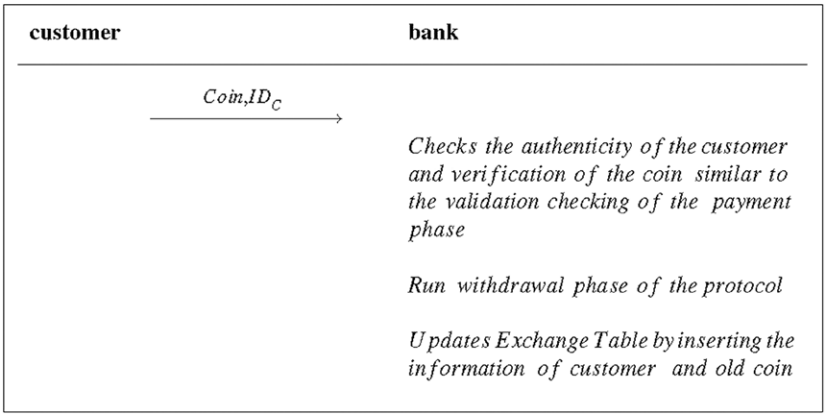


Figure 8: Exchange protocol.

the dependency of the coin on the possessor. In the proposed scheme, to forbid the exchange of the stolen coin with the new coin, the bank performs the required validations (related to coin validation and customer's authentication) in the exchange phase of the protocol. The other part of the problem is related to the size of the database. Removing the information of the expired coins, the bank would be able to manage the size of its own database.

5. Security discussion

In this section, first, we represent some assumptions used to analyze the properties of the protocol. Then, we discuss the satisfaction of anonymity, double spender detection and unforgeability properties using our protocol. Achieving these properties is related to the hardness of the RSA cryptosystem, the Discrete Logarithm problem and the Representation problem. Finally, we conduct a security comparison between our scheme, Eslami and Talebi scheme and some other electronic cash scheme.

5.1. Assumptions

Assumption 1 (*RSA Assumption*)**.** Given RSA module N as multiplication of the two prime elements, p and q, random element, $y \in \mathbb{Z}_N^*$ and the element, $e > 1$, such that $gcd(e, \phi(N)) = 1$. It is hard to compute the message x such that $x^e = y \pmod N$ using input 3-tuple (y, N, e) [39].

Assumption 2 (*DL Assumption*)**.** Given a large group, $\mathbb{G}$, of order p, a generator g and a random element, $h \in \mathbb{G}$. It is hard to find an integer, $a \in \mathbb{Z}_p^*$ such that $h = g^a$.

Assumption 3 (*Representation Assumption*)**.** Given a large group, $\mathbb{G}$, of order p and a k-tuple $(g_1, \ldots, g_k)$ and random element, h. It is hard to compute a tuple $(a_1, \ldots, a_k)$, as representation of h, with respect to the k-tuple $(g_1, \ldots, g_k)$, such that $\prod_{i=1}^{k} g_i^{a_i} = h$ [5].

5.2. Anonymity

An e-cash scheme is anonymous, if no one can reveal the identity of the payer. Hence, to prove this property, it is required to show that the information of payment does not reveal any knowledge about the identity of the customer who got the coin in withdrawal phase.

The information achieved in the spending phase of the protocol includes the parameters of *coin* (i.e. $(A', B, s_1, s_2, s_3, t)$) and extra information offered by the spender to the merchant (i.e. r_1 and r_2). Since in the withdrawal phase, while getting the signatures of the bank on coin parameters, B and $A' + B$ (i.e. s_2 and s_3), the parameters were blinded by blinding factors b_1 and b_2, they give no information to the adversary. Although $A' = A^s$ and $s_1 = O_1^s$ parameters lose their relations with A and O_1 by random parameter, s. In addition, parameter $B = g_1^{x_1} g_2^{x_2}$ seems random in the view of the adversary and finding x_1 and x_2 are hard (due to the hardness of *RP* problem). Finally, in the payment phase, while signing the response to the challenging parameter d, by r_1 and r_2, we have an equation system with two equations and four unknown parameters. So, it gives no information to the adversary.

5.3. Double spender detection

As we have shown in Section 3.5.1, in the case of double spending, the bank would be able to detect the u parameter of a malicious spender by equation $u = \frac{r_1 - r_1'}{r_2 - r_2'} \pmod{e_B}$ and consequently find his identity using $ID_C = g_1^u \pmod{n_B}$. To disrupt the computations, the adversary should either employ a forged identity or misbehave in providing the values of r_1 and r_2. Considering different phases of the scheme, the adversary could undertake these affairs in one of the following scenarios:

1. In the account opening phase, the customer chooses his identity in a way different from $ID_C = g_1^{u_1}$. As we discussed in Section 3.2, he should provide a zero knowledge proof to the bank that he knows the discrete logarithm of ID_C in the base of g_1. To provide the zero knowledge proof, the malicious customer should be able to solve the discrete logarithm problem.
2. In the withdrawal phase, the customer uses an identity different from his real identity. Since the bank has signed the parameter $A = ID_C g_2$ for his real identity ID_C (i.e. $O_1 = \text{sign}_B(A)$), the customer only accesses the signature of the bank on his real identity, unless he is able to forge the RSA signature of the bank. Moreover, due to considering the signature of the bank on A, B and additive parameter $A' + B$ using different private keys, it is impossible to forge extra O_1.
3. In the payment phase, the customer uses an identity different from his real identity in computing r_1 and r_2. Since in equation $g_1^{r_1} g_2^{r_2} = A'^d B$, the value of A' includes the real identity of the spender, employing different identity u in computing r_1, the equation does not hold.

So, in none of the steps of the protocol, could the adversary use the forged identity for himself. Consequently, if he spent more than once, his identity would be revealed.

Table 1: Security properties comparison.

	Juang	Martínez-Peláez et al.	Eslami and Talebi	Our scheme
Anonymity	✓	✓	✓	✓
Unforgeability	✓	✓	✓	✓
Double spender detection	✓	✓	×	✓
Date attachability	×	✓	×	✓

Table 2: C_1: computation cost of the withdrawing and spending for spender, C_2: computation cost of the withdrawing for the bank, C_3: computation cost of the verifying e-coin for merchant, C_4: communication cost of withdrawing an e-coin (bits), C_5: transaction mode.

	Juang	Martínez-Peláez et al.	Eslami and Talebi	Our scheme
C_1	3E+6M	2E+5H	5E+9M+1H	6E+8M
C_2	1E+2M	1E	1E+2M+1H	2E+1M
C_3	2E+2M	2H	6E+3M+2H	6E+2M+1H
C_4	2528	2152	2368	4256
C_5	Off-line	On-line	Off-line	Off-line

5.4. Unforgeability

An e-cash scheme is unforgeable, if no one could create valid coins by a means other than withdrawing them from the bank. Suppose that the adversary $\mathcal{A}$ wants to forge a coin $Coin = (A', B, s_1, s_2, s_3, t)$. To do that, it should change one of the parameters, A' and B, and properly change the values of $s_1 = \text{sign}_B(A')$, $s_2 = \text{sign}_B(B)$, and $s_3 = \text{sign}_B(A' + B)$. The adversary can do one of the following three scenarios:

1. To forge A' and related signature (i.e. s_1) $\mathcal{A}$ can either power both of them with the same value or use a homomorphic property and construct their new value using the corresponding value of two or more coins. In both cases, $\mathcal{A}$ would not be able to generate the valid RSA signature of the bank on $A' + B$.
2. To forge B and related signature (i.e. s_2), in a similar way, it could be shown that due to the dependency of the value of $A' + B$ on B, forging the value of B without the ability to generate the RSA signature is impossible.
3. The only remaining way is by choosing a new value for both A' and B, such that their addition (i.e. $A'+B$) is left unchanged. Due to the hardness of discrete logarithm and representation problems, it is hard to find the requisite value, u, x_1, x_2 (which is required for paying) corresponding to new values of A' and B.

5.5. Security properties comparison

In Table 1, we conduct a security comparison between our proposed scheme, Eslami and Talebi's scheme [25], Juang's scheme [40], and Martínez-Peláez et al.'s scheme [31] in which we consider *anonymity, unforgeability, double spender detection* and *date attachability*.

6. Performance comparison

In Table 2, we compare the computational and communicational complexity of our scheme, Eslami and Talebi's scheme and some other related schemes. For security consideration, the schemes suppose some computational and communicational assumptions: As [41,42], we assume that n is 1024 bites and e_B and e'_B are 160 bites. The assumptions about the size of parameters used in the compared schemes are similar to Eslami and Talebi's assumptions. We also assume that H is the computation time of one hashing operation, M is the computation time of one modular multiplication in a 1024-bit operation and E is the computation time of one modular exponential operation in a 1024-bit operation.

We conduct a comparison between our scheme, Eslami and Talebi's scheme [25], Juang's scheme [40] and Martínez-Peláez et al.'s scheme [31] in which we consider (1) the computation cost of the withdrawing and spending for the Spender, (2) the computation cost of the withdrawing for the Bank, (3) the computation cost of the verifying e-coin for the Merchant, (4) the communication cost of withdrawing an e-coin (bits), (5) the transaction mode of communication. It should be noted that on-line schema have lower costs of computation, but they have more limitations than off-line schema.

7. Conclusion

In this paper, we considered one of the latest untraceable electronic cash protocols and showed its weaknesses. Furthermore, we contributed an electronic cash scheme which is immune to the weaknesses of Eslami and Talebi's scheme. Our scheme satisfies anonymity, double-spending detection, unforgeability, date-attachability properties and prevents forging coins. To do this, we contributed a special structure which injects the expiration date and the identity of the customer into the coin and detects its identity in the case of double spending. We also showed that the efficiency of our scheme is comparable with other schemes. The security of the new protocol was also considered.

References

[1] Chaum, D. "Blind signatures for untraceable payments", *Advances in Cryptology: Proceedings of Crypto.*, 82, pp. 199–203 (1983).
[2] Sai Anand, R. and Madhavan, C. "An online, transferable e-cash payment system", *Progress in CryptologyINDOCRYPT 2000*, pp. 77–91 (2000).
[3] Martínez-Peláez, R. and Rico-Novella, F. "New electronic cash model: a script anonym", *Proceedings of the IADIS International Conference on E-Commerce, e-commerce'06*, pp. 392–396 (2006).
[4] Shi, L., Carbunar, B. and Sion, R. "Conditional ecash", *Financial Cryptography and Data Security*, pp. 15–28 (2007).
[5] Brands, S. "Untraceable off-line cash in wallet with observers", In *Advances in Cryptology-CRYPTO'93*, pp. 302–318, Springer (1994).
[6] Tan, Z. "An off-line electronic cash scheme based on proxy blind signature", *The Computer Journal*, 54(4), pp. 505–512 (2011).
[7] Everaere, P., Simplot-Ryl, I. and Traoré, I. "Double spending protection for e-cash based on risk management", *Information Security*, pp. 394–408 (2011).
[8] Huang, C. "Provably secure integrated on/off-line electronic cash for flexible and efficient payment", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(5), pp. 567–579 (2010).

[9] Anderson, R., Manifavas, C. and Sutherland, C. "NetCard-A practical electronic-cash system", In *Security Protocols*, pp. 49–57, Springer (1997).
[10] Davida, G., Frankel, Y., Tsiounis, Y. and Yung, M. "Anonymity control in e-cash systems", In *Financial Cryptography*, pp. 1–16, Springer (1997).
[11] Maitland, G. and Boyd, C. "Fair electronic cash based on a group signature scheme", *Information and Communications Security*, pp. 461–465 (2001).
[12] Chaum, D. and Brands, S. "Minting'electronic cash", *Spectrum, IEEE*, 34(2), pp. 30–34 (2002).
[13] Camenisch, J., Hohenberger, S. and Lysyanskaya, A. "Compact e-cash", *Advances in Cryptology-EUROCRYPT 2005*, pp. 302–321 (2005).
[14] Wang, H. and Zhang, Y. "Untraceable off-line electronic cash flow in e-commerce", *24th Australasian Computer Science Conference Proceedings. IEEE*, pp. 191–198 (2002).
[15] Chaum, D., Fiat, A. and Naor, M. "Untraceable electronic cash", In *Advances in Cryptology CRYPTO88*, pp. 319–327, Springer (1990).
[16] Okamoto, T. and Ohta, K. "Universal electronic cash", In *Advances in Cryptology-CRYPTO91*, pp. 324–337, Springer (1992).
[17] Chan, A., Frankel, Y., MacKenzie, P. and Tsiounis, Y. "Mis-representation of identities in e-cash schemes and how to prevent it", In *Advances in Cryptology-ASIACRYPT'96*, pp. 276–285, Springer (1996).
[18] Wang, C. "Untraceable fair network payment protocols with off-line TTP", *Advances in Cryptology-ASIACRYPT 2003*, pp. 173–187 (2003).
[19] Hou, X. and Tan, C. "On fair traceable electronic cash", *Communication Networks and Services Research Conference, 2005, Proceedings of the 3rd Annual, IEEE*, pp. 39–44 (2005).
[20] Liu, J., Tsang, P. and Wong, D. "Recoverable and untraceable e-cash", In *Public Key Infrastructure*, D. Chadwick and G. Zhao, Eds., In *Lecture Notes in Computer Science, 3545*, pp. 206–214, Springer, Berlin/Heidelberg (2005).
[21] Fan, C., Chen, W. and Yeh, Y. "Date attachable electronic cash", *Computer Communications*, 23(4), pp. 425–428 (2000).
[22] Chang, C. and Lai, Y. "A flexible date-attachment scheme on e-cash", *Computers & Security*, 22(2), pp. 160–166 (2003).
[23] Juang, W. "D-cash: a flexible pre-paid e-cash scheme for date-attachment", *Electronic Commerce Research and Applications*, 6(1), pp. 74–80 (2007).
[24] Fan, C. and Sun, W. "Efficient encoding scheme for date attachable electronic cash", *The 24th Workshop on Combinatorial Mathematics and Computation Theory*, pp. 405–410 (2007).
[25] Eslami, Z. and Talebi, M. "A new untraceable off-line electronic cash system", *Electronic Commerce Research and Applications*, 10(1), pp. 59–66 (2011).
[26] Elaalim, K. and Yang, S. "Electronic cash system with double spending tracing based on elliptic curve cryptography", *Journal of Computational Information Systems*, 6(9), pp. 2949–2957 (2010).
[27] Ku, C., Tsao, C., Lin, Y. and Chen, C. "An escrow electronic cash system with limited traceability", *Information Sciences*, 164(1–4), pp. 17–30 (2004).
[28] Yacobi, Y. "Efficient electronic money", *Advances in Cryptology-ASIACRYPT'94*, pp. 153–163 (1996).
[29] Abe, M. and Fujisaki, E. "How to date blind signatures", In *Advances in Cryptology ASIACRYPT'96*, pp. 244–251, Springer (1996).
[30] Cao, T., Lin, D. and Xue, R. "A randomized RSAbased partially blind signature scheme for electronic cash", *Computers & Security*, 24(1), pp. 44–49 (2005).
[31] Martínez-Peláez, R., Rico-Novella, F. and Satizábal, C. "TOMIN: trustworthy mobile cash with expiration-date attached", *Journal of Software*, 5(6), pp. 579–584 (2010).
[32] Song, R. and Korba, L. "How to make e-cash with non-repudiation and anonymity", *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2, pp. 167–172 (2004).
[33] Pfitzmann, B. and Sadeghi, A. "Self-escrowed cash against user blackmailing", In *Financial Cryptography*, pp. 42–52, Springer (2001).
[34] Wang, C. and Xuan, H. "A fair off-line electronic cash scheme based on RSA partially blind signature", *In 1st International Symposium on Pervasive Computing and Applications. IEEE*, pp. 508–512 (2007).
[35] Ferguson, N. "Single term off-line coins", In *Advances in Cryptology-EUROCRYPT'93*, pp. 318–328, Springer (1994).
[36] Coron, J., Naccache, D. and Stern, J. "On the security of RSA padding", In *Advances in Cryptology-CRYPTO99*, p. 789. Springer (1999).
[37] Desmedt, Y. and Odlyzko, A.M. "A chosen text attack on the rsa cryptosystem and some discrete logarithm schemes", In *Lecture Notes in Computer Sciences; 218 on Advances in cryptology-CRYPTO 85*, pp. 516–522, Springer-Verlag New York, Inc., New York, NY, USA (1986).
[38] Martinet, G., Poupard, G. and Sola, P. "Cryptanalysis of a partially blind signature scheme or how to make $100 bills with $1 and $2 ones", In *Financial Cryptography and Data Security: 10th International Conference*, pp. 171–176, Springer-Verlag New York Inc. (2006).
[39] Hohenberger, S. and Waters, B. "Short and stateless signatures from the RSA assumption", In *Advances in Cryptology-CRYPTO 2009*, S. Halevi, Ed., In *Lecture Notes in Computer Science, 5677*, pp. 654–670, Springer, Berlin/Heidelberg (2009).
[40] Juang, W. "A practical anonymous off-line multiauthority payment scheme", *Electronic Commerce Research and Applications*, 4(3), pp. 240–249 (2005).
[41] Lenstra, A., Tromer, E., Shamir, A., Kortsmit, W., Dodson, B., Hughes, J. and Leyland, P. "Factoring estimates for a 1024-bit RSA modulus", *Advances in Cryptology-ASIACRYPT 2003*, pp. 55–74 (2003).
[42] National Institute of Standards and Technology, FIPS PUB 186-2: Digital Signature Standard (DSS), NIST Publication (2000).

Yaser Baseri received his BS degree in Computer Science from Shahid Beheshti University, Tehran, Iran, in 2005 and his MS degree in Computer Science from Sharif University of Technology, Tehran, Iran, in 2007. Currently, he is a Ph.D. student in the Computer and Software Engineering Department of École Polytechnique de Montréal, in Québec, Canada. His research interests include formal methods, cryptography, and network security, cryptography, and formal method.

Benyamin Takhtaei received his BS and MS degrees in Software Engineering and Network Security from Sharif University of Technology, Tehran, Iran, in 2009 and 2012, respectively. His major research interests include information security, cryptographic protocols, and formal methods.

Javad Mohajeri received a BS degree from Isfahan University, Iran, in 1986 and an MS degree from Sharif University of Technology, Tehran, Iran, in 1989, both in Mathematics. Currently, he is Assistant Professor in the Electronics Research Institute at Sharif University of Technology, Iran. His research interests include design and cryptanalysis of cryptographic algorithms, and protocols and data security. He is author/co-author of over 60 research articles in refereed Journals/Conferences, and is a founding member of the Iranian Society of Cryptology.