International Journal of Research in Advent Technology, Vol.3, No.5, May 2015
E-ISSN: 2321-9637

A New Fair Exchange and Anonymous Online E-Cash Protocol for Electronic Commerce Transaction

Harshita[1], Sarvesh Tanwar[2]
Department of computer science[1, 2], Mody University Lakshmangarh (Rajasthan)
M. Tech Student[1], Assistant Professor[2]
E-mail: 2014harshita@gmail.com[1], s.tanwar1521@gmail.com[2]

Abstract -In the world of internet e-commerce plays a vital role. The transaction of e-commerce is done in the form of electronic payment. E-cash is one of the types of electronic payment methods. It can be divided into two forms on-line and offline; online e-cash suffers from anonymity issues of customer and off-line suffers from double spending of same coin. In this paper we propose an on-line e-cash protocol which ensures the anonymity of customer and also we applied our e-cash protocol in fair exchange of product using off-line TTP. Basically fair exchange of product is a main concern in electronic commerce. For this a trust factor should be maintain between customer and merchant. The main motivation of our research paper is to maintain the trust factor between customer and merchant while exchanging the digital product with secure online payment.

Index Terms: E-cash, TTP, Anonymity, Security, Transaction, Double spending

1. INTRODUCTION

As the internet technology increases use of e-commerce also increases. Gradual increase in e-commerce gives good impact to using electronic money in electronic transaction for internet users. E-cash is one of the types of electronic payment method. Electronic cash can be divided into two from of categories: on-line and off-line. In on-line electronic cash system bank should be present at time of merchant transaction whenever customer pay coin to merchant, bank should validate it and deposit into merchant account after checking double spending of coin. In offline e-cash system bank should not be present at time of online transaction validation of coin is done partially by merchant only. There are some basic features of electronic cash which are given as below:

(1) **Anonymity:** It means identities of buyer should be hidden from seller at time of electronic transaction. Anonymity of e-cash should also be hidden from bank; it means that while customer sends signing request to bank the bank will not get to know what actually signer wants to bank sign. This can be achieved by using chaum's blind signature concept which was proposed in 1982.

(2) **Double Spending:** Double Spending occurs when a person uses the same copies of their e-coins more than one time. This problem solved by creating a new coin and double coin database at bank's side.

(3) **Acceptability:** E-cash is universally acceptable as a payment form, for any amount of transaction.

(4) **Efficiency:** Payment system should be efficient and also should have effective payment processing.

(5) **Security:** While developing an electronic cash system privacy of user, merchant and bank should be kept in mind.

Basically in electronic cash protocol the main problem is to prevent anonymity and double spending of customer and coins respectively. If we maintain the anonymity of user and coin than traceability becomes also an issue. There are many other security considerations which should keeps in mind while implementing e-cash protocol like non-repudiation, confidentiality, authentication and availability.

The main objective of our research work is to how we use an on-line e-cash protocol in e-commerce for fair exchange of products.

In this paper, we have built an on-line e-cash protocol for electronic commerce transaction. We have use offline TTP model for our e-commerce protocol. There are many previous works did on e-cash and electronic commerce protocol for fair exchange of products and secure payment online. Many researchers use online TTP, offline TTP and inline TTP for fair exchange of products. In our protocol we propose a new online electronic cash scheme for fair exchange of product in electronic commerce and use the concept of compatible keys which is given by [4]. Propose work is describes in next section of this paper

2. PROPOSE WORK

Our work is to use on-line e-cash protocol in fair exchange of products in e-commerce using offline TTP model. We have proposes on-line e-cash protocol with some phases and slightly modified electronic commerce protocol of [4].

2.1) On-line e-cash protocol

The main difference between on-line e-cash and off-line protocol is that the bank should be present on-line at withdrawal and deposit phase of transaction or we can say that the bank should be present at all stages of transaction Our protocol contains four main servers bank, customer and merchant and TTP. There are five phases in our protocol initial setup, registration, issuing of e-cash, withdrawal, payment and deposit. In online e-cash protocol payment and deposit are comes in same setup where in offline they consider as different phases.

2.2) Initial Set up

First of all bank, merchant and TTP gets their public key certificate from the central authority CA and publish their public key online and keeps their private key separately with them. Merchant also gets the product certificate from TTP. For this he registers the product with TTP whereas TTP calculates different product key for each product i.e. public and private key pair using RSA algorithm and encrypt the product with public key which we denoted by (e, n1) and publish on website. Now the product which seen by customer is in encrypted form (m, e, n) along with its cost and name. Merchant also calculates the compatible public private key pair for that product which is represented as (e, n2) and (d2, n2) and keeps keys with him. In our protocol use of compatible keys is to make trust between customer and merchant. The steps of making compatible keys are given as below:

(1) Let m be the product which is encrypted with the public key (e, n1) generated by TTP whereas TTP keeps (d1, n1) with itself.
(2) Now, merchant also have (e, n1) and make compatible keys (e, n2) and (d2, n2) with itself. In compatible keys e should be the same where (n, d) key pairs are different.
(3) For proof of theorem let us take a message m encrypted with (m, e, n1) and same message encrypted with (m, e, n1*n2).
(4) (m, e, n1)=(m, e, n1*n2)
(5) $m^e mod\ n1 = m^e mod\ n1 * n2$
(6) $m^e mod\ n1 = m^e mod\ n1 (mod\ n1)$
(7) $m^e mod\ n1 = m^e mod\ n1$
(8) $m = m$, LHS = RHS hence proved

2.3) Registration

In this phase customer opens their account in his own respective e-cash bank. Suppose bank gives customer a unique-id, password and pin (note that password should be one time use only). Now the customer goes to online site of bank and re-registers him/her self to bank and calculates the security key which would act as symmetric key between bank and customer.

$$E_{bank\_e}(user_{id} + time_{customer} + seceret\_key + \mathrm{E}_{password\_cust}(H(msg))) \text{ ----- Eq. (1)}$$

In above eq. (1) customer sends his/her own id along with secret key and timestamp to bank which is sign by customer temporary password and encrypted with bank public key. Here H (msg) represents hash on the message which is sign by customer temporary password. After receiving eq. (1) bank first decrypts the message and check the sign of customer and sees whether message has been altered during transmission or not and saves security key in his database along with user-id and issues new password to customer. Here we are using time stamp for checking message has been altered or not.

2.4) Issuing e-cash

In our e-cash system we have used system online rather than using e-wallet concept. For issuing e-cash first of all customer login into the online e-cash bank and apply for e-cash than bank gives the e-cash by generating some random serial coin like "19115040612345". In this representation of coin 191 is bank number, 150406 (yymmdd) is coin creation date and 12345 is random serial number which issued to customer and hash of e-cash with its expiry date is stored in new coin database of bank.

2.5) Online electronic commerce protocol using online e-cash protocol

Now after issuing e-cash customer visit to the merchant website and chooses the product of his/her own choice which is already encrypted by TTP product public key (m, e, n1).Customer and merchant now proceed with withdrawal, payment and deposit protocol.

2.5.1) Withdrawal Phase

Customer request to bank for signing coin for this customer sends hash of e-cash, amount, timestamp, pin, security key, id of customer and blinded e-cash.

$$\mathrm{E}_{bank\_e}(user_{id} + \text{timestamp} + HasH\ E - casH + blinded\ ecas\square + amount + pin + \mathrm{E}_{security\_key}(H(msg))) \quad \text{---Eq. (2)}$$

In eq. (2) customer send hash of e-cash for checking value whether exists in new coin database or not, Security key acts as symmetric key and uses for signing the message, timestamp for avoiding replay attack, amount means how much account should be deduct form customer account, user-id gives information about from which user account amount will be debited and pin for authentication that the user is authorize or not. Blind e-cash means sending the message in an envelope so that signer of message not

gets knows what he is signing. Here we are signing coin blindly to hide the actual serial number from bank. For blinding e-cash we have to choose any random number r in Z_n^* public key group of bank (e, n) and multiply it to the e-cash. Now the blinded e-cash in eq. (2) becomes

Blinded e-cash= r^{eb}(e-cash XOR amount) mod n ----Eq. (3)

After receiving eq. (2) bank first decrypts the message by its own private key and checks the sign of user is correct or not and checks hash of e-cash present in new coin database or not after checking validity of e-cash bank will deduct appropriate amount form customer account and signs the blinded e-cash coin by its own private key and sends message bank to customer.

$E_{security\_key}(user_{id} + \text{time}\,stamp +$

$signed - ecas\square + E_{bank\_d}(H(msg)))$ ----Eq. (4)

After receiving eq. 4 customers first decrypts message by using its own symmetric security key then checks bank signature on message and gets the signed e-cash by removing its blinding factor. Now our representation of e-cash coin after removing blinding factor will be ((e-cash xor amount) ^{db}mod n, e-cash, amount) which will be send to the merchant.

2.5.2) Payment and deposit protocol

We have used the on-line e-commerce protocol using e-cash by [4]. The protocol contains the four messages should be exchanged between customer and merchant while online transaction. We have slightly modified their protocol. The steps of new protocol are given as below:

Step 1: Customer first creates a temporary public, private key pair for a transaction and creates a symmetric key for encrypting e-cash. The reason behind creating a temporary public and private key pair is to hide the actual identity of customer from merchant and remain anonymous during transmission.

$E_{merchant\_e}(Eseceret_{key(encrypted\ ecasH)} +$

$\text{timestamp} + (m, e, n1) + temp_e + temp_n +$

$product_{cost} + E_{cust\_temp\_d}(H(msg)))$ ----------Eq. (5)

Now customer sends eq. 5 to merchant which contains:

(1) (m, e, n1) encrypted product details

(2) Encrypted e-cash = E_{bank_e} (e-cash, e-cash xor amount ^{db}mod n) here we are encrypting e-cash with bank public key because merchant will not be able to alter coin serial number before depositing to bank, this attack was not prevented in [4].

(3) Timestamp for avoiding replay attack.

(4) temp_e, temp_n are temporary public key pair of customer.

(5) $E_{cust\_temp\_d}(H(msg))$ Signed message using customer temporary private key pair.

Here H is represents secure hash function which is used for integrity.

Step 2: after receiving eq. 5 merchant decrypts message using his own private key , checks the sign of customer and checks the $product_{cost}$. If sign of customer is correct than merchant calculates encryption on product using compatible public key of respective product (m, e, n1*n2) and send eq. 6 to customer otherwise merchant sends ABORT transaction to customer and transaction is terminated.

$E_{cust\_temp\_e}((m, e, n1 * n2) + \text{timestamp} +$

$\text{sign}_{merchant}(m, e, n1 *$

$n2+Emerchant\_dHmsg)$—Eq. (6)

Step 3: After receiving eq. 6 customer first decrypts the message and checks the sign on message than checks whether (m, e, n1) is equal to (m, e, n1*n2). If they are equal than customer will get to know that merchant is indeed genuine and sends symmetric key for encrypted e-cash in form of eq. 7 otherwise customer sends ABORT message to merchant and terminate the transaction.

$E_{merchant\_e}((m, e, n1 * n2) + \text{time}\,stamp +$

$\text{sign}_{merchant}(m, e, n1 *$

$n2+seceret\_key\_cas\square + Ecust\_temp\_dHmsg)$ ----- equation (7)

Step 4: after receiving eq. 7 merchant checks whether $(m, e, n1 * n2)$ and sign of merchant are correct or not, if the product is correct than merchant decrypts the encrypted e-cash from eq. 5 and sends it to bank for credit respective money of product to its own bank account. Note that e-cash is encrypted with bank's public key so that merchant cannot be able to check the actual message of e-cash.

$E_{bank_e}(encrypted\ ecasH + cost\_of\_product +$

$\text{timestamp} + account_{no} + transcatio\_id +$

$E_{merchant-d}(H(msg)))$---Eq. (8)

Now merchant sends eq. 8 to bank which includes encrypted e-cash, account no of merchant, transaction id and sign of merchant.

Step 5: After receiving eq. (8) bank decrypts the message and check sign of message on it. If they are correct than bank decrypt encrypted e-cash and gets (e-cash, (e-cash xor amount) ^{db}mod n). Now banks checks cost of product is equals to the e-cash amount which was signed by bank, if they are correct than bank checks serial number of e-cash coins whether they are in present in double spending database or not. If the coins are not present in double spending database than bank credits merchant account with appropriate amount and saves transaction id and sends message to merchant that you have completed your transaction with this transaction id.

Step 6: After receiving ok message from bank merchant now sends product decryption key, transaction –id and date slip to client.

$$E_{cust_{temp_d}}(product_{decryption_{key}} + trans\_id + \text{timestamp} + E_{merchant-d}(H(msg)))\text{---Eq. (9)}$$

Step 7: After receiving eq. 9 customers will decrypt the actual product by using $product_{decryption_{key}}$ and gets the license of goods. If in case customer do not receive correct product key from merchant in definite time than customer contacts TTP for this and sends the necessary details to take some action against merchant and TTP sends the correct decryption key (d1, n1) to customer.

3. LIFE CYCLE OF KEY IN PROTOCOL

In our protocol we are using symmetric key in place of asymmetric key pair between client and bank. Security key will behave as symmetric key for customer and bank, for security of system we applied more security to refresh the security key after every 3 months by customer.

Customer secret key

Customer generate secret key	expires after every 3 Months	customer again create keys

Customer temporary private-public key

In transaction customer generated temporary symmetric key pair as well as asymmetric key pair. Asymmetric key pair used for hiding the actual identity of customer from merchant whereas symmetric use to encrypt the e-cash to achieve the honesty of merchant. Life cycles of these keys are given as below:

Creates start of transaction	send to merchant
Expires after a transaction	

E cash Shared key

In eq. 5 customers creates a secret key to encrypt the e-cash to ensure the honesty of merchant. This shared symmetric key should be generated for every transaction.

Creates start of transaction	encrypt e-cash
Expires after a transaction	

4. CONCLUSION AND FUTURE WORK

In this paper, we have described our research work on fair exchange of digital product in electronic commerce using on-line electronic cash protocol. The future work of protocol is to check its efficiency by using various software tools and compare it with other protocol.

REFERENCES

1) A.T. Swe and K.K.K. Kyaw. "Improved E-cash Protocol." *International Journal of scientific and technology research.* 2.4 2013.
2) A.T. Swe and K.K.K. Kyaw, "Formal Analysis of Secure E-cash Transaction Protocol."*International conference on Advances in engineering and Technology.* ICAET.2014.
3) Hani M. AL-Matari, Abdalnaseer A. Hajer and Nidal F. Shilbayeh, *"Anonymous and Non-Repudiation E-Cash Scheme with Partially Blind Signature"*, Journal of Computing, Volume 3,Issue2, February 2011
4) Ray, Indrajit, Indrakshi Ray, and Narasimhamurthi Natarajan. "An anonymous and failure resilient fair-exchange e-commerce protocol." *Decision Support Systems* 39.3 (2005): 267-292.
5) *Make E-cash with Non-Repudiation and Anonymity"* , Proceedings of the International Conference on Information Technology: Coding and Computing, 2004
6) M.Abe and E.Fujisaki, ―*How to Date Blind Signatures"* ,Advances in Cryptology-*ASIACRYPT'96* (LNCS 1163),pp.244-251,1996.
7) M. Blum, How to exchange (secret) keys, ACM Transactions on Computer Systems 1 (1983) 175– 193.
8) B. Cox, J.D. Tygar, M. Sirbu, NetBill security and transaction protocol, in Proceedings of the 1st USENIX Workshop in Electronic Commerce, New York, NY, USENIX Association, California, 1995 (July), pp. 77– 88.
9) R.H. Deng, L. Gong, A.A. Lazar, W. Wang, Practical protocols for certified electronic mail, Journal of Network and Systems Management 4 (3) (1996).
10) M.K. Franklin, M.K. Reiter, Fair exchange with a semi-trusted third party, Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich,

Switzerland, Association for Computing Machinery, New York, 1997 (April), pp. 1–6.

11) G.R. Ganger, et al, Survivable storage systems, Proceedings of the DARPA Information Survivability Conference and Exposition, Anaheim, CA, vol. 2, IEEE Computer Society, California, 2001 (April), pp. 184–195.
12) B. Kaliski, M. Robshaw, The secure use of RSA, CryptoBytes 1 (3) (1995) 7 – 13.
13) S. Ketchpel, Transaction protection for information buyers and sellers, Proceedings of the Dartmouth Institute for Advanced Graduate Studies: Electronic Publishing and the Information Superhighway, Dartmouth College, New Hampshire, 1995.
14) J. Zhou, D. Gollmann, A fair non-repudiation protocol, Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, IEEE Computer Society, California, 1996 (May), pp. 55– 61.