# A Secure Fair E-cash Payment Protocol

Thae Nu Nge, and Aye Su Hlaing

***Abstract*—**E-cash payment systems refer to the technological breakthrough that enables us to perform financial transactions electronically. In an electronic commerce environment, the merchant and the consumer are unlikely to trust each other. Properly combining the payment protocol with a fair exchange procedure, the fair e-cash payment scheme allows the consumer and the merchant to fairly exchange their money and goods. This paper analysis and addresses the security flaws in a fair e-cash payment system which is based on DSA signature with message recovery and proposes a solution that would ensure user authentication and data integrity. The system also defenses against threats and misbehaviors related to unfairness and repudiation coming from insiders parties of the transaction.

***Keywords*—** e-commerce, fairness, security, offline, e-cash

## I. Introduction

As more business is conducted over the Internet, the fair-exchange problem is gaining greater importance. The fair e-cash scheme properly combines the payment protocol with a fair exchange procedure, to fairly exchange money and goods [2]. The fairness has been described with a lot of definitions. An exchange is fair if at the end of exchange, each party receives the expected item or neither party receives any useful information about the other's item [10],[ 11].

There are two different types of electronic cash systems: on-line and off-line. In an on-line e-cash system [7], the issuing bank should participate in the payment protocol to verify the e-cash. This may be a straightforward way to make sure of the validity of payments, but it is inefficient for real-time transactions. An off-line system can enhance performance [5] in which the bank is not required to be present to verify the e-cash during the payment procedure. The scheme can maintain fairness with the aid of an off-line trusted third party (off-line TTP or bank). That means in a normal case, the consumer and the merchant can receive their desired items without TTP's participation. However, only when a dispute occurs, the TTP can help both parties resolve the problem and ensure the fairness of the transaction. Numerous mechanisms have been proposed for offline e-cash system in the last decade. Security flaws in several systems have been discovered some time after their proposal. In many of the existing electronic cash systems, the banks and other third authorities are assumed to be trustworthy, and the insider attacks by untrusted authorities are not paid attention to. In paper [1] proposed an efficient e-cash system based on DSA multi-signature in which there is no withdrawal stage and e-cash is produced by consumer. It is very efficient because of not only reducing communication cost but also avoiding the storage and lose problem [8], [9]. However, from the view point of preventing crimes, the security of e-cash system is weak. The scheme does not satisfy the unforgeable property since an adversary can fake a signature for the consumer after the exchange phase. This paper addresses the issue of the security of e-cash system based on DSA signature with message recovery feature and adopts the concept of public key cryptosystem to e-cash procedure while still maintaining the efficiency but enhancing e-cash security.

The paper is organized as follows. Briefly introduce the concept of the DSA signature with message recovery feature in Section 2. In Section 3, a brief description of fair e-cash scheme based on DSA multi- signature is shown. In Section 4, improved payment system that satisfies the designed properties is proposed. Finally, the conclusion is given in Section 5.

## II. DSA Signature Scheme with Message Recovery Feature

This section briefly describes the concept of the message recovery feature of DSA signature [3],[4],[14],[15]. Let $p$ be a large prime, $q$ be a large integer factor of $p-1$ and an element $g \in Z_p^*$ whose order is $q$. Let $x$ is the private key, $y = g^x \bmod p$ is the public key, $k$ is random number $k \in Z_q$. The signature $\delta$:( $r$ , $s$ ) of a message $m$ is

$r = m\, g^k \pmod p$, $r' = r \pmod q$ ,

$s = k - r'x \pmod p$

The message m can be recovered from ($r$ , $s$) correctly,

$m = g^s y^{r'} r \pmod p$

and then the public key $y$ is also verified indirectly.

## III. Review of Fair E-cash Payment Scheme

In this section, a brief description of fair e-cash payment scheme based on DSA signature is presented. The basic scheme consists of three participants and four processes: set up process, exchange process, deposit process and dispute resolution process. Fig. 1 represents basic e-cash scheme.

Thae Nu Nge, Faculty of Information and Communication Technology, University of Technology Yatanarpon Cyber City, Pyin Oo Lwin, Myanmar (e-mail:thaenunge11@gmail.com).

Dr.Aye Su Hlaing, Faculty of Information and Communication Technology, University of Technology Yatanarpon Cyber City, Pyin Oo Lwin, Myanmar (e-mail:yinnsyut@gmail.com).

http://dx.doi.org/10.15242/IIE.E0314070

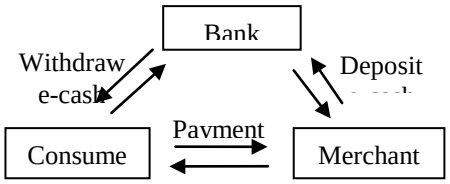


Fig. 1 The basic e-cash scheme

*A. Setup Process*

In setup process, national bank generates ($x_B$ , $y_B$) and publish public key. National bank issues certification for the other branch bank $i$ , $CA_{Bi} = E_{xB}( y_{Bi} )$ to prove the branch bank's validity. ( $x_{Bi}$ , $y_{Bi}$ ) are the secret key and public key of branch bank $i$.

The consumer generates $p$, $q$, $g$, $x$, $x_1$, $y = g^x$ and $y_1 = g^{x1}$, and opens $p$, $q$, $g$. For a supposed random exchange information $m$, consumer computes e-cash (denote as $\delta = (r, s)$) $r = m\, g^k \pmod p$ ), $r' = r \pmod q$ ), $s = k - r'x \pmod p$ ) and commitment $\delta_1 = (r_1, s_1)$, $r_1 = mg^k \pmod p$, $r_1' = r_1 \pmod q$ ) , $s_1 = k - r_1' x_1 \pmod p$ ) .

Consumer contacts bank $i$ to get the public key $y$ certified. The arbitration key $x_2$ is used by bank to make a fair dispute resolution when there is a dissension between user and merchant. The consumer sends $y$, $y_1$ , $\delta$, $\delta_1$ , $x_2$, $m$, $ID_c$ to bank $i$.

Bank $i$ checks $m = g^s y^{r'} r \pmod p$, $m = g^s y_1^{r1'} r_1 \pmod p$ , $s = s_1 + r_1' x_2 \pmod p$ is valid. After verifying the construction of arithmetic, bank $i$ issues a signed certificate $CA_c$ and an overdraft credit voucher $V_c$ to consumer, where

$$V_c = Sig_{Bi}( y_1 \,||\, N \,||\, E_\psi ( x_2 || ID_c ) ) ,$$

$$CA_c = ( E_{xBi} ( y ) \,||\, CA_{Bi} ) , CA_{Bi} = E_{xB} ( y_{Bi} )$$

$N$ stipulates the largest value of an e-cash which consumer can overdraft based on credit. After the setup process, consumer has ($x$, $y$), ($x_1$, $y_1$), $x_2$, $V_c$, $CA_c$, and bank $i$ has his secret arbitration key $x_2$, and $y$, $y_1$, $V_c$ , $CA_c$.

*B. Exchange Process*

When the consumer wants to purchase the digital merchandise, consumer and merchant cooperate to do the following steps.

1. C ⟶ M : $V_c$ , $CA_c$ , $\delta_1$
2. C ⟵ M : $E_r (u)$
3. C ⟶ M : $\delta$

Firstly, the consumer select a random number $k$ and compute $\delta_1(r_1, s_1)$ on the purchase information $m$ (which might contain consumer's unique identity, merchant's unique account number, price of the merchandise, description of the merchandise, and date of transaction) and sends $\delta_1$ , $V_c$ and $CA_c$ to merchant.

Second, merchant can verify the bank's public key $y_{Bi}$ and consumer public key $y$ using the national bank' public key $y_B$ from $CA_c$ . From $V_c$ , he obtains public key $y_1$ and checks $N$. If all items are valid, merchant sends the encrypted merchandise $E_r(u)$ to consumer. Otherwise, merchant does not send the merchandise, and stops the protocol.

Finally, if consumer satisfies the merchandise, he computes the e-cash $\delta$ and sends it to merchant. Otherwise, consumer stops the protocol.

After receiving e-cash $\delta$, Merchant verifies $\delta$ using $y$. If it is valid, merchant ends the protocol. Otherwise, merchant initiates the dispute resolution protocol.

*C. Deposit Process*

Merchant sends the e-cash $\delta$ and $CA_c$ to merchant's bank $j$. After checking the validity of e-cash $\delta$, bank $j$ request bank $i$ to transfer financing from consumer's accounts. Bank $i$ automatic provide a loan to consumer.

*D. Dispute Resolution Process*

If merchant does not receive the e-cash $\delta$, or if $\delta$ is invalid, he initiates these process.

1. M ⟶ B : $V_c$ , $CA_c$ , $\delta_1$ , $E_r(u)$, $E_{yBi} (r)$
2. M ⟵ B : $\delta$

Merchant encrypts the session key $r$ using $y_{Bi}$ with an asymmetric encryption algorithm, then sends $V_c$ , $CA_c$ , $\delta_1$, $E_r(u)$, $E_{yBi} (r)$ to bank $i$.

Bank $i$ decrypts $E_{yBi} (r)$ using his private key $x_{Bi}$, and uses $r$ to recover $u$. Next, he extracts all the system parameters and keys from $CA_c$ and $V_c$ , and then verifies $\delta_1$ using those values. If everything is in order, bank $i$ generates the e-cash using $\delta_1$ and his secret arbitration key $x_2$ as follow: $r = r_1$ , $r_1' = r_1 \bmod q$ , $s = s_1 + r_1' x_2 \bmod p$.

The e-cash $\delta$ is sent to merchant and the encrypted merchandise is forwarded to consumer. Otherwise, if any of the items received from merchant is invalid, bank $i$ halts the dispute resolution protocol without sending anything to either party.

*E. Attack on Fair E-cash Payment Scheme*

The DSA signature with message recovery feature is vulnerable to existential forgery attacks, that is, given a valid signature of a known message, an adversary can forge a valid signature of another different message without the knowledge of the secret key [3],[12],[13]. A forger gets A's signature ($r$ , $s$) for a message $m$. Then the forger can compute a signature ( $r'$ , $s'$) for a message m' without the knowledge of A's secret key by the following procedure. The forger computes

$$r_1' = (mr^{-1})\, g^{-1} = r_1\, g^{-1} = g^{k-1} \pmod p$$

Then, sets a message $m' = m\, g^{-1} \pmod p$ ,

$$r' = r \text{ and } s' = s\text{-}1.$$

Sends ($r'$ , $s'$) as a signature of $m'$, ($r'$ , $s'$) is a valid signature of m' since

$$\begin{aligned} g^{s'} y^r r &= g^{s-1} y^r r \\ &= g^s\, y^r\, r\, g^{-1} \\ &= m\, g^{-1} \\ &= m' \pmod p \end{aligned}$$

By this procedure, an adversary can make the signature on a message $mg{-}1$ and also can generate a signature for any

message in a subset $Sm,g = \{mg^{-n} | n \in Z_q\}$, within one time known-message attack.

It is obvious that an adversary can forge consumer's e-cash after he got the real e-cash by known message attack after the exchange process. Hence, the faked e-cash can be verified successfully and there is no evidence that whether merchant makes deposit with the consumer's real e-cash or not. Moreover, malicious merchant can make the illegal purchase with $V_c$ , $CA_c$ and fake e-cash. It may cause a great financial loss to the business partners and cannot guarantee the fairness of the exchange. In addition, a malicious bank can forge a fact of honest merchant's double deposit.

## IV. Improved Fair E-cash Payment Scheme

This section presents the improved fair e-cash payment scheme in order to solve the above flaws. The proposed solution is straightforward and it should not require to much modifications in the overall system. In addition to certified public key *y*, consumer applies another certified public key *e* to generate cipher of e-cash signature for verification in payment process.

Before registration process, the consumer needs to select two large prime numbers: p and q. Modulus n is : $n = p \times q$. A number *e* is chosen that is $0 < e < (p-1) \times (q-1)$ and also co-prime: gcd $(e; [(p-1) \times (q-1)]) = 1$. The public key is: **(** *n*, *e***)**. Private key *d* is $d = e^{-1} (\mathrm{mod}\ (p-1) \times (q-1))$.

### A. Setup Process

The registration process is the same procedure as the above protocol except the consumer submits the public key **(** *n* , *e* **)** and the encrypted message *c*, $c_1$ to verify the validity of e-cash to bank *i* . Consumer computes $c = s^d \bmod n$ , and $c_1 = s_1^d \bmod n$.

Next, consumer sends $y$ , $y_1$ , $e$ **,** $\delta$, $\delta_1$ ,$c$ ,$c_1$, $x_2$, $m$, $ID_c$ to bank *i*. Bank *i* checks consumer's credit file by performing the following operations.

$$s = c^e \bmod n \ , \ s_1 = c_1^e \bmod n$$

Bank *i* checks $m = g^s y^{r'} r \ (\mathrm{mod}\ p)$ , $m = g^{s1} y_1^{\ r1'} r_1 \ (\mathrm{mod}\ p)$ , $s = s_1 + r_1' x_2 \ \mathrm{mod}\ p$ is valid. After verifying the construction of arithmetic, bank *i* issues a signed certificate $CA_c$ and an overdraft credit voucher $V_c$ to consumer where

$$V_c = Sig_{Bi} ( y_1 \,||\, N \,\|\, e \,||\, E_\psi ( x_2 || ID_c ) ) ,$$

$$CA_c = ( E_{xBi} ( y ) \,||\, CA_{Bi} ) , \ CA_{Bi} = E_{xB} ( y_{Bi} )$$

After the setup process, consumer has $(x, y)$, $(x_1, y_1)$, $x_2$, $(e,d)$, $V_c$, $CA_c$ and bank i has consumer's authentic public key *e and* $y, y_1, x_2$ , $V_c$ , $CA_c$ .

### B. Exchange Process

In the exchange process, the consumer and merchant perform the following process.

1. C ⟶ M : $V_c$ , $CA_c$ , $\delta_1$ , $c_1$
2. C ⟵ M : $E_r (u)$
3. C ⟶ M : $\delta$ , $c$

At first, the consumer select a random number *k*, and compute $\delta_1$:($r_1$, $s_1$). To prevent from the signature $\delta_1$:($r_1$, $s_1$) modified or forged, the consumer encrypts part of signature $s_1$ with his private key *d*, $c_1 = s_1^d \bmod n$ and sends $\delta_1$ , $c_1$, $V_c$ and $CA_c$ to merchant.

Merchant can verify consumer's public key *y* from $CA_c$. Merchant can verify the commitment $\delta_1$:($r_1$, $s_1$) using $y_1$ and *e* as follow. Using consumer's certified public key **(** *n* , *e* **)**, merchant checks whether the commitment signature is modified or forged.

$$s_1' = c_1^e \bmod n$$

If $s_1' = s_1$ , computes, $m = g^{s1} y_1^{\ r1'} r_1 \ (\mathrm{mod}\ p)$

If all items are valid, merchant sends the encrypted merchandise $E_r(u)$ to consumer. Otherwise, merchant does not send the merchandise, and stops the protocol.

Finally, receiving expected merchandise, consumer computes $\delta$:($r$, $s$) and *c* where,

$$c = s^d \bmod n$$

Merchant checks the validity of $\delta$ and *c* using *y* and *e* respectively as follow.

$$s = c^e \bmod n$$

$$m = g^{s1} y_1^{\ r1'} r_1 \ (\mathrm{mod}\ p)$$

If both of them are valid, merchant ends the protocol. Otherwise, merchant initiates the dispute resolution protocol.

### C. Deposit Process

Merchant sends the e-cash $\delta$ , *c* and $CA_c$ to merchant's bank *j*. Bank *j* verifies $CA_{Bi}$ using $y_B$, verifies $E_{xBi} ( y )$ using $y_{Bi}$, verifies e-cash using *y*. In addition to checking e-cash $\delta$:($r$, $s$), the bank *j* needs to check the authenticity of signature by performing the following operations.

$$s' = c^e \ mod \ n$$

If s' doesn't match with s, the e-cash signature is onsidered as a modified or forged one by merchant and bank j won't accept the deposit of merchant. If $s' = s$ , then the e-cash signature $\delta$:($r$, $s$), is considered as a valid one generated by the consumer. If e-cash δ has not been deposit, bank j deposits it for merchant in her account.

### D. Dispute Resolution Process

If merchant does not receive the e-cash $\delta$, or if $\delta$ is invalid, he initiates these process.

1. M ⟶ B : $V_c$ , $CA_c$ , $\delta_1$ , $c_1$ , $E_r(u)$, $E_{yBi} (r)$
2. M ⟵ B : $\delta$

After receiving $V_c$ , $CA_c$ , $\delta_1$ , $c_1$ , $E_r(u)$, $E_{yBi}$ $(r)$ from merchant , bank *i* decrypts $E_{yBi}$ $(r)$ using his private key $x_{Bi}$, and uses *r* to recover *u*. Next, he extracts all the system parameters and keys from $CA_c$ and $V_c$ , and then verifies $\delta_1$ using those values.

And then, bank *i* checks the authenticity of the commitment signature by computing the following expression.

$$s_1' = c_1^e \ mod \ n$$

If $s_1' = s_1$, then the commitment signature is considered as a valid one generated by the consumer, not a modified one by the merchant.

If everything is in order, bank *i* generates the e-cash $\delta$ using $\delta_1$ and his secret arbitration key $x_2$ as follow: $r = r_1$ , $r_1' = r_1 \bmod q$ , $s = s_1 + r_1' x_2 \bmod p$.

The e-cash $\delta$ is sent to merchant and the encrypted merchandise is forwarded to consumer. Otherwise, if any of the items received from merchant is invalid, bank *i* halts the dispute resolution protocol without sending anything to either party.

### *E. Security Analysis*

In this section, the security issues with respect to the proposed system will be discussed.

1. *Authentication:* The merchant and the bank use consumer's certified public key *d* for authenticating the consumer. Preventing malicious merchant make use of $CA_c$, $V_c$ and fake signature for another purchase because he can't prove that he is the owner of $CA_c$ and $V_c$ without knowing *d.*

2. *Non-repudiation*: The improved scheme ensures the evidence of origin and fullfils non-repudiation. Hence, after the exchange phase, consumer cannot deny that he had spent the e-cash because nobody can compute *c* for e-cash verification.

3. *Integrity*: If merchant deposit forged or modified e-cash (*r',s'*), bank can check immediately. Although merchant can successfully achieve the verification of the forge e-cash using *y*, he can't produce *c* without knowing consumer's private key *d*. The merchant can't make a deposit with the forge e-cash in the deposit process because bank will not accept the e-cash if *s'* don't match the *s.* It also prevents the dishonest merchant to initiate the dispute resolution process with a fake e-cash. Hence, the improved scheme prevents the effects of existential forgery attack and ensuring the integrity of the e-cash.

4. *Impersonation*: Because the consumer's secret key *d* is not stored in the database of the bank, the malicious bank employee can't produce e-cash from the honest customer's account by impersonating the consumer with the secret key. Therefore, this incomplete information (for the bank) enhances security against the impersonation by the malicious bank [6].

## V. Conclusion

This paper addresses the security issue of e-cash system which is based on DSA based message recovery signature and shows the security weakness of that system. The proposed protocol adopts the same transaction scheme which is based on DSA based multi-signature. While still maintaining the efficiency, the improved scheme satisfies the fairness property since an adversary can't produce an authentic e-cash without knowing the consumer's certified private key. Hence, integrating authentic public key for e-cash verification makes the fair offline e-cash payment systems securely workable. In the future, it needs to formalize both the protocol and the security requirements to demonstrate that the protocol satisfies the desired security properties using one of the formal verification methods such as AVISPA.

## Acknowledgment

I would like to thank my supervisor and all of my teachers for their helpful comments in improving our manuscript.

## References

[1] Zhaoxia, Wang Shaobin, Nu Shuwang ," A DSA Multi-Signature Protocol and Applying in E-Bank and E-Voting ", 978-1-4244-5895-0, 2010 IEEE.

[2] C-H Wang and W-M Chiang, "The Design of a Novel E-cash System with the Fairness Property and Its Implementation in Wireless Communications", Department of Computer Science and Information Engineering National Chiayi University , Journal of Computers Vol.18, No.2, July 2007.

[3] Atsuko Miyaji, "Weakness in Message recovery signature scheme based on discrete logarithm problems 1", IEICE Japan Tech. Rep., ISEC95-11, 1994.

[4] Y.-M. Tseng, "Digital signature with message recovery using self-certified public keys and its variants", Applied Mathematics and Computation 136 (2003) 203–214.
http://dx.doi.org/10.1016/S0096-3003(02)00010-3

[5] C. Popescu, "A Secure E-Cash Transfer System based on the Elliptic Curve Discrete Logarithm Problem", INFORMATICA, 2011, Vol. 22, No. 3, 395–409.

[6] T.Nishide, S.Miyazaki, K.Sakurai, "Security Analysis of Offline E-cash Systems with Malicious Insider", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 3, number: 1/2, pp. 55-71.

[7] Al-Fayoumi, Mohammad. "Practical E-Payment Scheme", *International Journal of Computer Science Issues* (IJCSI)/16940784, 20100501

[8] Lei Hu, "Fair E-cash Payment Model on Credit O", 2006 *IEEE Asia-Pacific Conference on Services Computing* (APSCC 06), 12/2006.

[9] Guohua Cui, "A fair e-cash payment scheme based on credit", *in Proc. 7th international conference on Electronic commerce* - ICEC 05 ICEC 05, 2005.

[10] Xian Zhu. "Optimistic fair-exchange protocols based on DSA signatures", *IEEE International Conference on Services Computing* 2004 (SCC 2004) Proceedings 2004, 2004

[11] Xinmei Wang. "Fair Exchange Signature Schemes", 22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008), 03/2008

[12] Zichen Li. "A new forgery attack on message recovery signatures", *Journal of Electronics* (China), 07/2000

[13] Kaisa Nyberg. "Message recovery for signature schemes based on the discrete logarithm problem", Lecture Notes in Computer Science, 1995
http://dx.doi.org/10.1007/BFb0053434

[14] Gao, C. zhi, D. Xie, J. Li, B. Wei, and H. Tian. "Deniably Information-Hiding Encryptions Secure against Adaptive Chosen Ciphertext Attack", *Fourth International Conference on Intelligent Networking and Collaborative System*s, 2012.
http://dx.doi.org/10.1109/iNCoS.2012.88

[15] Kaisa Nyberg. "Message recovery for signature schemes based on the discrete logarithm problem", Lecture Notes in Computer Science, 1995.
http://dx.doi.org/10.1007/BFb0053434