INFORMATICA, 2019, Vol. 30, No. 2, 327–348

DOI: http://dx.doi.org/10.15388/Informatica.2019.208

Security, Trustworthiness and Effectivity Analysis of an Offline E-Cash System with Observers

Jonas MULERAVICIUS, Inga TIMOFEJEVA,
Aleksejus MIHALKOVICH *, Eligijus SAKALAUSKAS

Department of Applied Mathematics, Kaunas University of Technology,
Studentu Str. 50, LT-51368, Kaunas, Lithuania
e-mail: jonas.muleravicius@ktu.edu, inga.timofejeva@ktu.edu, aleksejus.michalkovic@ktu.lt,
eligijus.sakalauskas@ktu.lt.

Received: November 2018; accepted: March 2019

Abstract. In our previous paper we presented an offline e-cash system with observers. We have shown that the proposed system satisfies basic requirements for e-cash schemes. We also covered such security issues as chosen message attack resistance and forgery of protocols data. However, in that paper we focused more on the system itself, rather than its analysis.

Hence, here we present cryptanalysis of our system. We aim to prove that existential forgery of data is not possible due to complexity of the discrete logarithm problem. Furthermore, we perform the analysis of trustworthiness of the system using the so-called BAN logic. Also, we consider effectivity of the proposed e-cash system in observers with limited computational resources.

Key words: e-cash, BAN logic, observers, computation time.

1. Introduction

Since their discovery, e-cash systems have drawn attention of many scientific minds in cryptography. Developments in this branch of cryptography led to most famous cryptocurrency in the world – Bitcoin. Nowadays authors tend to propose transactions based on e-cash system, rather than focusing on check transactions. The following challenges for e-cash systems were pointed out by many authors (Pfitzmann and Köhntopp, 2001; Rosenberg, 2010; Eng and Okamoto, 1994; Chaum *et al.*, 1988; Chaum and Pedersen, 1992; Kreft and Adi, 2006; Muleravičius *et al.*, 2016):

1. Security against money laundering;
2. Double spending prevention;
3. Loss of e-wallet;
4. Preservation of customers' anonymity;
5. Minimization of online operations on a large database;
6. Security against e-coin forgery.

*Corresponding author.

Since e-cash is now considered a digital analogue of regular money, any proposed system of this type should satisfy the following main properties:

1. **Anonymity:** The customer using his e-cash to pay for a product must remain anonymous against the recipient of the money as well as the bank.
2. **Unreusability:** E-cash cannot be copied or double spent. This implies that the e-wallet system has to minimize the risks for forgery and/or provide ways for the identification of a dishonest user.
3. **Unforgeability:** Only authorized parties (i.e. the bank) can produce e-cash.
4. **Off-line Payment:** The payment transaction must be performed offline, i.e. no communication with the bank should be necessary during the payment protocol.
5. **Transferability:** Received e-cash can be applied for other payments among customers, regardless of whether transactions are online or offline.
6. **Divisibility:** E-cash must be divisible, i.e. the customer should be able to divide it into smaller amounts.

One of the crucial drawbacks of an e-cash system is the rapid growth of the data throughout its transfers. The latter point plays a major role in the effectiveness of the e-cash system. However, for a long time it was ignored by many proposed systems. According to Chaum and Pedersen (1992), the amount of data transferred among users through any divisible, offline and anonymous e-cash system is growing in size due to the information needed to store in order to ensure double spending prevention, divisibility and other properties.

Nevertheless, some alternative e-cash systems were proposed that managed to avoid the growth of the e-cash data (D'Amiano and Di Crescenzo, 1994; Okamoto, 1995). However, as mentioned in Chan *et al.* (1998), Tsiounis (1997), those e-schemes had other issues such as the limit of the total size of payments or lack of efficiency of e-cash protocols. In Fuchsbauer (2009) an attempt was made to construct a transferable e-cash scheme without the aforementioned data growth problem. However, it was later outlined in Waters (2005), Fuchsbauer (2009) that there was still a dramatic increase in the size of the public key.

A new direction in the development of offline e-cash systems without the data growth drawback was established when Brand first presented an e-wallet scheme using observers in (Brands, 1993). He proposed the idea of bank's trustee for the purchaser (e.g. a chip implemented in a purchaser's mobile device) which allows to perform payments without the online connection to the bank. However, the cryptographic security of Brand's e-cash system was never proven and hence the system was never initiated.

Another problem that often takes place in divisible, anonymous, offline e-cash systems is the lack of proof of the security of a complex cryptographic system. According to Rosenberg (2010), the majority of divisible e-cash systems to this day "use proofs about double-discrete logarithms and require similar sequences of primes in their setup". It was noted in Brands (1993) and Cramer and Shoup (2003), the decisional Diffie–Hellman (DDH) assumption is needed to prove the cryptographic security of a number of previously proposed protocols. This comes from the fact that the Diffie-Hellman key exchange (Diffie and Hellman, 1976) cannot be proved secure in any reasonable and standard way

just based on the computational Diffie–Hellman (CDH) assumption: the DDH assumption is required.

In Petersen and Poupard (1997) an efficient payment system with anonymity revocation and trusted third party (TTP) was presented. It was the first scheme that managed to achieve an offline prevention of all possible extortion attacks. Due to the system's scalable security and efficiency, secure realizations for an internet payment scheme as well as a highly efficient payment scheme for electronic purse applications were developed on the basis of this scheme. The system also incorporated a possible way to revoke anonymity using the collaboration of a judge and the bank if a malicious purchaser was to be detected (Stadler *et al.*, 1995). The judge could be implemented in a Purchaser's smart device. However, it should be protected to ensure Purchaser's anonymity.

This paper considers an offline, divisible, anonymous and transferable e-cash system with observers operating in the environment with TTP (the bank), which was previously presented in Sakalauskas *et al.* (2018). Detailed description of our scheme is presented in Sections 2.1–2.3. The analysed e-money system does not possess the data growth problem addressed above due to the utilization of observers. We also perform analysis of several attack scenarios, which include existential forgery of data by both parties (Purchaser and Vendor) of our system in Section 3. In Section 4 we show that both parties can trust each other. This analysis relies on the so-called BAN logic presented in Burrows *et al.* (1989) (hence it was named after the authors).

2. Offline E-Cash System with Observers

In this section we present a novel e-cash system with observers based on Schnorr identification and modified ElGamal e-signature. These cryptographic primitives are often implemented due to their provable security. Our e-cash system satisfies such e-cash system properties as divisibility, anonymity, offline payment, transferability and double-spending prevention requirements. To provide offline payment property and bypass the growth of the transferred e-cash data the implemented cryptographic bank's trustee Observer (i.e. cryptographic chip) is used.

Presented e-cash system is executed between the following parties: the Bank (**B**), the Purchaser (**P**), the Vendor (**V**) and their Observers ($\mathbf{O_P}$ and $\mathbf{O_V}$ respectively). Our e-cash system operates using withdrawal, payment and deposit protocols described in detail in Sections 2.1–2.3.

The general parameters and functions used in the proposed e-cash scheme are presented in Table 1 below:

Our e-cash system consists of Withdrawal, Payment and Deposit protocols. These protocols are presented below and are executed in order of presentation.

2.1. *Withdrawal Protocol*

Assume that the Purchaser is interested in acquiring some goods from the Vendor. Let the total price of these goods be m_i, where i denotes the number of the transaction. He

Table 1
E-cash system notations.

Parameter/functions	Description
q, p	Large prime numbers, such that $p = 2q + 1$
G	Generator of multiplicative group $\mathbf{Z}_p^*$
h_i	Value of hash function
$Sig_{ElG}^{X}(m)$	ElGamal signature function, where m and X correspond to the message to be signed on and the ElGamal private key of the signer
$Ver_{ElG}^{A}(s, m)$	ElGamal signature verification function, where m, s and X correspond to the message, signature on the message and the ElGamal public key of the signer
i	Serial number of the transaction
$PrK_P = x_P, PuK_P = \{G, A_P = G^{x_P}\}$	Purchaser's temporary private and public keys It is important to note that in the proposed scheme, the purchaser generates random temporary private and public keys for each transaction, which ensures the anonymity property of the proposed e-cash scheme
$PrK_O = x_O, PuK_O = \{G, A_O = G^{x_O}\}$	Observer's private and public keys
Id_P, Id_V	Unique identification number of the Purchaser's and Vendor's Observer chips respectively
m_i	Sum of money to be spent by the Purchaser
$\tilde{m}_i$	Actual price of the products to be bought by the purchaser
t_i	Time instance of the e-cash withdrawal
$m_i \| \| t_i$	Concatenation of the sum and the time instance
t_{w0}, t_{p0}, t_{d0}	Time instance of the last e-cash withdrawal (payment, deposit) protocol
$m_{\max}^{P}, m_{\max}^{V}$	The amount of money in the e-wallet of Purchaser and Vendor respectively
$\xi_i^{(1)}, \xi_i^{(2)}$	Random values of Z_q^* for Schnorr interactive identification protocol

initializes the purchase deal by requesting Vendor's identity indicator Id_V, which is sent to him via secure channel. We consider it the zeroth step of our scheme.

Execution of our e-cash system starts by performing the following steps of the Withdrawal protocol:

1. The Purchaser sends a request to his Observer to provide him with the desired sum m_i. He generates his temporary keys $PrK_P = x_P$ and $PuK_P = \{G, A_P = G^{x_P}\}$ and sends his public key A_P to the observer together with the desired sum, the time of request t_i and Vendor's identity indicator Id_V. Hence, the Purchaser's observer receives the following information:

$$\mathbf{P} \xrightarrow{m_i, t_i, Id_V, A_P} \mathbf{O_P}.$$

2. Observer $\mathbf{O_P}$ checks if the Purchaser possesses the desired sum and verifies if the request takes place in the current time and if time instance t_i is greater than the time instance t_{w0} of a previous request:

$$Ver(t_i > t_{w0}),$$

$$Ver\left(m_i < m_{\max}^P\right).$$

The protocol is aborted if any failures occur at this step. Purchaser receives an error message indicating the problem.

3. Observer $\mathbf{O_P}$ generates private data for the Purchaser – random values $\xi_i^{(1)}$, $\xi_i^{(2)}$, which he will later use for Shnorr identification during the Payment protocol:

$$Gen \rightarrow \xi_i^{(1)}, \xi_i^{(2)}.$$

4. Using the generator G, the Observer $\mathbf{O_P}$ computes public data for the Purchaser, which Vendor will later use during the Payment protocol to identify him:

$$w_i^{(1)} = G^{\xi_i^{(1)}}, \qquad w_i^{(2)} = G^{\xi_i^{(2)}}.$$

5. Using pre-generated public data as well as data generated at previous steps of this protocol the Observer $\mathbf{O_P}$ calculates the following public data for the Purchaser:

$$N_i^{(1)} = m_i \| t_i \| Id_V,$$
$$N_i^{(2)} = Id_P \cdot N_i^{(1)},$$
$$P_i^{(1)} = {A_P}^{N_i^{(1)}} \cdot w_i^{(1)},$$
$$P_i^{(2)} = {A_P}^{N_i^{(2)}} \cdot w_i^{(2)}.$$

During this step the Observer $\mathbf{O_P}$ also generates the following El-Gamal signatures to prevent existential forgery of data:

$$S_i^{(1)} = Sig_{ElG}^{x_O}\left(P_i^{(1)}\right),$$
$$S_i^{(2)} = Sig_{ElG}^{x_O}\left(P_i^{(2)}\right),$$
$$S_i^{(3)} = Sig_{ElG}^{x_O}\left(A_P^{N_i^{(1)}}\right),$$
$$S_i^{(4)} = Sig_{ElG}^{x_O}\left(A_P^{Id_P}\right).$$

6. Observer $\mathbf{O_P}$ renews a time instance of the last request:

$$t_{w0} \leftarrow t_i.$$

7. Observer $\mathbf{O_P}$ renews Purchaser's e-wallet balance:

$$m_{\max}^P \leftarrow m_{\max}^P - m_i.$$

8. Observer $\mathbf{O_P}$ sends all generated data and signatures during this protocol to the Purchaser thus completing the request:

$$\mathbf{O_P} \xrightarrow{\xi_i^{(1)}, \xi_i^{(2)}, w_i^{(1)}, w_i^{(2)}, N_i^{(1)}, N_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} \mathbf{P}.$$

As a result of this protocol the Purchaser can now spend the desired sum as he wishes using the data and signatures, obtained from his Observer.

2.2. *Payment Protocol*

After Withdrawal protocol has completed, the Purchaser initializes the Payment protocol. The steps of this protocol are as follows:

1. The Purchaser sends public data pre-generated by his Observer $\mathbf{O_P}$ to the Vendor together with the total price of the goods m_i and the time instance t_i of the transaction:

$$\mathbf{P} \xrightarrow{m_i \| t_i, A_P,\, A_P^{Id_P},\, w_i^{(1)}, w_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} \mathbf{V}.$$

2. The Vendor verifies if the total price of the goods m_i is correct and if the time instance t_i is greater than the time of the last purchase t_{p0}:

$$Ver(t_i > t_{p0}),$$
$$Ver(m = \tilde{m}_i).$$

   Note that the Vendor does not verify if the transaction takes place at the current time, since the Purchaser can initialize this protocol at any time after receiving data from his Observer. The protocol is aborted if any failures occur at this step. The Purchaser receives an error message indicating the problem.
3. The Vendor verifies signatures to ensure that the received data was not forged in any way:

$$Ver_{ElG}^{A_O}\left(A_P^{Id_P}, S_i^{(4)}\right),$$
$$Ver_{ElG}^{A_O}\left({A_P}^{m_i||t_i||Id_V}, S_i^{(3)}\right),$$
$$Ver_{ElG}^{A_O}\left({A_P}^{m_i||t_i||Id_V} \cdot w_i^{(1)}, S_i^{(1)}\right),$$
$$Ver_{ElG}^{A_O}\left(\left({A_P}^{Id_P}\right)^{(m_i||t_i||Id_V)} \cdot w_i^{(2)}, S_i^{(2)}\right).$$

   The protocol is aborted if any failures occur at this step, since the Vendor found forgery of the received data. The Purchaser receives an error message indicating the problem. The Purchaser can no longer use the data of this transaction to execute any new payments.
4. The Vendor generates a random challenge h_i for the Purchaser to ensure that he is not dealing with an attacker:

$$Gen \rightarrow h_i.$$

5. The Vendor initializes Shnorr identification protocol by sending a random challenge h_i to the Purchaser:

$$\mathbf{V} \xrightarrow{h_i} \mathbf{P}.$$

6. Using his private data $\xi_i^{(1)}, \xi_i^{(2)}$ pre-generated by the Observer $\mathbf{O_P}$, the Purchaser calculates the response values $r_i^{(1)}$ and $r_i^{(2)}$ in a following way:

$$r_i^{(1)} = h_i \cdot x_P \cdot N_i^{(1)} + \xi_i^{(1)},$$
$$r_i^{(2)} = h_i \cdot x_P \cdot N_i^{(2)} + \xi_i^{(2)}.$$

He forwards the response values $r_i^{(1)}$ and $r_i^{(2)}$ to the Vendor:

$$\mathbf{P} \xrightarrow{r_i^{(1)}, r_i^{(2)}} \mathbf{V}.$$

7. Using Purchaser's public data $w_i^{(1)}$, $w_i^{(2)}$ the Vendor verifies the validity of the received response values in the following way:

$$Ver\big(G^{r_i^{(1)}} \cdot \big(A_P{}^{m_i \| t_i \| Id_V}\big)^{-h_i} = w_i^{(1)}\big),$$
$$Ver\big(G^{r_i^{(2)}} \cdot \big(\big(A_P^{Id_P}\big)^{m_i \| t_i \| Id_V}\big)^{-h_i} = w_i^{(2)}\big).$$

The protocol is aborted if any failures occur at this step. The Purchaser receives an error message indicating the identification problem. He may retry to initialize the Payment protocol if the Vendor allows this possibility. Otherwise the data of this transaction can no longer be used.

If no failures during these steps occurred, then the payment has been made. However, the Vendor now has to confirm that the payment took place.

8. The Vendor turns to his Observer for signature generation by sending the received payment sum m_i and time instance t_i, i.e. he confirms that the Purchaser has paid the sum m_i for the goods at the time t_i:

$$\mathbf{V} \xrightarrow{m_i \| t_i, S_i^{(3)}} \mathbf{O_V}.$$

9. The Vendor's Observer confirms the validity of the received data by verifying the signature $S_i^{(3)}$:

$$Ver_{ElG}^{A_O}\big(A_P^{m_i \| t_i \| Id_V}, S_i^{(3)}\big).$$

If this verification fails, then the Observer blocks the transaction for deposit, i.e. the Vendor is no longer able to deposit the sum m_i to his e-wallet.

10. The Vendor's Observer generates a signature $S_V = Sig_{ElG}^{x_O}(Id_V^{m_i \| t_i})$ and sends it to the Vendor:

$$\mathbf{O_V} \xrightarrow{Id_V^{m_i \| t_i}, S_V} \mathbf{V}.$$

11. The Vendor sends the following data to the Purchaser for verification:

$$\mathbf{V} \xrightarrow{Id_V^{m_i || t_i}, S_V} \mathbf{P}.$$

12. The Purchaser performs the following actions to ensure that he is not dealing with a Malicious Vendor:
    - Raises $Id_V^{m_i || t_i}$ to power $(m_i || t_i)^{-1}$ and compares the result to Id_V. Clearly, the results have to match.
    - He verifies time instance and signature S_V:

$$Ver_{ElG}^{A_O}\left(Id_V^{m_i || t_i}, S_V\right).$$

    If verification is successful, then the deal is made and both parties receive messages of this result. Otherwise, the deal is off and the Purchaser may turn to the Bank in electronic or physical form to restore his wallet balance. Both parties receive error messages.
13. The Vendor **V** renews a time instance of the last purchase by the Purchaser **P**:

$$t_{p0} \leftarrow t_i.$$

Upon successful completion of this protocol the Vendor has received the total price of the goods and can send them to the Purchaser in electronic or physical form. Otherwise, if any errors occurred, the culprit is reported to the Bank.

2.3. *Deposit Protocol*

To complete the execution of our e-cash system, the Vendor has to deposit the received sum m_i. Hence he initializes the following protocol:

1. The Vendor sends the data of the latest transaction, i.e. the data he has received from the Purchaser **P** to his Observer $\mathbf{O_V}$:

$$\mathbf{V} \xrightarrow{m_i || t_i, A_P,\, A_P^{Id_P},\, w_i^{(1)}, w_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} \mathbf{O_V}.$$

2. The Vendor's Observer $\mathbf{O_V}$ verifies the validity of the time instance t_i, i.e. if it is greater than the time of the last deposit t_{d0}:

$$Ver(t_i > t_{d0}).$$

Note that the Observer $\mathbf{O_V}$ does not verify if the deposit takes place at the current time, since this protocol can be executed at any time. Failure at this step results in an error message indicating, that this deposition already took place sometime before. The Deposit protocol is aborted.

3. The Vendor's Observer $\mathbf{O_V}$ verifies the received signatures to ensure that no existential forgery took place:

$$Ver_{ElG}^{A_O}\left(A_P^{Id_P}, S_i^{(4)}\right),$$
$$Ver_{ElG}^{A_O}\left({A_P}^{m_i||t_i||Id_V}, S_i^{(3)}\right),$$
$$Ver_{ElG}^{A_O}\left({A_P}^{m_i||t_i||Id_V} \cdot w_i^{(1)}, S_i^{(1)}\right),$$
$$Ver_{ElG}^{A_O}\left(\left({A_P}^{Id_P}\right)^{(m_i||t_i||Id_V)} \cdot w_i^{(2)}, S_i^{(2)}\right).$$

4. The Vendor's Observer $\mathbf{O_V}$ renews a time instance of the last deposit:

$$t_{d0} \leftarrow t_i.$$

5. The Observer $\mathbf{O_V}$ renews the Vendor's wallet balance:

$$m_{\max}^V \leftarrow m_{\max}^V + m_i.$$

3. Security Against Existential Forgery Analysis

In this section we consider security of our scheme against adaptive inside adversary, i.e. we assume that an attacker is a legitimate user (Purchaser or Vendor) of the proposed system and hence has his own mobile device with an Observer and pre-generated data as described above. We consider the following attack scenarios:

1. Attack of a Malicious Purchaser (**MP**):
   (a) Double spending, i.e. using the same data to purchase goods more than once from the Vendor;
   (b) Forgery of transaction data, i.e. faking payment sum, time instance and any data sent to the Vendor. There are two alternatives to this attack: spend less money than demanded by the Vendor (forging payment sum) or present a previous transaction as a new one (forging time instance), i.e. perform double spending by forgery.
2. Man in the Middle Attack (**MitM**):
   (a) Purchaser impersonation by faking identity Id_P, i.e. using the e-wallet of another legitimate Purchaser to acquire goods for yourself;
   (b) Vendor impersonation by faking identity Id_V, i.e. acquire and deposit money, meant for another legitimate Vendor.
3. Attack of a Malicious Vendor (**MV**):

(a) Double deposit, i.e. using the same data to increase the balance of Vendor's e-wallet more than once;
(b) Deny of payment and refusing goods shipment, i.e. keep Purchaser's money for yourself without delivering the goods;
(c) Forgery of transaction data, i.e. faking payment sum, time instance and any data received from an honest the Purchaser. There are two alternatives to this attack: deposit more money than received from the Purchaser (forging payment sum) or present a previous transaction as a new one (forging time instance), i.e. perform double deposit by forgery.

To start our analysis we first focus on the Attack of **MP** scenario, i.e. actions, which can be executed by a dishonest Purchaser to benefit from the deal with an honest Vendor.

The prevention of double spending is guaranteed by Shnorr identification, i.e. upon receiving the same transaction twice the Vendor can recover the Purchaser's identity by calculating the following expression:

$$\frac{r_i^{(2)} - r_i'^{(2)}}{r_i^{(1)} - r_i'^{(1)}} = Id_P, \tag{1}$$

where responses $r_i^{(1)}$ and $r_i^{(2)}$ were received during the first sending whereas responses $r_i'^{(1)}$ and $r_i'^{(2)}$ were received during the second sending of the same transaction. The validity of (1) is proven in (Sakalauskas *et al.*, 2018). Note that expression (1) is calculated modulo q.

To consider forgery of the data by **MP** we recall the data sent during Payment protocol to the Vendor:

$$\mathbf{P} \xrightarrow{m_i||t_i, A_P,\, A_P^{Id_P},\, w_i^{(1)}, w_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} \mathbf{V}.$$

Since this data involves signatures, in order to fake his identity the Purchaser may try to forge signatures sent during this step. However, since he is not able to generate signatures by himself (only the Purchaser's Observer can do this), forgery of any ElGamal signature requires him to deal with discrete logarithm problem (DLP) as stated in Theorem 20 of (Pointcheval and Stern, 2000) considering modified ElGamal signature scheme security against an adaptive adversary. Based on the result of Poicheval and Stern we claim the following:

Proposition 1. *If Purchaser can forge any signature during Payment protocol, then he is able to recover Purchaser Observer's private ElGamal key x_O in reasonable time.*

Hence we focus on the data signed by these signatures, i.e. we assume that the adversary aims to alter this data to obtain a valid signature on a fake data.

Formally, the security of the Purchaser's identity relies on the uniqueness of signature

$$S_i^{(4)} = Sig_{ElG}^{x_O}\left(A_P^{Id_P}\right)$$

as stated in (Pointcheval and Stern, 2000). To prove this let us consider the data signed, i.e.

$$A_P^{Id_P} = \left(G^{x_P}\right)^{Id_P} = \left(G^{Id_P}\right)^{x_P}.$$

Since G is a generator of the multiplicative group Z_p^*, the value of A_P is unique and hence we assume that it is some other generator of the same group. In this case the private key x_P is relatively prime with group characteristic p. Due to A_P being a generator of the multiplicative group, the value of $A_P^{Id_P}$ is unique as well and hence if $A_P^{Id_P} = A_P^{Id_{P'}}$, where $Id_{P'}$ is some forged identity, then $Id_P = Id_{P'}$. Furthermore, if $A_P^{Id_P} = A_{P'}^{Id_{P'}}$, then $x_{P'} \cdot Id_{P'} = x_P \cdot Id_P$, where data with index P' is fake. However, for randomly chosen values x_P, Id_P, $x_{P'}$, $Id_{P'}$, the probability

$$\text{Prob}(x_{P'} \cdot Id_{P'} = x_P \cdot Id_P)$$

is negligible if the value of characteristic p is large enough. Assume that the adversary is in possession of $A_P^{Id_P}$ and $Id_{P'}$. In order to switch Id_P to a fake identity $Id_{P'}$ the adversary has to solve the following problem:

$$A_P^{Id_P} = \left(G^{Id_{P'}}\right)^{x_{P'}} \tag{2}$$

for some unknown value of $x_{P'}$, which is a private key of the fake Purchaser's $\mathbf{P'}$ Observer. Hence we obtain the DLP as stated above.

The correctness of time instance t_i, payment sum m_i and Vendor's identity Id_V follows from the structure of $N_i^{(1)}$ and signature

$$S_i^{(3)} = Sig_{ElG}^{x_O}\left(A_P^{N_i^{(1)}}\right).$$

Analogously the DLP to be solved in case of successful forgery is as follows:

$$A_P^{N_i^{(1)}} = \left(G^{N_i'^{(1)}}\right)^{x_{P'}} \tag{3}$$

for some unknown value of $x_{P'}$, where $N_i'^{(1)}$ is some garbage data.

Hence, the Vendor will discover any altering of data on the Purchaser's side by verifying signatures on step 3 of Payment protocol.

Valid signatures $S_i^{(1)}$ and $S_i^{(2)}$ ensure correct values of $w_i^{(1)}$ and $w_i^{(2)}$, which are required for successful Shnorr identification. This comes from the fact that the unaltered data $A_P^{N_i^{(1)}}$ and $A_P^{N_i^{(2)}}$ is invertible and hence

$$w_i^{(1)} = P_i^{(1)} \cdot \left(A_P^{N_i^{(1)}}\right)^{-1}, \tag{4}$$

$$w_i^{(2)} = P_i^{(2)} \cdot \left(A_P^{N_i^{(2)}}\right)^{-1}. \tag{5}$$

Since identities (4) and (5) hold, Vendor will discover forgery of these values on step 8 of Payment protocol. We now claim the following:

Proposition 2. *Purchaser cannot forge any data sent during Payment protocol.*

Hence the following corollary is true:

Corollary 1. *All unfair actions of **MP** adversary will be discovered by the Vendor.*

We now consider the scenarios of **MitM** attacks. Let us assume that an inside adversary **MP** has intercepted the Payment protocol and has acquired the data sent by another legitimate Purchaser. His objective is to obtain the goods using the victim's e-wallet. To achieve this goal, he has to forge victim's personal data by replacing it with his own. However, in this case he has to deal with the following DLP:

$$A_P^{m_i||t_i} = A_{MP}^{\tilde{m}_i||\tilde{t}_i} \tag{6}$$

for some unknown variable $\tilde{m}_i||\tilde{t}_i$, where A_{MP} is the attacker's public key. Furthermore, since an attacker cannot affect any of the signatures acquired, due to Proposition 1, he has to forge the value of $w_i^{(2)}$. Hence, he has to solve the following equation:

$$\left(A_{MP}^{Id_{MP}}\right)^{\tilde{m}_i||\tilde{t}_i} \cdot \tilde{w}_i^{(2)} = P_i^{(2)} \tag{7}$$

for some unknown value of $\tilde{w}_i^{(2)}$. This equation by itself does not pose any advantage for an attacker. However, to pass Schnorr identification phase an attacker lacks private values $\xi_i^{(1)}, \tilde{\xi}_i^{(2)}$ and thus has to solve the following equations:

$$w_i^{(1)} = G^{\xi_i^{(1)}}, \tag{8}$$

$$\tilde{w}_i^{(2)} = G^{\tilde{\xi}_i^{(2)}}. \tag{9}$$

Based on these facts we claim, that:

Proposition 3. *If **MP** can purchase goods using legitimate Purchaser's e-wallet, then he is able to solve the DLPs* (6)*,* (8) *and* (9) *in reasonable time.*

Let us now assume that an inside adversary **MV** has intercepted the Payment protocol and has acquired the data sent by an honest Purchaser. His objective is to deposit money meant for another legitimate Vendor. To achieve this goal an adversary has to forge victim's identity by switching it with his own. This is not possible, since the data sent to the Observer does not have this information. Furthermore, **MV**'s Observer can use only identity Id_{MP} and **MV** can in no way affect this. Hence the Observer discovers that the stolen transaction is not meant for **MV** on step 3 of the Deposit protocol by verifying signature $S_i^{(3)}$ and blocks the deposit.

Based on these results we claim that the following proposition holds:

Proposition 4. *Our system is resistant against Purchaser impersonation and Vendor impersonation **MitM** attack scenarios.*

To complete our analysis we consider Attack of **MV** scenario, i.e. actions, which can be executed by a dishonest Vendor to benefit from the deal with an honest Purchaser.

Double deposit is prevented by the fact that the Vendor is not able to forge time instance due to Proposition 2 which is also valid for him. Hence, his Observer discovers this attempt at step 2 of the Deposit protocol.

Deny of payment is prevented by writing a check during steps 8–11 of the Payment protocol since during steps 9 and 10 the Observer verifies signature $S_i^{(3)}$ and hence confirms that the payment took place by generating a signature S_V. Due to Proposition 1, which is also valid for the Vendor, honest Purchaser discovers a fake check $(Id_V^{m_i||t_i}, S_V)$ at step 12 of the Payment protocol.

The main goal or manipulation of data received by the Vendor is increasing the e-wallet balance disproportionately by affecting the payment sum. These manipulations may also involve forging other parameters, such as A_P. Note, however, that the Vendor is incapable to affect any of the signatures received due to Proposition 1, which is also valid for him. Any attempts to forge the value A_P result in a solution of discrete logarithm problem since $A_P = G^{x_P}$ and hence:

$$A_P{}^{m_i||t_i} = G^{x_P \cdot m_i||t_i} = \left(G^{m_i||t_i}\right)^{x_P}.$$

Due to the latter identity, forgery of the payment sum would imply the following equation:

$$A_P{}^{m_i||t_i} = \left(G^{\tilde{m}_i||\tilde{t}_i}\right)^{x} \tag{10}$$

for some unknown x, where $\tilde{m}_i$ is the forged payment sum and $\tilde{t}_i$ is the forged time instance. Hence, we state that:

Proposition 5. *If Vendor can manipulate the payment sum, then he is able to solve the discrete logarithm problem* (10) *in reasonable time.*

Remark 1. The latter proposition is also valid for Purchaser.

Due to validity of signatures received, the Vendor's Observer discovers any forgery by the Vendor on step 3 of Deposit protocol.

Based on the presented results we state that:

Proposition 6. *Any unfair actions of **MV** will be discovered.*

Hence relying on Propositions 1, 4 and 6 we conclude that:

Proposition 7. *Our e-money system is secure against active inside attacks.*

Table 2
BAN logic notation.

Notation	Description
$A\|\equiv X$	A trusts X
$A\|\Rightarrow X$	A has jurisdiction over X, in other words A is the authority on X and is to be trusted on it;
$P\overset{k}{\leftrightarrow}V$	Shared key k between P and V
$\#X$	X is fresh
$A \triangleleft X$	A sees X
$A\|\sim X$	A said X(without implying that this utterance was recent or not)
(X, Y)	X or Y is one part of (X, Y)
$\langle X\rangle_k$	X is combined with k
$\{X\}_k$	X is encrypted with k
$A \ni M$	A possesses M
$\overset{K}{\rightarrow}P$	P has a public key.

4. Trustworthiness Analysis

Trustworthiness of the proposed e-cash system is analysed using Burrows–Abadi–Needham (BAN) logic. BAN logic was first presented in Burrows *et al.* (1989) and is a set of rules that can be used to define and analyse the trustworthiness of a cryptographic protocol. BAN logic seeks to determine whether the exchanged information between different parties is trustworthy from malicious insiders such as malicious bank, vendor, purchaser or others. BAN logic starts with a set of goals that are to be proven, and relies on the assumptions which should be made and used as a basis for the proof. The main BAN logic notations are presented below.

Firstly, as mentioned above, the proposed scheme uses the following parameters:

Purchaser's parameters: $PrK_p = x_P$, $PuK_p = \{G, A_P = G^{x_P}\}$.

Purchaser Observer's parameters: $PrK_O = x_O$, $PuK_O = \{G, A_O = G^{x_O}\}$.

We are going to keep original notation for parameters $\xi_i^{(1)}, \xi_i^{(2)}, A_P^{Id_P}, w_i^{(1)}, w_i^{(2)}$, $N_i^{(1)}, N_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}$ in order to provide clarity for further analysis.

In order to check the correctness and security of our payment protocol, we will set the following goals:

G1: The Vendor believes in the validity of the received payment:

$$\mathbf{V}|\equiv m_i.$$

G2: The Vendor trusts the Purchaser:

$$\mathbf{V}|\equiv \mathbf{P}.$$

G3: The Purchaser trusts the Vendor:

$$\mathbf{P}|\equiv \mathbf{V}.$$

We use the following assumptions as a base for proving the correctness of these goals:

A1: The Vendor trusts that public parameters of the Purchaser G, A_P, $A_P^{Id_P}$ as well as the public key A_O of his Observer O_P are not forged in any way:

$$\mathbf{P}, \mathbf{V}| \equiv G, A_P, \ A_P^{Id_P}, A_O.$$

Note that this is a valid assumption, since the Purchaser generates temporary key during each transaction whereas his Observer's data was pre-generated by the Bank.

A2: The Vendor trusts Observers as they represent the Bank:

$$\mathbf{P}, \mathbf{V}| \equiv \mathbf{O_P}, \mathbf{O_V}.$$

A3: The Vendor receives correct public information from the Purchaser:

$$\xrightarrow{G, A_P, A_P^{Id_P}, A_O} \mathbf{V}.$$

A4: The Purchaser receives correct public information from his Observer:

$$\xrightarrow{A_O, G, A_P^{Id_P}} \mathbf{P}.$$

E-cash withdrawal and payment protocols involve sending the following parameters:

M1: Data, generated by the Purchaser's Observer, is sent to the Purchaser:

$$\mathbf{O_P} \rightarrow \mathbf{P}: \quad \begin{array}{c} \xi_i^{(1)}, \xi_i^{(2)}, \ g^{\xi_i^{(1)}}, \ g^{\xi_i^{(2)}}, \\ \left(m_i||t_i||Id_V\right), \ \left(m_i||t_i||Id_V\right) \cdot g^{\xi_i^{(1)}}, \\ \left\langle {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} \right\rangle_{x_O}, \\ \left\langle A_P^{Id_P} \cdot {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} \right\rangle_{x_O}, \\ \left\langle a^{(m_i||t_i||Id_V)} \right\rangle_{x_O}. \end{array}$$

M2: The Purchaser sends the transaction data to the Vendor:

$$\mathbf{P} \rightarrow \mathbf{V}: \quad \begin{array}{c} \left(m_i||t_i\right), \langle {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} \rangle_{x_O}, \\ \left\langle A_P^{Id_P} \cdot {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} \right\rangle_{x_O}, \\ \left\langle {A_P}^{(m_i||t_i||Id_V)} \right\rangle_{x_O}, \ \left\langle {A_P}^{(m_i||t_i||Id_V)} \right\rangle_{x_O}. \end{array}$$

M3: The Vendor sends Shnorr identification challenge to the Purchaser:

$$\mathbf{V} \rightarrow \mathbf{P}: \ h_i.$$

M4: The Purchaser sends response parameters to authenticate himself to the Vendor:

$$\mathbf{P} \rightarrow \mathbf{V}: \begin{array}{c} h_i \cdot x_P \cdot (m_i||t_i||Id_V)\xi_i^{(1)}, \\ h_i \cdot x_P \cdot Id_p \cdot (m_i||t_i||Id_V) + \xi_i^{(2)}. \end{array}$$

M5: The Vendor sends response parameters to authenticate himself to the Purchaser:

$$\mathbf{V} \rightarrow \mathbf{P} : Id_V, \left\{Id_V^{m_i||t_i}\right\}_{A_O}.$$

It follows from M2 that the Vendor receives the following data from the Purchaser:

$$\mathbf{P} \rightarrow \mathbf{V}: \begin{array}{c} (m_i||t_i), \left\langle {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} \right\rangle_{A_O}, \\ \left\langle A_P^{Id_P} \cdot {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} \right\rangle_{x_O}, \left\langle a^{(m_i||t_i||Id_V)} \right\rangle_{A_O}. \end{array}$$

The application of message seeing rule results in the fact that the Vendor sees the data, received from the Purchaser:

$$\mathbf{V} \triangleleft \begin{array}{c} (m_i||t_i), \left\langle {A_P}^{(m_i||t_i||Id_V)} \right\rangle_{A_O}, \left\langle {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} \right\rangle_{A_O}, \\ \left\langle A_P^{Id_P} \cdot {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} \right\rangle_{A_O}. \end{array}$$

The application of message meaning and belief rules and the use of Purchaser Observer's public key results in the fact that the Vendor believes in the validity of data, generated by the Purchaser's Observer:

$$\mathbf{V}| \equiv \mathbf{O_P}| \sim \left({A_P}^{(m_i||t_i||Id_V)}, {A_P}^{(m_i\|t_i\|Id_V)} \cdot g^{\xi_i^{(1)}}, A_P^{Id_P} \cdot {A_P}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}}\right).$$

It follows from the belief rule that:

$$\mathbf{V}| \equiv \mathbf{O_P}| \sim {A_P}^{(m_i\|t_i\|Id_V)}.$$

Since A_P is a public key, the Vendor believes in the fact, that the received transaction is meant for him and that the sum m_i and the time instance t_i are approved by the Bank:

$$\mathbf{V}| \equiv \mathbf{O_P}| \sim m_i||t_i.$$

Subsequently the Vendor believes in the validity of these parameters:

$$\mathbf{V}| \equiv \mathbf{O_P}| \equiv m_i||t_i,$$

$$\mathbf{V}| \equiv \mathbf{O_P}| \equiv m_i.$$

The application of nonce-verification rule, jurisdiction and control, and the assumption that the Observer is trusted by all parties' results in the proof of the goal G1:

$$\mathbf{V}| \equiv m_i.$$

Now we consider the second goal. The Vendor sees the following information received from the Purchaser:

$$\mathbf{V} \triangleleft \left[\left\langle h_i'; \left(m_i'||t_i'||Id_V\right)\right\rangle_{A_P,\ \xi_i^{(1)}},\ \left\langle h_i',\ Id_p',\ \left(m_i'||t_i'||Id_V\right)\right\rangle_{A_P,\ \xi_i^{(2)}}\right].$$

By applying the message meaning rule and assumption A3 we obtain:

$$\mathbf{V}| \equiv \mathbf{P}| \sim \left[\left(m_i'||t_i'\right),\ {A_P}^{(m_i\|t_i\|Id_V)} \cdot g^{\xi_i^{(1)}},\ A_P^{Id_P} \cdot {A_P}^{(m_i\|t_i\|Id_V)} \cdot g^{\xi_i^{(2)}},\ {A_P}^{(m_i\|t_i\|Id_V)}\right],$$

i.e. the Vendor believes that it was the Purchaser, who sent him the specified data. Moreover, it follows from assumption A3 and concatenation rules that, due to correct values of total price m_i and the time instance t_i, it is the Purchaser, who is interested in acquiring the goods:

$$\mathbf{V}| \equiv \mathbf{P}| \sim \left[(m_i||t_i),\ {A_P}^{(m_i||t_i)} \cdot g^{\xi_i^{(1)}},\ A_P^{Id_P} \cdot {A_P}^{(m_i||t_i)} \cdot g^{\xi_i^{(2)}},\ {A_P}^{(m_i||t_i)}\right],$$

$$\mathbf{V}| \equiv \mathbf{P}| \sim (m_i||t_i).$$

We now apply the nonce-verification rule:

$$\mathbf{V}| \equiv \mathbf{P}| \Rightarrow m_i||t_i,$$

$$\mathbf{V}| \equiv \mathbf{O_P} \Rightarrow Id_p,$$

and hence the Vendor trusts that the Purchaser obtained the desired sum m_i from the Bank at the time t_i, i.e. the Purchaser has jurisdiction to spend this sum of money. Furthermore, the Vendor also believes that the Bank has the jurisdiction over the Purchaser via his representative (Observer $\mathbf{O_P}$). Furthermore, the Vendor believes that the Purchaser knows his identity:

$$\mathbf{V}| \equiv \mathbf{P} \triangleleft Id_p.$$

Finally, using jurisdiction, control and referencing to the rules above, the Vendor trusts the Purchaser's identity:

$$\mathbf{V}| \equiv Id_p.$$

The second goal $\mathbf{V}| \equiv \mathbf{P}$ now follows from the proven results $\mathbf{V}| \equiv Id_p$ and $\mathbf{V}| \equiv m_i$, since Vendor trusts the Purchaser's identity and fairness (the sum m_i is not forged).

Now we consider the third goal. Due to M5, the Purchaser sees the following information received from the Vendor:

$$\mathbf{P} \triangleleft \left[Id_V, \left\{Id_V^{m_i||t_i}\right\}_{A_O}\right].$$

Note that the Vendor received this data from his Observer, implying that:

$$\mathbf{V} \triangleleft \left[Id_V, \left\{Id_V^{m_i||t_i}\right\}_{A_O}\right].$$

By applying the message meaning rule, concatenation rule, and assumption A4 we obtain:

$$\mathbf{P}| \equiv \mathbf{O_V}| \sim \left[Id_V, \left\{Id_V^{m_i||t_i}\right\}, A_O\right],$$

i.e. the Purchaser believes that it was the Vendor's Observer, who generated the signature. Moreover, it follows from assumptions A4 and concatenation rules that, due to correct values of total price m_i and the time instance t_i, the Purchaser is dealing with an honest Vendor:

$$\mathbf{P}| \equiv \mathbf{O_V}| \sim Id_V.$$

We now apply the nonce-verification rule:

$$\mathbf{P}| \equiv \mathbf{O_V}| \Rightarrow Id_V.$$

Hence, the Purchaser believes that the Vendor's Observer has jurisdiction over the Vendor. Furthermore, due to this fact, the Purchaser trusts that the Vendor knows his identity since his Observer possesses this information:

$$\mathbf{P}| \equiv \mathbf{V} \triangleleft Id_V,$$

$$\mathbf{P}| \equiv \mathbf{O_V} \triangleleft Id_V.$$

Finally, using jurisdiction, control and referencing to the rules above, the Purchaser trusts the Vendor's identity:

$$\mathbf{P}| \equiv Id_V.$$

Hence, the validity of the third goal $\mathbf{P}| \equiv \mathbf{V}$ now follows from the proven results.

5. Investigation of Execution Time

Since the considerable amount of payment operations is performed in Observer having restricted computation resources, the effectivity of proposed e-wallet system depends on the estimation of the operation time.

The computation time is directly related with the processor's clock frequency. If processor is running at 1 GHz clock frequency, then its clock cycle takes 10^{-9} s $= 1ns$ time.

The arithmetic operations required to perform a payment protocol is multiplication and addition together with shifting operation all performed in the registers of Observer. We name those operations as elementary operations.

We assume that 32 bits' microprocessor is used in Observer. It is far less than the bit length of variables used in payment protocol represented by 2048 bit integers. Without the loss of generality, we assume that all elementary operations take one clock cycle.

Table 3
Bit length of variables.

Variable	Bit length
$p, q, x_P, x_O\ A_P, A_O,\ G,\ Id_P, h_i,\ R$	2048 bits
$\xi_i^{(1)}, \xi_i^{(2)}, w_i^{(1)}, w_i^{(2)}, N_i^{(1)}, N_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, r_i^{(1)}, r_i^{(2)}$	2048 bits
m, t	∼18 bits
$m_i \| \| t_i$	∼ 36 bits
$H(m)$	∼256 bits

The most time consuming operation is the exponentiation modulo p of length in 2048 bits. For the assessment of computation time, firstly, we must estimate the number of elementary operations required for the calculation of the modular exponent function $r = g^k \bmod p$.

According to Knuth (1981), Hwang *et al.* (2005), the modular exponent function is computed using *addition chain method* (Knuth, 1981). The formulas to find the number of those operations are the following:

$$MOD_E(k, p) = 1,5 \cdot l(k)\big[M\big(l(p)\big) + 2Mod\big(l(p)\big) + 1\big],$$

where:

$$M(w) = 3M(w/2) + 5A(w) + 2S,$$

$$A(w) = w/32,$$

$$Mod(w) = Mod(w/2) + 4M(w/2) + 1,5A(w) + 3S.$$

1. $MOD_E(k, p)$ – denotes an operation of modular exponentiation $r = g^k \bmod p$;
2. $M(w),\ A(w),\ Mod(w)$ – denote operations of multiplication, addition and modulus with the bit length of operand is w;
3. $l(w)$ – denotes the bit length of w;
4. S denotes the shift operator.

The bit lengths of the variables in our scheme are presented in Table 3.

By default, we take 82 clock cycles for SHA-2 computation (Guilford *et al.*, 2012).

After the number N of clock cycles is found, the operation time can be estimated in the following way: $Time = N \cdot T$, where $T = 1/F$ and F is a clock frequency. We assume $F = 1.6$ GHz in further steps for the demonstration of calculations results.

By Hinterwälder *et al.* (2013), Hinterwälder *et al.* (2015) all Brands e-cash protocols take about 2966 ms in all protocols generated in cards. By Au *et al.* (2007) the computational time of CHL e-cash protocol in single payment is 30 modular exponentiations and takes about 2111 ms by Juang (2010) approximations, but it's cost of each operation is somehow hard to compute because it depends of how many transactions have been made before and how many coins will be used. The comparison of our system with Brands and CHL systems is presented in Table 4.

Table 4
Computation time comparisons in ms.

Protocol	Our system	Brands	CHL
Withdrawal	665	–	–
Payment	1241	–	–
Deposit	629	–	–
Total	2535	2996	2111

Table 5
Comparison of e-money schemes.

E-money systems	Year	Transf.	Trace.	Data grows	Anon. against V.	Anon. against B.
CHL	2005	Yes	No	Yes	Yes	Yes
fairCASH	2006	Yes	No	Yes	Yes	Yes
Endorsed	2007	Yes	Yes	Yes	Yes	Yes
Secret splitting	2009	Yes	Yes	Yes	Yes	Yes
GS proof e-cash	2011	Yes	Yes	Yes	Yes	Yes
Baldimtsi	2015	Yes	Yes	Yes	Yes	Yes
Canard e-cash	2015	No	No	Yes	No	No
Märtens	2015	No	Yes	Yes	Yes	Yes
Scalable e-cash	2015	Yes	No	Yes	Yes	Yes
Dissertation	2018	Yes	No	No	Yes	No

Hence, our system requires approximately the same computation time, while its functionality has a significant advantage with respect to others.

In Table 5 we present a comparison of some offline payment e-cash systems and explore such properties as: transferability, traceability, data growth and anonymity (against Vendor and Bank). However, each of these systems possesses the flaw of money growth in size when transferred. Furthermore, any previously presented e-cash system, which eliminates this flaw, also loses anonymity against Vendor or other offline payment properties.

As it can be seen, our e-cash system has the following functional advantages: anonymity against Vendor, offline payment, divisibility, transferability and double-spending prevention requirements, and most important one – data does not grow in size when transferred.

Based on the presented comparison we conclude that our system stands out from the other systems since it possesses similar characteristics as those schemes while also avoiding a drawback of growing in size. Furthermore, it has better or approximate computation time if compared to other schemes presented in Table 5.

6. Conclusions and Discussion

Proposed offline, divisible, anonymous and transferable e-cash system with observers is analysed in few ways: an existential forgery, trustworthiness analysis using BAN logic, and computation time comparison using addition chain method is provided.

Our transferable e-cash system does not possess the data growth in size problem due to the usage of observers.

The estimated execution time of all three protocols is 2535 ms, independently of how many e-cash has been spent.

All these characteristics allows us to claim that proposed e-cash system is adequate to medium payment usage.

References

Au, M.H., Susilo, W., Mu, Y. (2007). Practical compact e-cash. In: *Australasian Conference on Information Security and Privacy*. Springer, Berlin, Heidelberg, pp. 431–445.

Brands, S.A. (1993). An efficient off-line electronic cash system based on the representation problem. *Centrum voor Wiskunde en Informatica*.

Burrows, M., Abadi, M., Needham, R.M. (1989). A logic of authentication. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 426(1871), 233–271.

Chan, A., Frankel, Y., Tsiounis, Y. (1998). Easy come-easy go divisible cash. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, pp. 561–575.

Chaum, D., Fiat, A., Naor, M. (1988). Untraceable electronic cash. In: *Conference on the Theory and Application of Cryptography*. Springer, New York, NY, pp. 319–327.

Chaum, D., Pedersen, T.P. (1992). Transferred cash grows in size. In: *Workshop on the Theory and Application of Cryptographic Techniques*, Springer, Berlin, Heidelberg, pp. 390–407.

Cramer, R., Shoup, V. (2003). Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1), 167–226.

D'Amiano, S., Di Crescenzo, G. (1994). Methodology for digital money based on general cryptographic tools. In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, Berlin, Heidelberg, pp. 156–170.

Diffie, W., Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644–654.

Eng, T., Okamoto, T. (1994). Single-term divisible electronic coins. In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, Berlin, Heidelberg, pp. 306–319.

Guilford, J., Yap, K., Gopal, V. (2012). Fast SHA-256 implementations on Intel architecture processors. *IA Architects*.

Fuchsbauer, G. (2009). Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. *IACR Cryptology ePrint Archive*, 2009, 320.

Hinterwälder, G., Zenger, C.T., Baldimtsi, F., Lysyanskaya, A., Paar, C., Burleson, W.P. (2013). Efficient e-cash in practice: NFC-based payments for public transportation systems. In: *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, Berlin, Heidelberg, pp. 40–59.

Hinterwälder, G., Riek, F., Paar, C. (2015). Efficient E-cash with attributes on MULTOS smartcards. In: *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, Cham, pp. 141–155.

Hwang, R.J., Su, F.F., Yeh, Y.S., Chen, C.Y. (2005). An efficient decryption method for RSA cryptosystem. In: *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*. IEEE, pp. 585–590.

Juang, W.S. (2010). RO-cash: an efficient and practical recoverable pre-paid offline e-cash scheme using bilinear pairings. *Journal of Systems and Software*, 83(4), 638–645.

Knuth, D.E. (1981). *The Art of Programming, Vol. 2, Semi-Numerical Algorithms*. Addison Wesley, Reading, MA.

Kreft, H., Adi, W. (2006). fairCASH-A digital cash candidate for the proposed GCC gulf dinar. In: *Innovations in Information Technology*. IEEE, pp. 1–5.

Muleravičius, J., Sakalauskas, E., Timofejeva, I. (2016). On methodology of E-wallet construction for partially ff-line payment system. In: *International Conference on Information and Software Technologies*, Springer, Cham, pp. 753–765.

Okamoto, T. (1995). An efficient divisible electronic cash scheme. In: *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, pp. 438–451.

Pfitzmann, A., Köhntopp, M. (2001). Anonymity, unobservability, and pseudonymity – a proposal for terminology. In: *Designing Privacy Enhancing Technologies*. Springer, Berlin, Heidelberg, pp. 1–9.

Petersen, H., Poupard, G. (1997). Efficient scalable fair cash with off-line extortion prevention. In: *International Conference on Information and Communications Security*. Springer, Berlin, Heidelberg, pp. 463–477.

Pointcheval, D., Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3), 361–396.

Rosenberg, B. (2010). *Handbook of financial cryptography and security*. CRC Press.

Stadler, M., Piveteau, J.M., Camenisch, J. (1995). Fair blind signatures. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, pp. 209–219.

Sakalauskas, E., Timofejeva, I., Michalkovič, A., Muleravičius, J. (2018). A simple off-line E-cash system with observers. *Information Technology and Control*, 47(1), 107–117.

Tsiounis, Y. (1997). *Efficient Electronic Cash: New Notions and Techniques*. PhD thesis, College of Computer Science.

Waters, B. (2005). Efficient identity-based encryption without random oracles. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, pp. 114–127.

J. Muleravicius is currently seeking a PhD degree at Kaunas University of Technology. His research interest is in cryptography focusing in development and analysis of e-cash systems. During the last 5 years he made contributions to 3 papers on this topic.

I. Timofejeva is currently seeking a master degree at Kaunas University of Technology. One of her many scientific research interests is cryptography. Currently her interests are focused in the development and analysis e-cash systems. She contributed to 3 papers on this topic.

A. Mihalkovich received PhD degree from Kaunas University of Technology, in 2015. Currently he is an assistant professor in Department of Applied Mathematics in Kaunas University of Technology. His main research interest is focused in cryptography and cryptanalysis. During the last 5 years he made contributions to 4 papers published in Thompson Reuters database journals.

E. Sakalauskas received PhD degree from Kaunas Polytechnical Institute, in 1983. Currently he is a professor in Department of Applied Mathematics in Kaunas University of Technology. In recent time his research interests are focused in cryptography. The main research results in this field were published in over 15 papers.