International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.16
July-2014,
Pages:3318-3321

www.semargroup.org,
www.ijsetr.com

Design and Formal Analysis of E-cash System

AYE THANDAR SWE[1], KHIN KHAT KHAT[2]

[1]Dept of Information and Communication Technology, University of Technology, Yatanarpon, Cyber City, Pyin Oo Lwin, Myanmar, E-mail: ayethandarswe84@gmail.com.

[2]Dept of Information and Communication Technology, University of Technology, Yatanarpon, Cyber City, Pyin Oo Lwin, Myanmar, E-mail: khat81@gmail.com.

Abstract: E-cash system is one of the most popular e-commerce applications. Many researchers tried to improve security such as secrecy, non-repudiation, anonymity, and fairness. Although many researchers have proposed the E-cash payment protocols to the literature, there are few papers that analyze formally those protocols. The current e-cash protocols still need to check whether they meet fairness property. This paper will focus on analyzing non-repudiation, anonymity and fairness properties in e-cash payment protocol by using AVISPA tool. The proposed e-cash system consists of three parties: merchant, customer and bank. The bank is considered as Trust Third Party (TTP). The merchant and customer first need to apply and get their certificates from the bank by opening their accounts in the bank. When the customer wants to buy some e-goods from the Internet, the customer first needs to buy some e-cashes. This paper is to be secure for clients such as customers and merchants. The security architecture of the system is designed by the combination of the partial blind digital signature and anonymous digital signature. Finally, verification of the proposed protocol is done by using the formal verification tool, AVISPA. Therefore, this paper demonstrates the design and implementation of the e-cash system. Then, the proposed e-cash system is verified by using AVISPA tool.

Keywords: Fairness; Non-Repudiation; Anonymity; E-Cash; AVISPA.

I. INTRODUCTION

Electronic payment systems can be classified into four categories. They are Online Credit Card Payment System, Online Electronic Cash System, Electronic Cheque System and Smart Cards based Electronic Payment System. Each payment systems have many problems like denying, losing, misusing, stealing, and double-spending, etc. E-cash system is one of the most popular e-commerce applications. Many researchers tried to improve security such as secrecy, non-repudiation, anonymity, and fairness. Although many researchers have proposed the E-cash payment protocols to the literature, there are few papers that analyze formally those protocols. The current e-cash protocols still need to check whether they meet fairness property. The authors described the basic operations that manipulate E-cash coins in [1].In the withdrawal phase, the payer transfers some of money from the bank account to his or her payment card. In the Payment stage, the payer transfers the money from the card to the payee. In the Deposit process, the payee transfers the money received to the bank account. In the case of security mechanism, for the authentication of messages the authors use the RSA public key cryptosystem as signatures.

The major contribution is secrecy (privacy). It needs to check whether the protocol meets fairness property. In [2], the author described to develop a baisc website where a customer is provided with a shopping cart application and also to know about the technologies used to develop such an application. These include multi- tiered architecture, server and client side scripting techniques, implementation technologies such as ASP.NET. In [3], the author proposed an efficient e-cash system. To provide the non-repudiation service, a one-time public key is embedded in the partial blind signature. In order to get anonymity service and non-repudiation service for the customers and build a fair e-cash system, the author proposed a new e-cash system using a modified partial blind signature scheme proposed by Abe [4]. Their protocol achieves non-repudiation and anonymity services between customer and merchant. However, there still has weak fairness in their protocol. In this paper, we will propose a modified protocol to avoid weak fairness and analyze to show that our modified protocol can achieve more secure. The analysis and verification of the proposed protocol is done using AVISPA (Automated validation of internet security protocols and applications).The rest of paper is organized as follows. The next section explains the proposed e-cash system. Then, section 3 presents the implementation of the system and analyses the proposed system using AVISPA. The last section presents our conclusion.

II. PROPOSED E-CASH SYSTEM

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

- A: a customer
- B: a bank
- ES: an e-commerce store
- ID_A: 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

A. 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.

1. A → B: ID_A, $Account_A$, PK_A, α, v, $Time_A$, $Sign_A$
2. B → 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^*n_b, 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.

$$Sign_A \equiv (H(ID_A, Account_A, PK_A, \alpha, v, Time_A)\, d_A \bmod n_A$$

Finally, the customer sends the bank the messages (ID_A, $Account_A$, PK_A, α, v, $Time_A$, $Sign_A$)

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

$$Sign_B \equiv (H (ID_A, ID_B, \beta, Time_B))d_b \bmod n_b .$$

Then, it sends the messages (ID_A, ID_B, β, $Time_B$, $Sign_B$) to the customer. 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: $Time_B$ and $Sign_B$ are correct. If they are correct, he/she then computes $s \equiv (r^{-1}\beta \bmod n_b)$ as the signature of the bank and gets his/her e-cash (e_t, n_t, v, s).

B. Online Shopping Protocol

The online shopping e-cash system consists of three parties: merchant, customer and bank. The bank is considered as Trust Third Party (TTP). The merchant and customer first need to apply and get their certificates from the bank by opening their accounts in the bank. When a customer wants to buy e-goods by using online shopping, a customer could use the following protocol and e-cash to purchase and download the licenses of the e-goods.

1. A→ES: E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$

2. ES→B: E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$, $Sign_{ES}$

3. B→A: License, $Receipt_A$, e_t, n_t, v, s, RM, s', $Time_B$, $Sign_B$

4. B→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.

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

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

Step2: After receiving the above messages, the merchant checks whether or not the messages: Cost, $Account_{ES}$, $Time_A$, $Sign_t$, and $s^{ebv} \equiv (H (e_t||n_t) \bmod n_b)$ are correct. If they are correct, the merchant forwards the bank the messages (E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$).

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

$$s' \equiv (H(e_t, n_t, v, s, RM))d_b \bmod n_b$$

$$Sign_B \equiv (H (License, Receipt_A, e_t, n_t, v, s, RM, s', Time_B,)) d_B \bmod n_B$$

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 e-cash.

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.

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

III. EXPERIMENTAL RESULTS

A. Implementation of the System

The proposed system is implemented using Java programming language and Microsoft Office Access 2007 Database. The proposed system consists of three parties: merchant, customer, and bank. The bank behaves as the

trusted third party (TTP). The merchant and customer first need to apply and get their certificates from the bank by opening their accounts in the bank as shown in Figs.1 and 2.

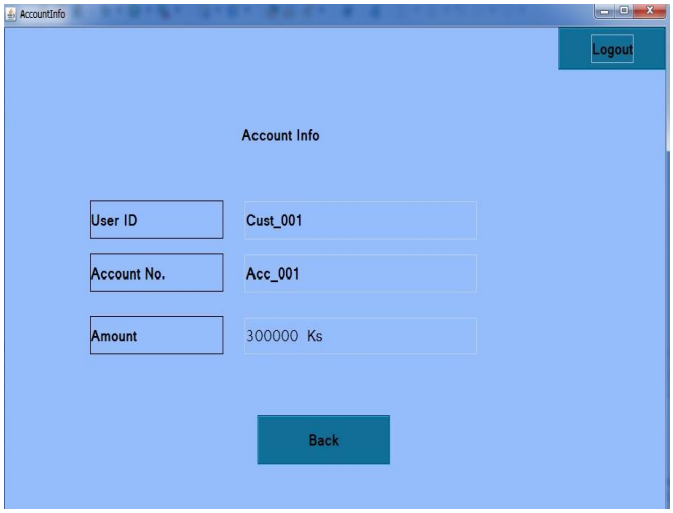

Fig.1. Account Information of customer.

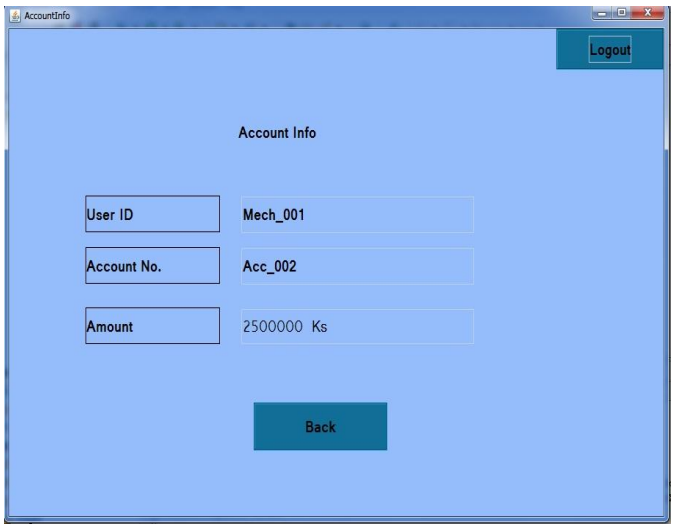

Fig.2. Account Information of merchant.

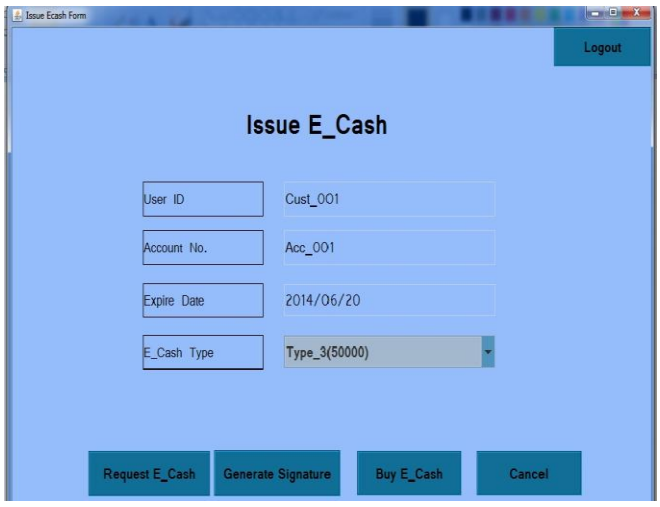

Fig.3. Issue E-cash at the bank.

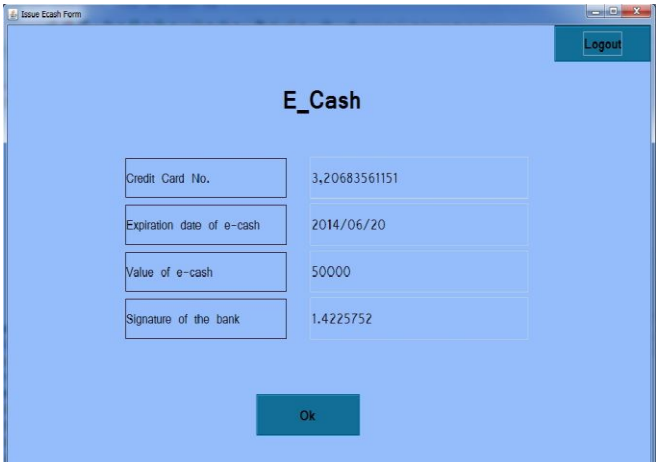

Fig.4. E-cash.

When a customer wants to buy e-goods by using online shopping, he/she first needs to buy some e-cashes at the bank. When a merchant wants to sell e-goods, he must register e-goods at the bank. The merchant sends the e-goods, its description which includes the cost, and a key pair (K, K^{-1}) to the bank. Then, the merchant encrypts the e-goods with key K and advertises it on the web as shown in Figs.3 and 4. When the customer wants to buy some e-goods by using online shopping with the e-cash, he/she first selects the e-goods, and computes $Sign_t$ with the private key corresponding to the temporary public key of the e-cash as shown in Fi.5,

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

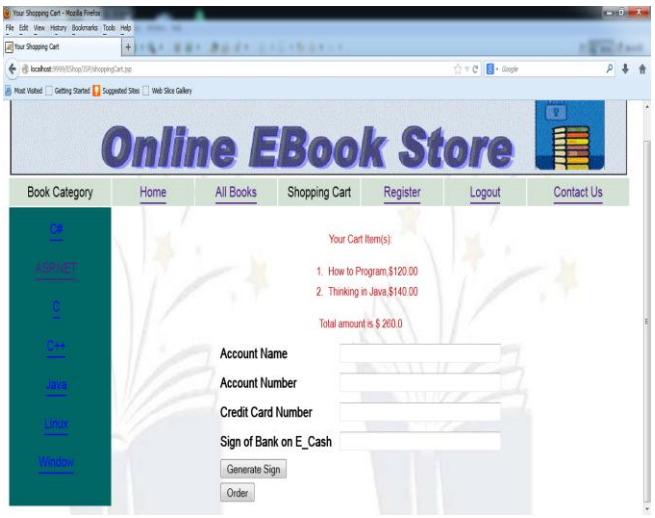

Fig. 5. Online Shopping Page.

Then the customer sends the messages (E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$) to the merchant. After receiving the above messages, the merchant checks whether or not the messages: Cost, $Account_{ES}$, $Time_A$, $Sign_t$, and $s^{ebv} \equiv (H (e_t||n_t) \bmod n_b)$ are correct. If they are correct, the merchant forwards the bank the messages (E-goods, Cost, $Account_{ES}$, e_t, n_t, v, s, $Time_A$, $Sign_t$). Then, 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. And 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 e-cash. Finally, the bank 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.

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

B. Formal Analysis of the System

Automated validation of internet security protocols and applications (AVISPA) works properly on Linux Operating System (Fig.6). AVISPA requires at least 512 MB RAM and works well with 20 MB Hard Disk. AVISPA [5] 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) [6] 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 protocol is verified by using AVISPA. Basic role is concerned with the actual activities of the each party. For this, three basic roles are played as Customer (A), 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 (A 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 protocol returns the following output

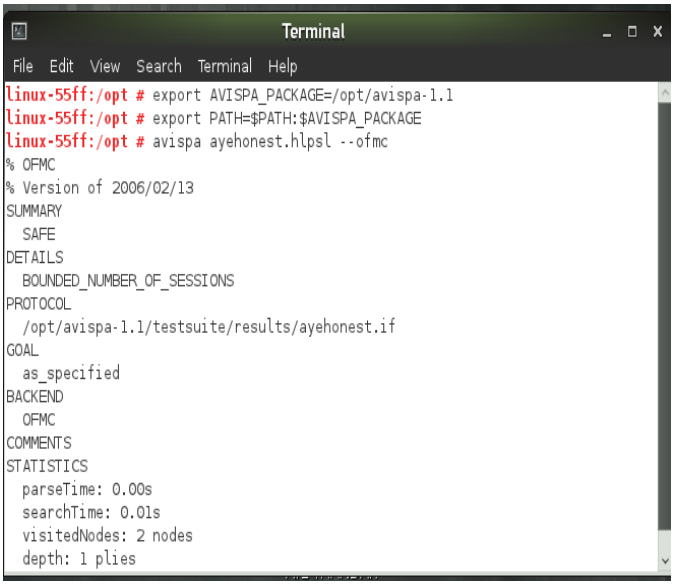

Fig.6. Analysis of proposed protocol using AVISPA.

IV. CONCLUSION

The proposed protocol has some properties. First, it gives strong fair exchange property for all players in the protocol. Second, the protocol uses the bank like as trusted third party (TTP). Third, before actually paying for the e-goods, the customer is confident by paying for the correct product. Fourth, the protocol provides anonymity for the customer. Fifth, the protocol supports non-repudiation service for customer, merchant and bank. Moreover, this paper introduces the AVISPA protocol analysis tool. AVISPA provides a powerful specification language, HLPSL, for protocols and integrates four different back-ends that perform the actual protocol analysis. Finally, the proposed protocols have been validated by using AVISPA tool that is an automated tool for the verification of security protocols.

V. REFERENCES

[1] B. Schoenmakers, “Basic Security of the E-cash 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

[2] Swapna Kodali, " The Design and Implementation of an E-commerce Site for Online Book Sales", May (2007) .

[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] M.Abe and E.Fujisaki, “How to Date Blind Signatures” ,Advances in Cryptology-ASIACRYPT'96 (LNCS 1163), pp.244-251,1996

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

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