Cryptanalysis on “Secure untraceable off-line electronic cash system”

Yalin Chen¹ and Jue-Sam Chou*²

¹ Institute of information systems and applications, National Tsing Hua University
² Department of Information Management, Nanhua University, Taiwan R.O.C
*: corresponding author: jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

d949702@oz.nthu.edu.tw

Abstract

Recently, Baseri et al. proposed a secure untraceable off-line electronic cash system. They claimed that their scheme could achieve security requirements of an e-cash system such as, untraceability, anonymity, unlinkability, double spending checking, un-forgeryability, date-attachability, and prevent forging coins. They further prove the un-forgeryability security feature by using the hardness of discrete logarithm problems. However, after cryptanalysis, we found that the scheme cannot attain the security feature, untraceability. We, therefore, modify it to comprise this desired requirement, which is very important in an e-cash system.

Keywords: digital signatures, discrete logarithm problem, cryptanalysis, RSA, electronic commerce and payment

1. Introduction

There have been many cryptographic scientists working within the field of e-cash system design [1-33] since Chaum first proposed the concept of e-cash and its paper cash-like properties of anonymity, verifiability, and unforgeability (Chaum 1982) in 1982 [11]. An e-cash system typically contains three roles: customer, bank, and the merchant, and three protocols: withdrawal protocol, payment protocol, and the deposit protocol. In the protocol design principle, the user’s identity cannot be revealed, to assure his purchasing privacy. Conversely, it can be disclosed when double spending or illegal transaction occurs. In an off-line e-cash scheme, the bank cannot prevent the double spending on-line. Therefore, it must have the ability to revoke the anonymity of the user who doubly spent his e-cash. In 2013, Baseri et al. [27] pointed out that Eslami et al.’s untraceable off-line electronic cash system [17] is flawed. It suffers three attacks, double spender detection attack, expiration date forgery, and frauds on exchange protocol. They also proposed an excellent untraceable off-line e-cash scheme system and claimed that their scheme contains anonymity, double spending
detection, un-forgeability, date attachability properties, and forgery prevention. Meanwhile, they also show the reasons why their scheme immune to the three faults that Eslami et al.’s scheme suffers. However, after examining their scheme, we found that it does not have the untraceability property. We, therefore, for enhancing its security, modify it to comprise this feature, which is very important in an e-cash system. We demonstrate it in this article.

2. Review of Baseri et al.’s scheme

Baseri et al.’s e-cash scheme [27] consists of four participants in the scheme: a central authority, the bank, the spender, and the merchant. It contains five phases: initialization, withdrawal, payment, deposit, and the exchange. Meanwhile, they use Chaum’s signature to design the scheme and take advantage of RSA-based method to attach the time to the structure of the signature. The used notations can be referred to the original article. Here, we only list the withdrawal protocol and the payment protocol to illustrate its weakness.

2.1 The withdrawal protocol

The central authority sets some public parameters, including two publicly known elements, \( g_1, g_2 \), and the bank’s two RSA public/private key pairs \(((e_b, n), 1/e_b)\) and \(((e_B, n), 1/e_B)\) such that \( e_B \geq e_b \). Before withdrawing and asking for a coin, the spender should provide his 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. In addition, the bank periodically publishes the fresh time by two parameters, \( t \) and \( e_B^t \pmod{\phi(n)} \), where \( t \) is constant during the period and used to synchronize customers and \( e_B^t \) plays the role of a public key for the bank and is chosen in such a way that its reverse exists. The coin is represented by a five tuple \( (A', B, s_1, s_2, s_3) \). The withdrawal protocol is depicted as follows.

Step 1. The spender S:

(a) Chooses three random numbers, \( x_1, x_2, s \in_R Z_{e_B}^* \) and \( s_1 \in_R Z_{n}^* \), and two

blinding factors, \( b_1, b_2 \in Z_{n}^* \)

(b) Computes: \( A' = A^x \pmod{n} \), \( B = g_1^{s_1} g_2^{s_2} \pmod{n} \), \( w_1 = B b_1^{s_1} \pmod{n} \),

\( w_2 = (A' + B) b_2^{s_2} \pmod{n} \),

(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 \(w_1, w_2\) by computing:
\[
O_2 = w_1^{e_B} \pmod{n}, \quad O_3 = w_2^{e_B + n} \pmod{n}
\]
(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 \(1/e_B, 1/e'_B, 1/(e_B * t)\), respectively:
\[
s_1 = O_1' \pmod{n} = \text{sign}_B(A'), \quad s_2 = O_2 / b_1 \pmod{n} = \text{sign}_B(B),
\]
\[
s_3 = O_3 / b_2 \pmod{n} = \text{sign}_B(A' + B).
\]

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

### 2.2 The off-line payment protocol

The off-line payment protocol is described as follows.

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 whether \(A' \neq 0\).
(b) Checks the coin’s expiration date.
(c) Verifies the signatures, \(s_1\), using the public key, \(e_B\), \(s_2\) using the public key, \(e'_B\), and \(s_3\) using the public key \(e_B * t\).
(d) Computes:
The challenge \(d = H(A', B, ID_M, date || time)\), where \(H\) is the hash function determined in the initialization phase, \(ID_M\) is the merchant’s identity and \(date || time\) represents the transaction’s date and time.
(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.

Step 4. The merchant M:
(a) Accepts the coin if \(g_1^{n} g_2^{x_1} = A^{v_B}\).

### 3. The weakness
An insider attacker can collect the transmitted message on the Internet, and obtain some information as follows:

(1) From the messages in one of the withdrawal protocol executions, the attacker can know the values, \( w_1, w_2, t, O_2 \), and \( O_3 \).

(2) From the messages in the off-line payment protocol, the attacker can know the coin, \((A', B, s_1, s_2, s_3, t)\).

Assuming that the attacker has gathered all \( m \) (with \( m \leq 2^q \), where \( q \) is the security parameter, e.g., \( q=80 \)) coins \((A'_i, B, s_{i1}, s_{i2}, s_{i3}, t_i)\), for \( i=1 \) to \( m \), he then can launch an offline attack for the \( m \) coins using the following ways.

(1) Computes \( O_2^{\ast} = w_1 = Bb_1^{\ast} \pmod{n} \). From this equation, he knows \( b_1^{\ast} = O_2^{\ast} / B \pmod{n} \). Although, he cannot have the right value of \( B \), with the help of \( m \) observed coins \((A'_i, B, s_{i1}, s_{i2}, s_{i3}, t_i)\), he can compute \( b_i^{\ast} = O_{i}^{\ast} / B_i \pmod{n} \), for \( i=1 \) to \( m \). Then, he randomly chooses \( a, f \in z_n^{\ast} \), forms the value \( w_i = a^{s_i} b_i^{\ast} \pmod{n} \), and executes the withdrawal protocol by sending \( w_i, f, t \) to the bank for acquiring \( O_2 = w_i^{\ast} = ab_1, O_3 = f^{d_{\ast}} \). He is able to deduce \( b_1 \) using the value \( a^{-1} \pmod{n} \).

(2) Computes to see if \( O_2 = s_{i2} \cdot b_1 \).

If the equation in (2) holds, the insider knows that the e-cash \((A'_i, B, s_{i1}, s_{i2}, s_{i3}, t_i)\) is related to the parameters \( w_1, w_2, t, O_2 \), and \( O_3 \) in a specific withdraw protocol. Else, he continues through the rest of each of the \( m-1 \) coins. Definitely, he will find one coin satisfying the equation. Thus, the feature of untraceability is violated. Even if \( m \geq 2^q \), the attacker can use the parameter \( t \), observed in the withdraw protocol, to sieve the coins which have the same time \( t \), and then launch the two-step attack shown above.

4. Modification

From the weakness found in section 3, we see that the key point is that the insider can use \( b_1^{\ast} (= O_2^{\ast} / B \pmod{n}) \) to produce \( w_i (= a^{s_i} b_i^{\ast} \pmod{n}) \) for sending to the bank to obtain \( b_1 \), and then check to see if \( O_2 = s_{i2} \cdot b_1 \) holds. To further blur the relation...
between $O_2$ and $s_{t2}$, we introduce two other parameters $b_3 \in \mathbb{Z}_n, x_1 \in \mathbb{Z}_n^*$, and modify $w_i = b_i B b_i^{x_i}$, form $w_{i1} = (b_i b_i^{x_i}) x_i^{x_i}$, and let the spender send $w_i, w_{i1}, w_2, t$ to the bank. Thus, the bank will return $O_2, O_{22}, O_3$. $O_2$ from the bank will become $w_i^{x_i} (mod n) = (b_i B)^{x_i} b_i$ and $O_{22} = b_i^{x_i} b_1 x_1$. After this, the spender can use $x_i^{-1} \mod n$ to acquire $O_{22-x_i} = b_i^{x_i} b_1$ from $O_{22}$, and then take advantage of it to obtain $s_2 = B^{x_i} = O_2 \cdot O_{22-x_i}^{-1}$. Accordingly, if an attacker launches the above attack on our modification; although, he knows $O_{22}$, without $x_i^{-1} \mod n$, he cannot have $O_{22-x_i}$ to form the value $w_i = a^{x_i} O_{22-x_i}^{x_i} (mod n)$, and executes the withdrawal protocol by sending $w_{i1}, f, t$ to the bank for breaking the untraceability.

5. **Conclusion**
In this paper, we showed that Baseri et al.’s untraceable off-line e-cash’s scheme is flawed. It suffers from traceability. We, therefore, for enhancing its security, modified it to avoid this weakness. From the analysis shown in section 4, we see that we have reached the goal of the security promotion.
References


