Some general traits of the e-cash system and a review of a compact e-cash scheme with practical and complete tracing

Ema Stamenkovska, Vesna Dimitrova and Aleksandra Popovska-Mitrovikj
Faculty of Computer Science and Engineering,
University "Ss Cyril and Methodius" - Skopje, P.O. Box 393, R. of Macedonia
emails: e.stamenkovska92@gmail.com, vesna.dimitrova@finki.ukim.mk, aleksandra.popovska.mitrovikj@finki.ukim.mk

Abstract *– The e-cash methodology has its advantages compared to other payment systems and it has brought big changes to the way business is being conducted. Money becomes an intangible item and travels electronically across the world in a widely open network that might expose it to risks. This means that secure end-to-end connections are needed and many different cryptographic algorithms are used to achieve it. In this paper we will go through the main metrics that characterize them and the main properties of the e-cash system. Finally, a review of a compact e-cash scheme with practical and complete tracing will be given.*

Keywords *– COMPACT E-CASH, CIA TRIAD, E-CASH PROPERTIES, PAYMENT SCHEME*

1. Introduction

New methodology and innovation need to be crafted to improve the current payment system. One such innovation in which research and development is being invested is the e-cash methodology. It has the potential to bring big changes to the way business is being conducted. E-cash is a convenient replacement for traditional coins and banknotes, which are not viable for e-commerce [1].

E-cash transactions, over wired or wireless public networks demand secure end-to-end connections, and they must assure confidentiality (measures taken to guarantee that users' data is protected from unauthorized access), integrity (safeguarding the accuracy of data as it moves through users' workflows) and availability (seamless, uninterrupted access to users). This concept is known as the CIA triad [2, 3]. We can see a colorful representation of its symbol in Figure 1.

Fig.1 *CIA triad* [*3*]

A wide variety of electronic payment systems exists and most of them are incompatible with each other. Known electronic payment systems are:

- Electronic cash system
- Electronic cheque system
- Smart card-based electronic payment system
- Online credit card payment system

Among these models, the use of e-cash is more secure than the others, more private and has less marginal transaction costs [4, 5]. The e-cash model is a token-based payment scheme, and it does not require transactions to be recorded, since the token itself allows straightforward verification by the merchant [1].

In Section 2 we will describe the main metrics that characterize cryptographic algorithms used to secure end-to-end connections. Then, in Section 3 we will present some of the important properties for e-cash implementation. The review of a compact implementation of the e-cash methodology with practical and complete tracing will be given in Section 4.

2. Main metrics that characterize cryptographic algorithms

In Section 1 we mentioned that the e-cash methodology demands secure end-to-end connections. For the security of communication channels, more algorithms are used, depending on the needed action and result. They are different in terms of:

A. Size

Memory capacity required for implementation is determined by the size of the plaintext, the number of operations in the algorithm, the key size that is used etc. The used memory impacts the cost of the system, so it is desirable the required memory to be as small as possible, but still the algorithm can process plaintext smoothly and quickly.

B. Time

Time required by the algorithm to complete the encryption and decryption process should be as short as possible to make the system responsive and fast. The speed of the processor and the complexity of the algorithm will affect the performance of the algorithm.

C. Throughput

Throughput of the encryption and decryption algorithm is obtained by dividing the plaintext by the total time. Greater throughput means better performance [2, 6].

3. Properties of e-cash system

Payment via coins and banknotes is transferable, acceptable, dividable, untraceable, and anonymous. To be able to replace coins and banknotes, e-cash should be as good as coins and banknotes according to the features [1]. Some of the important properties for e-cash implementation are:

A. Security

The originality of the e-cash message being transferred among customers, merchants and banks needs to be secured. Any unauthorized intercepting or changing the content of the message is not allowed. The e-cash system must possess qualities such as integrity, nonrepudiation (cannot be successfully disputed) and ability to authenticate. All involved parties must know whom they are dealing with before engaging or committing any transaction. Integrity is achieved when the message sent by consumers, merchants and banks is intact when it reaches the respective recipients. Once the e-cash service provides proof of the integrity, the origin of data and an authentication that can be said to be genuine with high confidence, consumers, merchants or banks could no longer deny the transaction. Then nonrepudiation is achieved [1, 7]. The coins must not be forgeable and no one except the bank should be able to generate valid e-cash [8].

B. Privacy

Privacy is needed to protect consumers' privacy from being monitored for the purpose of financial surveillance. However, anonymity does impose certain danger such as counterfeiting, money laundering and blackmailing. Consumers should be aware that the more anonymity is offered the less security can be achieved with the e-cash system. This property of the e-cash system implies that no one should be able to identify the customer who uses e-cash for a transaction, link or trace his/her behaviors [1, 8].

C. Portability

The e-cash system should have environmental independence, similar to the conventional money system which does not depend on physical location. It should be transferable via network through portable storage devices [1].

D. Transferability

The transferability feature allows consumers to transfer e-cash from one person to another without a need to refer to the bank, which means that the bank is not participating in the payment process. Similar to the conventional cash where coins or banknotes can be easily transferred from one person to another, e-cash should be able to do the same [1]. Transferability of e-cash coins is a missing feature in most e-cash systems proposed so far, whether they are online or offline. The lifetime of a coin in transferable e-cash systems is equal to the lifetime of the transaction that the coin is involved in. There is no need to issue new coins for each transaction. In online systems, a great disadvantage is the possibility of creating a bottleneck since each coin in a transaction needs to be verified by a central server [9]. The transferability feature imposes the problem when double spending cannot be tracked, since the e-cash coins may have been transferred to different entities too many times. Therefore, it is important the owner of the e-cash coins to be identified, if he/she spends the same coin more than once [1, 8].

E. Divisibility

If it is needed, the e-cash can be divided into small denominations to allow small value transaction (this is known as micropayment) [1]. Micropayments are used for transaction of small sum of money for online content, download, a service, or Web-based content [10]. The challenge for divisible system is to be able to divide the e-cash value to small values where the total of the small e-cash values is equal to the original value. Many systems are being developed to solve this problem, such as the ones proposed by Eng and Okamoto's scheme, Okamoto's scheme, Okamoto and Ohta's scheme etc [1].

4. Phases of a compact e-cash scheme with practical and complete tracing

In this section we will give a review of a compact implementation of the e-cash methodology where different tracings should be provided by the appropriate entity, and the provided tracings should meet the demand for real-world applications [8]. In Section 1 we mentioned that the e-cash methodology can achieve anonymity in its implementation, but for higher security it can also be implemented as traceable [1]. A compact e-cash scheme is the one that efficiently minimizes the cost of the protocols involving the bank and minimizes the storage space [11].

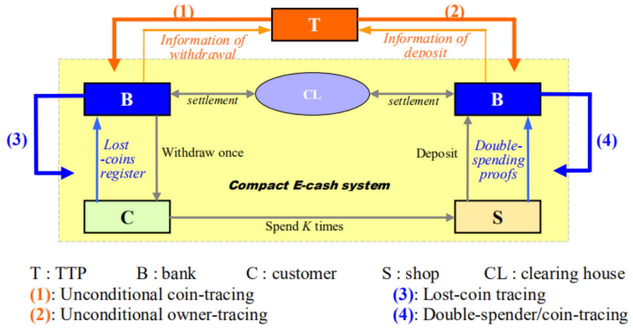

Fig. 2 *Compact e-cash system with practical and complete tracing* [8]

The scheme given in Fig. 2 consists of withdrawal protocol, payment protocol, deposit protocol, loss register protocol and practical complete tracing [8].

A. Withdrawal Protocol

This protocol starts with the customer and the bank generating wallet parameters (e_1, e_2, x) (digital equivalent of a physical wallet), which are used to generate 2^l coins, and the bank signs them using the anonymity - enhanced CL (Camenisch-Lysyanskaya) signature known as a signature with efficient protocols. These coins are stored in O(l) bits [8].

The security of this signature relies on Strong RSA assumption, which means that if modulus n and an element $u \in Z_n^*$ are known it is hard to compute values $e > 1$ and $v \in Z_n^*$ such that $v^e \equiv u \bmod n$ [12].

Then to achieve loss tracing and unconditional tracing, the customer provides two ElGamal encryptions. ElGamal$_{\mathrm{PKc}}$(g^{e_2}) encrypts g^{e_2} using the Customer's (C) public key PK$_\mathrm{C}$ (for loss tracing) and ElGamal$_{\mathrm{PKT}}$(g^{e_2}) encrypts g^{e_2} using Trusted Third Party's (T) public key PK$_\mathrm{T}$ (for unconditional coin tracing).

Here g should satisfy two assumptions:

- The Decisional Diffie–Hellman (DDH) assumption (hypothesis that a particular problem cannot be solved efficiently in polynomial time), which means that if $G = \langle g \rangle$ is a cyclic group generated by g of order $u = \#G$ then for given (g, g^x, g^y, g^z) $\in G^4$, it is hard to decide whether g^z and g^{xy} are equal.
- The Decisional Diffie–Hellman Inversion (DDHI) assumption, which means that if $G = \langle \mathrm{g} \rangle$ is a cyclic group generated by g then for given elements (g, g^x,..., $g^{(x^q)}) \in (G^*)^{q+1}$, it is still hard to decide whether $g^{1/x}$ and a random element in G are equal.

If the message space is restricted to G the system is semantically secure under DDH and DDHI [8, 13].

Finally, C (costumer) provides a knowledge proof to validate that the encryption is generated correctly [8].

In [14] the authors consider schemes for signatures of knowledge that allow one to issue signatures on behalf of any NP (nondeterministic polynomial time) statement, which can be interpreted as follows: "A person in possession of a witness w to the statement $x \in L$ has signed message m". The game Sudoku is an example of NP (quickly checkable) but is not P (quickly solvable).

B. Payment Protocol

When the client wants to spend coins from his/her wallet, he/she performs the payment protocol with the shop.

To prove the validity of the coins, customer (C) provides the zero-knowledge proof of (e_1, e_2, x) to validate that the coin spent in this protocol is from a signed wallet [8].

Zero-knowledge proof is useful especially if the customer does not want to share data about himself/herself with the shop (S) [12].

It characterizes with interaction (between the prover, in this case C and the verifier S), hidden randomization (the group elements are random, but the reference string is not, since it contains a certain structure that is distinguishable from randomness) and computational difficulty [15].

The main difference between a zero-knowledge proof and a proof of knowledge is that when a zero-knowledge proof is used, the prover attempts to convince the verifier that something is true without revealing any additional information, and when a proof of knowledge is used, the prover attempts to convince the verifier that it knows some secret information [16].

For achieving loss tracing and unconditional tracing (both of these are optional modules), the customer computes $T_2 = g^{H(J \parallel r)e_2}$ (mod n_T) with the random input r, where n_T is RSA modulus and T has its factor knowledge, integer $J \in [0, 2^l\text{-}1]$. The customer also provides the zero-knowledge proof to validate that T_2 is generated correctly [8].

To prevent double spending (the same digital token to be spent more than once), each coin is assigned with face value (serial number). The customer provides the serial number of the coin, i.e., Θ = PRF(e_2, J), where PRF is a pseudorandom function. Pseudorandom function is a deterministic function with the payee's secret key e_2 and the public input J that is indistinguishable from a truly random function of the input (the algorithm cannot tell whether the function is not truly random) [8, 17].

The serial number records the spent coins for the bank and the integer $J \in [0, 2^l\text{-}1]$ records the spent coins for the customer. The customer provides the zero-knowledge proof to verify that Θ is generated correctly [8].

We can notice that in the payment protocol the customer provides zero-knowledge proof for proving coin validity, achieving loss tracing and unconditional tracing and double-spending prevention.

For tracing double-spending without the Trusted Third Party, the customer constructs a special knowledge proof of e_1, i.e., PK_Θ (e_1), related to the serial number. Its special property is that if PK_Θ (e_1) is showing with the same serial number (Θ) twice, the knowledge of the parameters (e_1, e_2) is leaked [8].

The Trusted Third Party could be a credit card company, stakeholder, escrow agent, legal adjudication or arbitration of disputes or the use of a reputation system to build trust by allowing parties to gain some understanding of the prior behaviour of the other. It threatens anonymity of honest users because in underground economy it is difficult to obtain trust [18].

C. Deposit Protocol

The shop sends to the bank the information received from the customer in the payment protocol. The bank verifies it as the shop does in payment protocol (by zero-knowledge proof) and makes sure that the coin has not been delivered by the shop before (there is no other vector (Θ, J, C) in the bank's database). Then, the transfer is made [8]. Usually, there is a fixed time period after which the shop sends to the bank all the payment transcripts made in that period [19].

D. Loss Register Protocol

To registrate for tracing his/her lost coins in case they are lost from the database, the customer can send to the bank the information of the remaining coins in his/her e-wallet (LR_x), so that they cannot be spent by others. In this protocol only LR_x is shown, so that it does not affect the customer's anonymity and is not shown how he/she spent the coins when he/she spent them [8].

The bank can only refuse to perform loss tracing if it provides a proof that the coins were spent by the customer before the information is published.

The bank sends the customer $\mathrm{ElGamal}_{\mathrm{PKc}}(g^{e_2})$ and the related zero-knowledge proof of it, so that the customer can verify that the $\mathrm{ElGamal}_{\mathrm{PKc}}(g^{e_2})$ was generated by himself in the withdrawal protocol explained above and then uses his private key to decrypt $\mathrm{ElGamal}_{\mathrm{PKc}}(g^{e_2})$ to get g^{e_2} [8].

E. Practical Complete Tracing Protocol

The following two kinds of tracing are useful when crimes happen.

Unconditional coin-tracing. Getting the information of withdrawal from bank, the Trusted Third Party uses its private key to decrypt $\mathrm{ElGamal}_{\mathrm{PKT}}(g^{e_2})$ and publishes g^{e_2}. The parameter $T_2 = g^{H(J \parallel r)e_2}$ (mod n_T) will be provided to the shop. Actually, the parameters J and r will be shown, so T_2 can be identified if the Trusted Third Party decrypts $\mathrm{ElGamal}_{\mathrm{PKT}}(g^{e_2})$. Then all coins can be traced [8]. This kind of tracing happens before the purchase.

Unconditional owner-tracing. Getting the information of deposit from a shop, the Trusted Third Party uses the factor knowledge (category of authentication credentials because transactions leave digital footprint) of n_T to compute the inverse of $H(J \parallel r)$ and to obtain g^{e_2} from T_2. Then Trusted Third Party can identify the owner (customer) according to withdrawal database [8]. This kind of tracing occurs after the purchase (payment) is made, and it allows the authorities to prevent money laundering and identify customers that made an illegal purchase [20].

Double-spender-tracing. If the customer double-spends a coin, he has to show the same serial number Θ = PRF(e_2, J) more than once, so it can be found out by the bank. As mentioned earlier, if the special knowledge proof PK_Θ (e_1), with the same Θ is shown twice, (e_1, e_2) will be leaked. Since e_1 is used as tracing information of customer in withdrawal protocol, the double-spender can be traced [8].

Double-spender's coin-tracing. With the leaked e_2, each serial number Θ = PRF(e_2, J) is computed and published for $J \in [0, 2^l\text{-}1]$, so the coins from double-spender cannot be spent anymore [8]. It is desirable to have coin traceability of double-spender such that all coins of the cheating user can be traced. It is even more desirable to implement a module where when a user is a double-spender, all of his other spending can be properly identified [21].

After the loss register, the bank publishes g^{e_2}. In the payment protocol, $T_2 = g^{H(J \parallel r)e_2}$ (mod n_T) is provided to the shop with J and r. Then the shop can identify T_2 and the lost coins cannot be spent by others [8].

This scheme has recoverability, so when the lost e-cash is published and found, the amount will be returned to the rightful owner. In fact the customer will receive a refund from the bank [19].

5. Conclusion

Different algorithms are used during different stages of the compact e-cash scheme and for different purposes and properties that should be achieved for successful e-cash implementation. Each of them has its own strengths and weaknesses and this scheme combines them very efficiently. RSA and ElGamal algorithms implement a public key cryptosystem. RSA algorithm has shorter encryption and decryption time than ElGamal, but ElGamal algorithm is more secure due to the complicated calculation for solving discrete logarithms. RSA is an algorithm that maintains data confidentiality at the time of authentication and ElGamal has implemented Diffie-Hellman key distribution scheme to generate the public key for encryption and decryption processes. Zero-knowledge proof verifies the required signature on several messages, without giving away the signature (or additional sub-information on the messages) but requires greater computation which can be more expensive. Here, we consider a particular knowledge proof, where if it is used honestly, it keeps perfect zero-knowledge property, but if it is used dishonestly, it leaks the information of proven knowledge.

Acknowledgment

This research was partially supported by Faculty of Computer Science and Engineering at Ss. Cyril and Methodius University in Skopje.

6. References

1. R. Razali, *The overview of E-cash: Implementation and security issues* (GSEC, 2002)

2. M. N. A. Wahid, A. Ali, B. Esparham, M. Marwan, Journal Comp. Scien. App. and Inf. Tech., *A comparison of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish for guessing attacks prevention* 1-7 (2018)

3. https://spanning.com/blog/cia-triad-best-practices-securing-your-org/

4. P. Aigbe, J. Akpojaro, Intern. Jour. of comp. app., *Analysis of security issues in electronic payment systems* (2014)

5. W. J. Tsaur, J. H. Tsao, Y. H. Tsao, *An Efficient and Secure ECC-based Partially Blind Signature Scheme with Multiple Banks Issuing E-cash Payment Applications, Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government* (WorldComp, 2018)

6. A. P. U. Siahaan, Elviwani, B. Oktaviana, *Comparative analysis of rsa and elgamal cryptographic public-key algorithms* (2018)

7. https://en.wikipedia.org/wiki/Non-repudiation

8. B. Lian, G. Chen, J. Cui, D. He, *Compact E-Cash with Practical and Complete Tracing, KSII Transactions on Internet & Information Systems* (2019)

9. R. S. Anand, C. E. Veni Madhavan, *An online, transferable e-cash payment system, International Conference on Cryptology in India* (Springer, Berlin, Heidelberg, 2000)

10.https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11 /MicropaymentsAndTheNet/history.html

11. B. Lian, G. Chen, J. Cui, M. Ma, *Compact E-Cash with Efficient Coin-Tracing, IEEE Transactions on Dependable and Secure Computing* (2018)

12. J. Camenisch, A. Lysyanskaya, *A signature scheme with efficient protocols, International Conference on Security in Communication Networks* (Springer, Berlin, Heidelberg, 2002)

13. J. P. Buhler, ed. *Algorithmic Number Theory: Third International Symposium, ANTS-III, Portland, Orgeon, USA, June 21-25, 1998, Proceedings*. Vol. **1423**. (Springer Science & Business Media, 1998)

14. M. Chase, A. Lysyanskaya, *On signatures of knowledge, Annual International Cryptology Conference* (Springer, Berlin, Heidelberg, 2006)

15. M. Blum, P. Feldman, S. Micali, *Non-interactive zero-knowledge and its applications, Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali* 329-349 (2019)

16. https://www.cs.jhu.edu/~susan/600.641/scribes/lecture10.pdf

17. https://link.springer.com/referenceworkentry/10.1007%2F0-387-23483-7_329

18. A. Asgaonkar, B. Krishnamachari, *Solving the buyer and seller’s dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator, 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (IEEE, 2019)

19. J. K. Liu, P. P. Tsang, D. S. Wong, *Recoverable and untraceable e-cash, European Public Key Infrastructure Workshop* (Springer, Berlin, Heidelberg, 2005)

20. G. Davida, J. Frankel, Y. Tsiounis, M. Yung, *Anonymity control in e-cash systems, International Conference on Financial Cryptography* (Springer, Berlin, Heidelberg, 1997)

21. M. H. Au, Q. Vu, W. Susilo, Y. Mi, *Compact e-cash from bounded accumulator, Cryptographers’ Track at the RSA Conference* (Springer, Berlin, Heidelberg, 2007)