International Journal of Science and Engineering Applications
Volume 8–Issue 08,362-365, 2019, ISSN:-2319–7560

# Improvement of Fairness in E-cash Protocol

Dr. Khin Khat Khat Kyaw
Associate Professor
Information Technology Department
West Yangon Technological University, Yangon
Myanmar

**Abstract**: Nowadays ecommerce applications are widely used in the world since it is easier than conventional payment system. The various e-cash protocols have been proposed into the literature in order to get better efficiency. This paper analyses the e-cash protocol that was designed to provide non-repudiation and anonymity. The protocol was checked with AVISPA tool and the results show that it meets weak fairness. Therefore, a modified e-cash protocol is proposed in this paper. The proposed protocol also maintains anonymity and non-repudiation. So, this paper presents analysis of the original e-cash protocol and modified protocol by using AVISPA tool.

**Keywords**: AVISPA, anonymity, electronic cash, electronic commerce, fairness, non-repudiation

## 1. INTRODUCTION

Although current e-cash systems can make transactions in a few seconds, they do not fully guarantee the client's privacy. The current systems cannot execute the transactions in a completely anonymous way. For example, the bank or payment provider knows the details of the client's transaction.

Physical cash provides better privacy: the payments are difficult to trace as there is no central authority that monitors all transactions, in contrast to most electronic payment systems. This property is the inspiration for 'untraceable e-cash' systems[1].

To be secure, an e-cash protocol should not only ensure the client's privacy, but must also ensure that a client cannot forge coins which were not issued by the bank. Moreover, it must protect against double spending. Otherwise a client could try to use the same coin multiple times. This can be achieved by using on-line payments,i.e., a seller has to contact the bank at payment before accepting the coin, however it is an expensive solution. An alternative solution, which is usually used to support online payments (i.e., a seller can accept the payment without contacting the bank), is revealing the client's identity if he spent a coin twice. Finally, exculpability ensures that an attacker cannot forge a double spend, and hence incorrectly blame an honest client for double spending.

In so-called payment by instruction type of systems, a payer basically orders the bank to move a sum of money from her account into a payee's account[2]. The central security aspect in these systems is to ensure that only legitimate account holders are able to issue payment instructions. Of course, digital signatures are the solution for doing this over a large, open network such as the Internet. Since digital signatures only make sense if there is an infrastructure for certifying public keys, a lot of effort is devoted to just this.

Prepaid systems are conceptually close to electronic equivalents of cash. The central security aspect in this type of system is to ensure that cards or representations of cash cannot be forged. When forgery happens, the float will ultimately be insufficient to credit all of the payees' accounts for received payments. Of course, it should also be ensured that only legitimate account holders can reload cash from their accounts. However, this security aspect is now limited to the infrequent withdrawal protocol, and is no part anymore of the more frequent payment protocol.

A basic requirement of a payment protocol is that it allows a payee to receive payments from any payer. A payment can be seen as some sort of authentication of the payer towards the payee. Authentication can be based on secret key cryptography or on public key cryptography.

## 2. Related Work

The E-cash scheme was the closest to a system that mimicked _at currency with the property that it provided anonymity for users when buying coins from the Bank and spending them when a merchant premises [6].

The authors propose an extension to the E-cash scheme which allows for the anonymous transfer of coins between users without the involvement of a trusted third party. We make use of a powerful technique which allows for distributed decision making within the network - namely the Bitcoin blockchain protocol. Combined with the proof-of-work technique and the classical discrete logarithm problem the proposed protocol is able to continuously reuse coins, and also prevent double-spending of coins without revealing the identities of the users.

The major contribution in [7] is the practical E-cash payment system that provides secrecy and comfortability. The protocol gets security based on X9.59 in which they use Payment Routing Code (PRC) instead of consumer and merchant account/card numbers. It provides security, authentication and integrity.

Non-repudiation is one of the most important security services in electronic commerce. Non-repudiation means that an entity cannot deny its participation in a message exchange. Therefore, non-repudiation protocols provide for undeniable data exchange between two or more principals. Judson Santiago and Laurent Vigneron [8] proposed how to define

non-repudiation properties with the AVISPA tool and explained how they can be automatically verified.

## 3. RONGGONG SONG AND LARRY KORBA'S PROTOCOL

The following section discusses the e-cash protocol provided in [3].

*A. Terminology and Notations*

Terminology and notations used in the paper are defined as follows.

- *A*: a customer
- *B*: a bank
- *ES*: an e-commerce store
- *IDA*: customer *A*'s identity
- *H*(): one-way hash function
- $Z_n$ : the integers modulo *n*
- $Z^*_n$: the multiplicative group of *Zn*
- *M mod n*: residue of *M* divided by *n*
- $Time_A$: time stamp made by customer *A*
- $Sign_A$: customer *A*'s signature
- *gcd(m, n)* : greatest common divisor of *m* and *n*
- *A→B:M*: customer *A* sends message *M* to the bank B
- *RM*: remainder money after *A* purchases the e-goods
- *EMD*: e-goods message digest

*B. E-cash Issue Protocol*

When a customer wants to buy e-goods by using online shopping, he/she first needs to buy some e-cashes. It is issued by the bank using the following protocol where all communications are supported by the SSL security channel.

1. $A \rightarrow B$: $ID_A$, $Account_A$, $PK_A$, α, *v*, *TimeA*, $Sign_A$
2. $B \rightarrow A$: $ID_A$, $ID_B$, β, $Time_B$ , $Sign_B$

***Step 1:*** If a customer decides to purchase an e-cash from the bank, he/she first makes a temporary public key ($e_t$, $n_t$), and keeps its private key ($d_t$, $p_t$, $q_t$) secret (using RSA public key cryptosystem). Then, the customer selects a random integer *r* in $Z^*_{nb}$, and computes $\alpha \equiv (r^{ebv} H(e_t||n_t)) \bmod n_b$) where || denotes the concatenation symbol, and *v* contains the following basic information predefined by the bank, i.e. expiration date and money. Then, the customer computes the signature $Sign_A$ as follows.

$$Sign_A \equiv (H(ID_A, Account_A, PK_A, \alpha, v, Time_A)^{dA} \bmod n_A$$

Finally, the customer sends the bank the messages ($ID_A$, $Account_A$, $PK_A$, *α, v,* $Time_A$, $Sign_A$) by using SSL security channel.

***Step 2:*** After achieving the above messages through the SSL security channel, the bank checks whether or not the messages: *AccountA*, *TimeA*, *SignA*, and *v* are correct. If they are correct, the bank computes $\beta \equiv (\alpha^{(ebv)-1} \bmod nb)$ and the signature:

$$Sign_B \equiv (H(ID_A, ID_B, \beta, Time_B))^{db} \bmod n_b .$$

Then, it sends the messages (*IDA*, *IDB*, β, *TimeB*, *SignB*) to the customer through the SSL security channel. In the meantime the bank deducts the money from the customer's account. Finally, after achieving the messages sent by the bank through the SSL security channel, the customer checks whether or not the messages: *TimeB* and *SignB* are correct. If they are correct, he/she then computes $s \equiv (r\text{-}1\beta \bmod nb)$ as the signature of the bank and gets his/her e-cash (*et*, *nt*, *v*, *s*).

*C. Online Shopping Protocol*

When the customer wants to buy some e-goods like e-book, software, and movie, etc. from the Internet, since it is notnecessary for the shipping service, he/she could use the following protocol. When the customer wants to download the licenses of the e-goods and hide his/her identity, he/she could use that online shopping protocol.

1. $A \rightarrow ES$: *E-goods, Cost,* $Account_{ES}$, $e_t$, $n_t$, *v, s,* $Time_A$,$Sign_t$
2. $ES \rightarrow B$: *Cost,* $Account_{ES}$, $e_t$, $n_t$, *v, s,* $Time_A$, *EMD*,$Sign_t$
3. $B \rightarrow ES$: $Receipt_{ES}$ , $e_t$, $n_t$, *v, s, RM, s',* $Time_B$ , $Sign_B$
4. $ES \rightarrow A$: *License,* $Receipt_A$, $e_t$, $n_t$, *v, s, RM, s',* $Time_{ES}$, $Sign_{ES}$

*D. Security Analysis*

In this system, the bank and merchant do not know anything about the customer except how much money the customer spends for e-cashes. This provides anonymity property for the customers. The owners of the messages signed all transferred messages with their own signatures in the protocol, they can ask a Court to judge it if there is a dispute later. Therefore, the protocol provides the non-repudiation service for the customer, merchant and bank.

However, their protocol still provides weak fairness for the customer. After receiving the correct payment from the bank in step 3, the merchant can deny to send the product decryption key to the customer because the merchant did not send Non-repudiation of Receipt (NRR) to anyone. Therefore, that protocol has weak fairness for customer.

## 4. PROPOSED ONLINE SHOPPING PROTOCOL

The proposed e-cash system only modifies the online shopping protocol. The e-cash issue protocol remains the same. The modified e-cash system consists of three parties: merchant, customer and bank. The bank is considered as Trust Third Party (TTP).

1. $A \rightarrow ES$: *E-goods, Cost,* $Account_{ES}$, $e_t$, $n_t$, *v, s,* $Time_A$, $Sign_t$
2. $ES \rightarrow B$: *E-goods, Cost, License,* $Account_{ES}$, $e_t$, $n_t$, *v, s,* $Time_A$, $Sign_t$, $Sign_{ES}$
3. $B \rightarrow A$: *License,* $Receipt_A$, $e_t$, $n_t$, *v, s, RM, s',* $Time_B$, $Sign_B$
4. $B \rightarrow ES$: $Receipt_{ES}$, $Account_{ES}$, $Time_B$ , $Sign_B$

***Step 1:*** The protocol starts with the customer (A). The customer downloads an encrypted product from the merchant (ES). Then, A sends ES a purchase order, and computes the following signature $Sing_t$ with the private key corresponding to the temporary public key of the e-cash.

$$Sign_t \equiv (H(Cost, Account_{ES}, e_t, n_t, v, s, Time_A) \,||\, H(E\text{-}goods))^{dt} \bmod n_t$$

Then A sends the messages (*E-goods, Cost,* $Account_{ES}$, $e_t$, $n_t$, *v, s,* $Time_A$, $Sign_t$ ) to the ES by using the SSL security channel.

***Step2:*** After receiving the above messages, the merchant checks whether or not the messages: *Cost,*$Account_{ES}$, $Time_A$, $Sign_t$, and $seb_v \equiv (H(et||nt) \bmod nb)$ are correct. If they are correct, the merchant forwards the bank the message (*E-*

*goods, Cost, License, Account$_{ES}$, $e_t$, $n_t$, v, s, Time$_A$, Sign$_t$, Sign$_{ES}$* ).

***Step3:*** The bank verifies whether or not the messages: *Account$_{ES}$, Time$_A$,* and *Sign$_t$* are correct. If they are correct, it deducts the money from the e-cash. Then, the bank computes the remainder money *RM* and the signature:

$s' \equiv (H(e_t, n_t, v, s, RM))^{db} \bmod n_b$

$Sign_B \equiv (H(License, Receipt_A, e_t, n_t, v, s, RM, s', Time_B,))^{dB} \bmod n_B$

The bank makes a receipt for the customer and sends the customer the messages (*License, Receipt$_A$, $e_t$, $n_t$, v, s, RM, s', Time$_B$, Sign$_B$* ). After achieving the messages, the customer obtains the licenses of the e-goods and his/her remainder ecash.

***Step4:*** Finally, the bank then deposits the money into the merchant's account and the bank makes a statement (receipt) for the merchant and sends the messages (*Receipt$_{ES}$, Account$_{ES}$, Time$_B$ , Sign$_B$*) to the merchant.

$Sign_B \equiv (H(Receipt_{ES}, Account_{ES}, Time_B, Sign_B))_{db} \bmod n_b$

The modified version change the step 2,3 and 4. The key idea is that the License for E-goods must be first forwarded to the bank(Trusted Third Party). Therefore, the customer can get the License from the bank even if the merchant deny sending. By this way, the proposed protocol meets the strong fairness for every party.

## 5. SECURITY ANALYSIS

The formal analysis for the original and modified protocol was done in AVISPA Tool. Automated validation of internet security protocols and applications (AVISPA) [4] is a push button tool for the automated validation of security protocols.

A modular and expressive formal language called HLPSL (High level protocols specification language) [5] is used by AVISPA to specify the security protocol and their properties.

HLPSL language is a role-based language, which means that actions of each participant are defined in a separate module, called a basic role. The security of original protocol is verified by using AVISPA. For this, three basic roles are played as Customer (C), Merchant (M) and Bank (B). Basic roles describe what information the corresponding participant has initially (parameters), its initial state and how the state can change (transitions). The users use channels SND (send) and RCV (receive) for communication. Dolev-Yao (dy) is the intruder model that is assumed for the communication channel.

Security goals of the protocol are presented in HLPSL language in section called goals. Security goals are actually defined in transition section of basic roles. The definitions of security goals in transition section are called goal facts. The goals section simply describes which combinations of these goal facts indicate an attack. The following goals are considered: (1) the parties (C and M) shall authenticate each other (2) payment information including customer's bank details shall remain secret from any other parties. Therefore, a goal section of the protocol definition can be as follows:

goal
authentication_ on deal
weak_ authentication_ on deal
secrecy _of order
secrecy_ of payment
end goal

Running the AVISPA tool on the original protocol returns the following output.

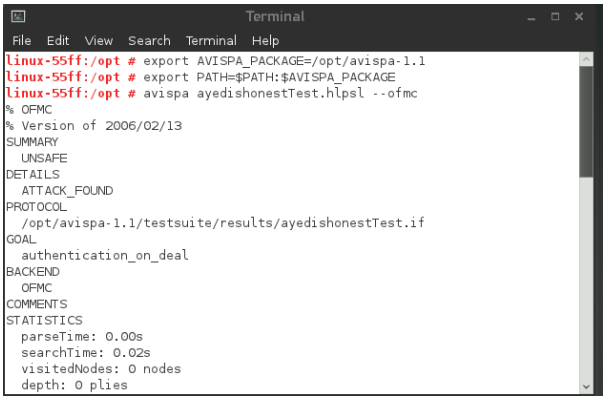


Figure 1. Security Analysis of the Original Protocol

As shown in Figure 1, the AVISPA tool output provides that

SUMMARY
UNSAFE
DETAIL
ATTACK_FOUND

This means that the protocol has been found to be unsafe and that an attack has been found. AVISPA tool found *authentication_on_ deal* attack. This means that both nonrepudiation of origin and non-repudiation of recipient breaks down. Therefore, fairness breaks down.

The proposed e-cash protocol designed to provide the anonymity service for customers and non-repudiation services for all players in the protocol. In the e-cash issue protocol, the customer sends the bank the message that is signed with the customer's certificate. When the customer repudiates this action, the bank can show the customer's signature. On the other hand, if the customer does not do this, the bank also cannot charge the customer because it cannot give evidence (i.e., signature) to prove it.

The proposed system gives the facility to the user, as the customer can make anonymous payment with the merchant as the merchant cannot know the identification of a customer; the merchant can only receive a coin from the user and verify the validity of the signature but cannot determine the customer's identity. Therefore, the customers get strong privacy protection for the e-cash.

The bank needs only to keep the still-alive e-cashes in its database to prevent double-spending because its database can remove all expired e-cash. Moreover, the modified protocol gives strong fairness property for customer because the bank is considered as TTP.

If the merchant did not send the product decryption key(License) to the customer, the bank would directly send the customer the product decryption key. The proposed protocol analyses in AVISPA tool whether it meets fairness, non-repudiation and anonymity.

As shown in Figure 2, the AVISPA tool output shows that

SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS


Figure 2. Security Analysis of Proposed Protocol

This means that the protocol has been found to be safe and an attack has not been found. Therefore, the proposed protocol provides anonymity for customer and non-repudiation for customer, merchant and bank. Moreover, the modified protocol provides strong fairness for all playing parties.

## 6. CONCLUSION

This paper provides the formal analysis of the original protocol. According to the result, the original protocol does not meet fairness for the customer. Therefore, the modified protocol was proposed in order to guarantee fairness for all parties. In addition, the modified protocol maintains anonymity and non-repudiation.

## REFERENCES

[1] Jannik Dreier, Ali Kassem, Pascal Lafourcade. Formal Analysis of E-Cash Protocols. 12th International Conference on Security and Cryptography (SECRYPT 2015), Jul 2015.

[2] Berry Schoenmakers, Basic Security of the ecash Payment System, State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography, Leuven, Belgium, June 3–6, 1997, vol. 1528 of Lecture Notes in Computer Science, pp. 338–352. Springer-Verlag.

[3] Ronggong Song and Larry Korba, “How to Make E-cash with Non- Repudiation and Anonymity” , Proceedings of the International Conference on Information Technology: Coding and Computing, 2004

[4] Avispa - a tool for automated validation of internet security protocols, http://www.avispa-project.org.

[5] Specification of the problems in the high-level specification language, http://www.avispa-project.org.

[6] Hitesh Tewari and Arthur Hughes, Fully Anonymous Transferable Ecash, http://www.scss.tcd.ie

[7] A. Levi, Ç.K. Koç, CONSEPP: CONvenient and Secure Electronic Payment Protocol Based on X9.59, Proceedings of The 17th Annual Computer Security Applications Conference, pages 286-295, New Orleans, Louisiana, IEEE Computer Society Press, Los Alamitos, California, December 10-14, 2001.

[8] Judson Santiago and Laurent Vigneron, Study for Automatically Analysing Non-repudiation.