Crypto-analyses on “user efficient recoverable off-line e-cashs scheme with fast anonymity revoking”

Yalin Chen¹ and Jue-Sam Chou²

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, Fan et al. proposed a user efficient recoverable off-line e-cash scheme with fast anonymity revoking. They claimed that their scheme could achieve security requirements of an e-cash system such as, anonymity, unlinkability, double spending checking, anonymity control, and rapid anonymity revoking on double spending. They further formally prove the unlinkability and the unforgeability security features. However, after crypto-analysis, we found that the scheme cannot attain the two proven security features, anonymity and unlinkability. We, therefore, modify it to comprise the two desired requirements which are very important in an e-cash system.

1. Introduction

There have been many cryptographic scientists working within the field of e-cash system design [1-21] since Chaum first proposed the concept of e-cash and its paper cash-like properties of anonymity, verifiability, and unforgeability (Chaum 1982) in 1982 [1]. 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, Fan et al. [16] proposed an excellent off-line e-cash scheme with fast anonymity revoking. They claimed that each user possessed anonymity and un-linkability, when spending e-cash in their scheme, and the user is allowed to recover his e-cash when lost. Besides, the bank can detect the double spending and efficiently derive the identity of the user, without any help of the TTP. Moreover, TTP can revoke the anonymity of the e-cash owner when illegal transaction occurs. Additionally, their scheme allows the police to trace a specific user. However, after examining their scheme, we found that it does not have anonymity and
un-linkability. We, therefore, for enhancing its security, modify it to comprise these two features which are very important in an e-cash system. We demonstrate it in this article.

2. Review of Fan et al.’s IBS scheme

Fan et al.’s e-cash scheme [16] consists of two main protocols: the withdrawal protocol and the payment (and deposit) protocol, and four entities user, bank, shop and the judge. Meanwhile, they use Chaum’s signature and the chameleon hashing functions to design the scheme. 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 scheme assumes that the bank can authenticate the user through a secure channel. They omit the design relating to this portion. The withdrawal protocol is depicted as follows.

1. User→Bank: $E_{pk_j}(k,m,r)$.

The user randomly chooses three strings $(k,m,r)$, where $k \in \{0, 1\}^l$ and $m \in \mathbb{Z}_q^s$. Then, he sends $E_{pk_j}(k,m,r)$ to the bank.

2. Bank→The judge device: $(E_{pk_j}(k,m,r), \mu)$.

After the bank authenticates the user, it knows the user’s identity $ID_u$. It then sets $\mu = ID_u$, and inputs $E_{pk_j}(k,m,r)$ and $\mu$ into the judge device.

3. The judge device→Bank: $(\beta, E_q(x, \overline{x}, c, k, \delta))$.

After receiving $E_{pk_j}(k,m,r)$ and $\mu$, the judge device uses $sk_j$ to decrypt $E_{pk_j}(k,m,r)$ and gets $(k,m,r)$. Then, it randomly chooses three strings $(r_1, r_2, c)$, where $r_1, r_2 \in \{0, 1\}^l$ and $c \in \mathbb{Z}_b^s$ and computes $x = (\mu \| r_1) \in \mathbb{Z}_q^s$, $\overline{x} = x^{-1} \mod q$, $\delta = E_{pk_j}(\mu, r_2)$, and $y = g^x \mod p$. Finally, it computes $\beta = (c^{-1})^b (g^m y^c \mod p)H(\delta \| y) \mod n_h = (c^{-1})^{n_h} h_{jk}(m,r)H(\delta \| y) \mod n_h$ and outputs
(β, Ek(x, x, c, k, δ)) to the bank, where HK = (p, q, g, y).

4. Bank→User: (t, Ek(x, x, c, k, δ)).

After receiving (β, Ek(x, x, c, k, δ)) from the judge device, the bank computes

t = \beta^d \mod n_b

and returns (t, Ek(x, x, c, k, δ)) to the user. Then the bank stores (ID_u,

E_{pik}(k, m, r), Ek(x, x, c, k, δ)) for e-cash tracing and recovery.

5. Unblinding: After receiving (t, Ek(x, x, c, k, δ)), the user decrypts

Ek(x, x, c, k, δ) and parses the 4th parameter in the decryption result as k’.

Then he checks whether k’ = k. If it’s true, he computes \Sigma = ct \mod n_b. At last, the user obtains an e-cash \((\Sigma, y, m, r, \delta)\).

### 2.2 The off-line payment protocol

The off-line payment protocol is described as follows.

1. Shop→User: (m').

   When a user makes a payment to a shop, the shop will randomly choose a string rs
   and compute m' = (ID_s || rs), such that m' \in \mathbb{Z}_q^*, where ID_s is the shop’s identity.

   Then the shop sends m' to the user.

2. User→Shop: (\Sigma, y, r', \delta).

   After receiving m', the user computes r' = x(m + xr - m') \mod q. \quad (1)

   Then, he sends (\Sigma, y, r', \delta) to the shop.

3. Shop→Bank: (\Sigma, y, m', r', \delta).

   After receiving (\Sigma, y, r', \delta), the shop verifies if \Sigma_s = h_{hk}(m', r')H(\delta \| y) \mod n_b,

   where HK = (p, q, g, y). If it’s true, shop accepts the e-cash and stores

   (\Sigma, y, m', r', \delta) . Later, the shop will send the bank the received e-cash.

4. Bank: acceptance or rejection.

   The shop deposits e-cash (\Sigma, y, m', r', \delta) to the bank. The bank first verifies it by
   checking if \Sigma_s = h_{hk}(m', r')H(\delta \| y) \mod n_b and (\Sigma, y, \delta) has not existed in the
database. If both are true, the bank stores e-cash \((\Sigma, y, m', r', \delta)\) in the database and deposits it into the shop’s account.

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

(1) From message 2, 3, and 4 in the withdrawal protocol, the attacker can know the values, \(\mu, \beta, \text{ and } t\).

(2) From message 3 in the off-line payment protocol, the attacker can know the values, \((\Sigma', y', m', r', \delta')\).

He then can launch an offline attack by the following ways.

(1) Computes \(e' = \sum' t^{-1} \mod n_b\)

(2) Computes to see if \(\beta = [(c')^{-1}]^{\circ} h_{pk}(m', r')H(\delta \| y')\).

If the equation in (2) he knows that the e-cash \((\Sigma, y, m', r', \delta)\) owner is \(\mu ( = IDc)\). Thus, the features of anonymity and un-linkability are broken.

4. Modification
From the weakness found in section 3, we see that the key point is that \(\mu\) and \(t\) in messages 2 and 4 of the withdrawal protocol were not hidden from the attacker. This makes it suffer from the above attack. To enhance, we hide the two parameters into \(E_{pk} (k, m, r)\) and \(E_k (x, \overline{x}, c, k, \delta)\) to become \(E_{pk} (k, m, r, \mu)\) and \(E_k (x, \overline{x}, c, k, \delta, t)\), respectively.

Accordingly, if an attacker launches the above attack on our modification; although, he knows \(\beta\), without the value of \(t\), he cannot break the un-linkability; and without the value of \(\mu\), the anonymity is assured.

5. Conclusion
In this paper, we showed that Fan et al.’s recoverable off-line e-cash’s scheme is flawed. It suffers from linkability and identity leakage. We, therefore, for enhancing its security, modified it to avoid these two weaknesses. From the analysis shown in section 5, we see that we have reached the goal of the security promotion.
References


