KAUNAS UNIVERSITY OF TECHNOLOGY

JONAS MULERAVIČIUS

THE E-WALLET IN AN E-COMMERCE SYSTEM

Doctoral dissertation
Natural Sciences, Informatics (N 009)

2019, Kaunas

This doctoral dissertation was prepared at the Faculty of Mathematics and Natural Sciences, Department of Applied Mathematics of Kaunas University of Technology during the period of 2014-2018.

Scientific Supervisor:
Prof. Dr. Eligijus SAKALAUSKAS (Kaunas University of Technology, Natural Sciences, Informatics, N 009).

Doctoral dissertation has been published in:
http://ktu.edu

Editor:
Vaiva Teniukaitė

ISBN 978-609-02-1643-9

The bibliographic information about the publication is available in the National Bibliographic Data Bank (NBDB) of the Martynas Mažvydas National Library of Lithuania.

KAUNO TECHNOLOGIJOS UNIVERSITETAS

JONAS MULERAVIČIUS

ELEKTRONINĖS PINIGINĖS REALIZACIJA E. VERSLO SISTEMOJE

Daktaro disertacija
Gamtos mokslai, Informatika (N 009)

2019, Kaunas

Disertacija rengta 2014−2018 metais Kauno technologijos universiteto Matematikos ir gamtos mokslų fakultete Taikomosios matematikos katedroje.

Mokslinis vadovas:
prof. dr. Eligijus SAKALAUSKAS (Kauno technologijos universitetas, Fiziniai mokslai, Informatika, N 009).

Interneto svetainės, kurioje skelbiama disertacija, adresas:
http://ktu.edu

Redagavo:
Vaiva Teniukaitė

ISBN 978-609-02-1643-9

Leidinio bibliografinė informacija pateikiama Lietuvos nacionalinės Martyno Mažvydo bibliotekos Nacionalinės bibliografijos duomenų banke (NBDB).

TABLE OF CONTENTS

TABLE OF FIGURES

ABBREVIATIONS AND NOMENCLATURE

This section describes some of the nomenclature used throughout this thesis:

MAC - message authentication code

SHA-2 - crypto hash function algorithm

PUF - physical unclonable functions

TTP - trusted third party

Observer – trusted third party implemented in the device

DDH assumption - decision-related Diffie Hellman assumptions

BAN logic - a set of rules for defining and analysing information exchange protocols

CDH assumption - Computational Diffie Hellman assumptions

CFN - the Chaum, Fiat, and Naor e-cash system

CHL - the Camenisch, Hohenberger, and Lysyanskaya e-cash system

FOLC - the fair off-line e-cash system

PID - proportional-integral-derivative controller

TPM - trusted platform module

BOOKMARKS

q, p - large prime numbers, such as p, which satisfy the strong prime property $p = 2q + 1$.

G_q - a cyclic subgroup.

G - a generator of multiplicative group Z_p^*.

$h(\,)$ - a hash function.

$Sig_{ElG}^{X}(m)$ - the ElGamal signature function, where m and X correspond to the message to be signed and the ElGamal private key of the signee.

$Ver_{ElG}^{A}(s, m)$ - the ElGamal signature verification function, where m, s, and A correspond to the message, the signature on the message, and the ElGamal public key of the signee.

i - a serial number of the transaction.

$PrK_P = x_P$ - the purchaser's temporary private key.

$PuK_P = \{G, A_P = G^{x_P}\}$ - the purchaser's temporary public key.

$PrK_O = x_O$ - the observer's private key.

$PuK_O = \{G, A_O = G^{x_O}, A_P^{Id_P}\}$ - the observer's public key.

Id_P - a unique identification number of the observer's chip.

m_i - an amount of money to be spent by the purchaser.

$\widetilde{m_i}$ - an actual price of the products to be bought by the purchaser.

t_i - a time instance for the e-cash withdrawal.

$m_i || t_i$ - a concatenation of the sum and the time instance.

t_{w0}, t_{p0}, t_{d0} - the time instance for the last e-cash withdrawal protocol (whether payment or deposit).

m_{max}^{P}, m_{max}^{V} - an amount of money in the e-wallet of the purchaser and the vendor, respectively.

$\xi_i^{(1)}, \xi_i^{(2)}$ - random values of Z_q^* for the Schnorr interactive identification protocol.

S_m - a signature of the message m.

$MOD_E(k, p)$ - denotes an operation of modular exponentiation $r = g^k mod p$.

$M(w), A(w), Mod(w)$ - denotes multiplication, addition, and modulus operations with the bit length of operand being w.

$l(w)$ - denotes a bit length of w.

S_m - a signature of the message m.

$MOD_E(k, p)$ - denotes an operation of modular exponentiation $r = g^k \bmod p$.

$M(w), A(w), Mod(w)$ - denotes multiplication, addition, and modulus operations with the bit length of operand being w.

$l(w)$ - denotes a bit length of w.

THE MAIN PARAMETERS AND ABBREVIATIONS USED IN THE E-CASH SYSTEM

Anonymity: a customer must remain anonymous in relation to the recipient of the money, as well as to the bank when he or she is using e-cash to pay for a product.

Partial anonymity: a customer must remain anonymous in relation to the recipient of the money, as well as to the bank when he or she is using e-cash to pay for a product. The possibility of the customer's identity being revealed must arise only when the money is being spent illegitimately.

Unreusability: e-cash cannot be duplicated or spent twice. This implies that the e-wallet system has to minimise the risk of forgery and/or provide ways in which dishonest users can be identified.

Unforgeability: only authorised parties (i.e. the bank) can produce e-cash.

Off-line payment: a payment transaction that is carried out off-line means that no third party needs to be involved, i.e., no communication with the bank should be necessary during the payment process.

Online payment: a payment transaction that requires internet and confirmation from a third party (e.g., bank) that the customer has enough money in his or her account.

Transferability: any e-cash amounts that are received can be applied to other payments amongst customers, regardless of whether transactions are online or off-line.

Divisibility: e-cash must be divisible; i.e., a customer should be able to divide it into smaller amounts.

INTRODUCTION

Problem: these days, a great number of various payment transactions are taking place. Almost half of them involve cash, while the other part of transactions are digital payments. The latter consists of card payments, bank transfers, e-payments and more. However, e-payments that take place off-line involve divisibility and full anonymity which has resulted in data expansion when e-cash transfers are made, as was shown by Chaum & Pedersen (1993).

The aim of the research is as follows:

Develop a new e-cash system which will aid in the construction of an e-wallet for mobile devices, with these characteristics:

1. The ability to make off-line payments;
2. Ensuring that e-cash is properly divisible;
3. Transfers between users has to be possible;
4. Data which is required to prevent 'double spending' - fraud - should not expand when carrying out transfers between users;
5. The purchaser must retain anonymity from the vendor;
6. The system has to be secure.

Basic tasks that are being aimed at in this work:

1. Carry out an analysis of the existing e-cash systems and their main properties;
2. Work towards the construction of a more attractive, secure, and advanced e-cash system for an e-wallet on a mobile device;
3. Check the trustworthiness of BAN logic under the new e-cash scheme;
4. Estimate digital simulation time of the new e-cash scheme in terms of theoretical processing time;
5. Certify security of the newly-constructed e-cash system.

Research methods:

Research methods used in this work in order to reach the desired conclusions included those with a mathematical basis, plus a trustworthiness analysis, a security analysis, as well as a digital simulation. The construction of a new e-cash system can be achieved using two of the main crypto systems, both of which are well known when it comes to analysing the existing e-cash systems.

An overview of the novelty:

The proposed e-cash system will ensure that data will not expand when transfers are made, thanks to the fact that it removes anonymity between the bank and the purchaser. This scheme retains the following main properties: off-line and divisible payments, anonymity between the purchaser and the vendor, a prevention of 'double spending', as well as untraceability and security.

The benefits of the new system as it is proposed here:

1.The new e-cash system ***will not expand in size*** when e-cash is being transferred between users. In addition, it has all of the other main properties of such a system: off-line payments; divisible e-cash; the ability to transfer e-cash between users; legal payments are untraceable; and the purchaser retaining anonymity from the vendor;

2. The new e-cash system is secure;
3. The new e-cash system can be fully trusted under BAN logic;
4. The new e-cash system can effectively be used on mobile devices.

Approbation of the research:

Two scientific papers have been published on this subject and both have been included in the list of papers in the "ISI Web of Science" database with the citation index. In addition, one paper was published in journal without citation index, and another one was published in conference proceedings. The results of the research of this dissertation were also presented at two international conferences.

The structure and volume of the dissertation:

The introduction of this dissertation provides an overview of the existing crypto systems. Section 2 shows methodology which is used for the model proposed. Section 3 contains a definition in an abstract form of the main protocols (regarding withdrawal, payment and deposit), as well as it outlines the scheme of the suggested e-cash system.

After the presentation of the new system, it is being checked in respect of BAN logic and security in Section 4; a digital simulation was carried out in order to determine the duration of the new e-cash system operations. Finally, the last section draws conclusions.

Results of the previously published papers are used in the research: we refer to 'A simple off-line e-cash system with observers' for the creation of the e-system protocols - Section 3; 'Security, trustworthiness, and effectiveness analysis for an off-line e-cash system with observers' for the security analysis and related improvements - Section 4; and 'Computational resources for a mobile e-wallet system with observers' for a digital simulation of the new e-system - Section 4.

1 AN OVERVIEW OF EXISTING E-CASH SYSTEMS

E-cash is digital money which makes it possible to pay for products and services without using paper currency. Transactions can be carried out over the internet or via email, using a personal computer or a mobile device, they are usually safe in terms of the point of sale and also anonymous.

As a good deal of the available literature describes (Au, Susilo, & Mu, 2011; Baseri, Takhtaei, & Mohajeri, 2013; Blazy et al., 2011; Brands, 2012; Chaum, Fiat, & Naor, 1988; Chaum & Pedersen, 1993; de Solages & Traorè, 1998; Eng & Okamoto, 2006; Eslami & Talebi, 2011; Fan, Huang, & Yu, 2013; Fuchsbauer, Pointcheval, & Vergnaud, 2009; Kreft & Adi, 2006; Muleravičius, Sakalauskas, & Timofejeva, 2016; Muleravicius, Timofejeva, Mihalkovich, & Sakalauskas, 2019; Okamoto, 1995; Pfitzmann & Köhntopp, 2007; Rosenberg, 2010; Yan Liang & Zhi-ming, 2016), any form of digital currency faces issues which can be linked to the following challenges:

- money laundering;
- the prevention of paying twice for the same purchase, a concept which can more easily be referenced as ‘double spending’;
- a loss of e-wallet storage;
- preserving customer anonymity;
- reducing online operations in a large database;
- e-coin forgery (which already takes place with physical currency).

Data expansion

The other drawback in the use of e-cash systems is that, according to Chaum & Pedersen (1993), divisible, off-line, untraceable, and anonymous e-cash being transferred between users is something that tends to expand its storage requirements, i.e., the amount of information storage that is required for e-cash is continually growing. Such information is needed in order to prevent ‘double spending’ and to retain its characteristics, such as divisibility.

Alternative systems have been created (in terms of an e-cash system) which will avoid data expansion such as has been proposed by D’Amiano & Di Crescenzo (2006), or by Okamoto (1995) but, as Tsiounis in Chan, Frankel, & Tsiounis (1998) suggests, an analysis of these e-schemes has produced one or two issues, such as the total payment amount not being able to exceed x, and protocols becoming inefficient under certain conditions.

In the two decades since the first system was created, the search is still on to find a solution to this problem. Fuchsbauer et al. (2009) attempted to construct ‘transferable e-cash without any increase in size’, but Fuchsbauer (2009) and Waters (2005) both stated that they still saw a dramatic increase in the public key size.
Abe, Haralambiev, & Ohkubo (2010) have left the construction of constant-sized signatures as an open problem.

Security is hard to prove when using complex cryptographic systems

The biggest problems in relation to the e-cash systems could be divisibility, off-line payments, and retaining the purchaser’s anonymity. According to Rosenberg (2010), ‘almost all divisible e-cash systems in the available literature to date rely on a

proof about double-discrete logarithms and almost all require similar sequences of prime numbers ('primes') in their setup'.

Simultaneously, the use of off-line e-cash with observers was first mentioned by Brands (1994). He proposed the idea of a trustee for the purchaser in order to carry out payments without connecting to a bank. The cryptographic security of Brands' e-cash system was never proven and hence this system was never activated.

It was noted in Brands (1994) and Cramer & Shoup (2004) that the decisive Diffie-Hellman (DDH) assumption is required in order to prove the cryptographic security of e-cash system protocols. As Brands (1994) and Cramer & Shoup (2004) describe, this comes from the fact that the Diffie-Hellman key exchange cannot be proved as being secure in any reasonable and standard way based only on the computational Diffie-Hellman (CDH) assumption: the DDH assumption is also required.

As a result, a practical, divisible e-cash system, therefore, remains an open problem.

The usability of e-cash

Despite its initial introduction as far back as 1980, anonymous e-cash has still not become especially widespread around the world. The DigiCash based CFN (see Chaum et al., 1988; Rabin, 1978) and MojoNation (which used its own e-cash system) both stumbled along the way and are no longer in use today according to Rosenberg (2010).

The main challenge of all e-cash systems, as described in the available literature (e.g., Rosenberg, 2010), is the lack of any strong security analysis of the existing e-cash systems due to the complexity of their realisation.

Generally, e-cash uses some form of tracking technique ('blind' techniques, trustee-based techniques, or open/close-loop payment) to ensure the integrity of the system.

Rosenberg (2010) also emphasised that the recent trend is towards a non-anonymous e-cash system. Partially anonymous systems, such as Octopus cards, make use of this approach. Octopus cards are used anonymously, but every transaction can be checked by the staff of Octopus itself. In other words, they know the details of every transaction that is carried out by the customer, while the vendor receives no information about the customer. This shows that a level of partial anonymity has been achieved.

The new trend in terms of a working e-system could be the sacrifice of full anonymity for the purchaser (transforming any usage into partial anonymity), in order to avoid data expansion or to solve other problems.

1.1 CHAUM INVENTS E-CASH AND OTHER CASH SYSTEMS

One of the first e-cash systems, one which was based on a cut-and-choose approach (Chaum et al., 1988; Rabin, 1978) was introduced by Chaum, Fiat, and Naor (abbreviated to 'CFN') in 1988. In fact, and as acknowledged by some of the available literature (e.g., Rosenberg, 2010), the system was ineffective.

The bank had to store $2k + 3k^2$ bits of data (k is the bank's secret key) after each deposit. The user had to store $2k + 4k^2$ bits for each e-coin in his or her e-wallet, while the vendor had to manage with $2k + 3k^2$ bits.

In addition, as Everett comments in his blog (Everett, 2016), Chaum was one of the pioneers of the 1980s who was promoting digital cash, but it failed to catch on and quietly slipped away at the end of the 1990s following the product launch by the Mark Twain bank.

One of the problems that were proven in theory by Chaum and Pedersen (Chaum & Pedersen, 1993), and one of the reasons for e-cash being forgotten in the 1990s, was that transferred divisible cash expands in terms of its data requirements. This is a side effect of having to work to prevent money laundering. In other words, a bank has to store a large volume of data, in order to prevent 'double spending' and in addition, each and every e-coin has to carry its own transaction history wherever it goes.

In 2005, Camenisch, Hohenberger, and Lysyanskaya (CHL) introduced Compact E-Cash (Camenisch, Hohenberger, & Lysyanskaya, 2005). The basic idea behind this e-cash system was to use a pseudo-random function to generate a sequence of serial numbers from a single seed (a form of a unique ID).

The bank signs on the purchaser's secret seed value, s, and then sets up e-coins with serial numbers: $F_s(0), F_s(1)\ to\ F_s(W-1)$, where W is the amount of money in the purchaser's e-wallet (Camenisch et al., 2005; Rosenberg, 2010). Rosenberg (2010) stated that the bank had to store $3k$ bits of data after each deposit, where k is the bank's secret key. So, this format shows some improvement when compared to Chaum's e-cash system. The purchaser had to store $11k + \log(W)$ bits on his or her device while the vendor also had to store $3k$ and $k\ +\ \log(W)$. The CHL compact e-cash system was no better than other e-cash systems were, thanks to the amount of data that had to be stored in each database by the bank, the purchaser, and the vendor (in other words the e-wallet of the purchaser, the vendor, and the bank). Every purchaser could make a payment himself or herself (by generating his or her own e-coins). Thus, instead of applying to the bank a user could generate one's e-coins with the help of a secret seed value *s* individually.

In 2007, the same group of authors (Camenisch, Lysyanskaya, & Meyerovich, 2007) modified the CHL e-cash system, referring to it as unendorsed e-cash. They split the payment protocol into two stages. Firstly, the purchaser gives the vendor a blind e-coin (an unendorsed coin). This e-coin is not real and cannot be deposited in a bank. The purchaser is allowed to produce unendorsed coins as often as they want to, and it is impossible to generate two e-coins that are the same.

In Crypto '95, Okamoto (Okamoto, 1995) was the first one who presented a really efficiently performed e-cash protocol with divisibility parameters in the e-cash system. This result has been proven as being asymptotically optimal by Chan et al. (1998), and by Okamoto & Ohta (2007). All of the protocols (except the registration protocol) are more efficient or have at least the same level of efficiency as that of other systems without the divisibility. For the registration protocol which takes more than 4,000 multi-exponentiations modules, a 1030 bit prime is used. Hence, according to Okamoto (1995), this e-cash system which was invented by Okamoto himself is practical only when an account is opened infrequently (generally once) for each user.

In 1994, the concept of off-line e-cash with its built-in 'observers' was first mentioned by Brands (1994). He proposed the idea of a trustee for the purchaser to carry out payments without connecting to the bank. The cryptographic security of Brands' e-cash system was never confirmed and hence this system was never adopted. In the following sections, we will extend the concept of the e-system with its built-in 'observers'.

1.2 MONDEX

At the end of the 1990s, Jones and Higgins, with architecture by Everett, invented a form of e-cash which could be transferred in off-line mode by using public key cryptography (Stepney, Cooper, & Woodcock, 2000). It was implemented in the payment card's microchip. In 1997, Mondex came under the control of MasterCard when they invested a 51% stake into the company, and in 2001 they became a wholly-owned subsidiary of MasterCard International.

A number of forums have claimed that Mondex has not yet divulged information about the algorithms that are employed in their system.

As explained in Clarke (1996), Mondex maintains monetary values in microchips (RFID) in the form of electronic information - an entirely different approach from that of the use of physical money. When required, this information securely moves from the card's chip to the chip of another card (for example the vendor card's chip), which means that the Mondex e-cash system permits person-to-person payments. Mondex is an e-payment system which meets the requirements of high levels of security, while also allowing person-to-person payments, divisible payments, off-line payments and supporting multi-currency.

This payment method allows Mondex to empower institutions to tap into new markets, answering the growing need for secure internet payments that are widely used for a low cost, giving non-bank users their first e-cash cards and enabling person-to-person payments (without any third party intervention).

Thanks to the rating which was given in Common Criteria (2017), in 1999 Mondex achieved Level E6 security rating of 'Information Technology Security Evaluation Criteria' (ITSEC) for producing the only available smart card application. That is the highest possible rating that can be achieved under the rigorous, internationally recognised ITSEC security process. As a set of criteria for evaluating computer security levels, ITSEC operates on a scale of ascending levels of assurance (levels E0 to E6), which can be placed in the security functions, thereby determining the rigor of the evaluation.

In 2001, Mondex reached an important and highly essential step in terms of implementing its security requirements when EAL4+ certification was achieved under the terms of the Common Criteria IT security assessment system. No other e-cash system has received such a high level rating from the security community.

As Stepney et al. (2000) states, 'The e-cash facility enables the instant transfer of value between vendor and purchaser (or between other consumers) and does not require bank authorisation. It also allows users to make secure online purchases without giving up any personal details'. This is the basic idea behind any off-line, anonymous, and divisible e-cash system, but it has usually been promoted as a simple

credit card without promoting its real added value. The fact that there is a noticeable difference in the MasterCard and Mondex card systems in terms of how they treat the customer is something that has not been properly promoted.

However, the Mondex system carries with it one of the disadvantages: if a Mondex card is lost, the funds it contains are also lost, as was described in the introduction above - most e-cash wallets have the same issue.

The electronic purse was first implemented as 'The Byte' card in 1992. The Byte was distributed to about six thousand employees for use in the lunchroom. It was selected to act as an electronic purse, as a test to prove usability for the technology that comprises an e-wallet. When its activity was successfully demonstrated in a retail environment, a second trial was managed by Mondex UK.

Two issues arose which were of some concern: the fact that a regulator would be needed for this e-currency system; and the possibility of money laundering. It was suspected that sooner or later criminals would figure out how to forge e-cash or use e-cash for money laundering on the black market, or perhaps one of several other scenarios.

Good security, according to some works (e.g., Rosenberg, 2010) required three components - 'prevention, detection, and recovery'. All of these together contribute more towards security than they would alone. But there is no solid proof that the Mondex system is secure.

1.2.1 Mondex off-line value-transfer data

Based on the available literature (Clarke, 1996; Schmitt & Tonin, 2007; Stalder, 2002; Tam & Ho, 2011), we provided an overview check on Mondex payments in its off-line mode. A value transfer transaction is the result of communication between two of the Mondex's chips that are located on smart devices. It can be carried out in off-line mode which means that there is no need of an internet connection. Transaction data is recorded in three locations:

1. on the purchaser's card, which retains the following information about the last ten transactions:
   - date and time as provided by the terminal of where the payment is carried out;
   - whether the transaction is a debit from or a credit to the card's balance;
   - a payment figure;
   - information which identifies the party which receives the e-money, as provided by the payee's chip ID;
   - PID of the recipient's Mondex chip through which the payment was carried out;
2. on the payee's card (which could include a retailer's card, which may be located at a retailer's point of sale or in their e-wallet; or on another purchaser card, using an e-wallet). The payee's card records the same information as is recorded on the purchaser's card for the last ten transactions;
3. in the case of the retailer's terminals data is recorded in non-volatile memory in the terminal. The terminal retains the same data as the payee's card, but only for a

limited period of time. This limit is currently imposed by the card's capacity which can hold only the most recent three hundred transactions.

Mondex International (Schellhorn, Grandy, Haneberg, & Reif, 2006) argues that the e-scheme does not require any identity details of a card to be stored or transferred to other cards with which it carries out transactions. The Swindon study, and hence the proposed implementation, provides incomplete or confirmatory data about the cardholder's identity. In the case of Midland Bank, this includes the cardholder's initials while NatWest requires the first seven characters from the cardholder's name. Depending on how this feature is used, this may have a relatively limited impact, or it may dramatically change the privacy profile of each Mondex installation.

Clarke (1996) points out that the terminal transaction route allows the retailer to download data into a database. The value of such a database is limited because it contains nothing about the goods or services that were sold and is therefore not very useful as an inventory maintenance tool or even as a basis for sales analysis or market research. It contains only a partial customer identifier.

In general, the card issuer does not participate in the Mondex operation and does not have direct access to the data traffic; so, Mondex has no influence on the use of the scheme and cannot access data flow.

1.3 E-WALLET WITH OBSERVERS

A system which involves off-line e-cash with observers was first mentioned by Brands in Brands (1993). He invented the concept that there has to be a trustee if a purchaser wants to accept a payment without connecting to a bank.

In 1996, the term 'Fair Off-line e-Cash' (FOLC) was presented, which had the idea of eliminating TTP (external) and replacing the trustee with a user.

Petersen & Poupard (2005) developed an efficient payment system with anonymity abolition and TTP. This was the first e-scheme to achieve the off-line prevention from all kinds of extortion-related attacks. Thanks to this we have assumed that any attacks would be of short duration and without any physical involvement by the attacker, as otherwise, no cryptographical protection would be possible. Due to efficiency-enhancing security, the safe realisation of the internet payment scheme has been proven as having a highly efficient payment scheme for electronic wallets.

Stadlerl, Piveteau, & Camenisch (1995) point out that in normal cases the anonymity of the purchaser's payment is guaranteed. This means that the 'judge' is not involved in the operation for whatever legal reasons may be envisioned. However, in certain situations and for the same legal reasons, this anonymity can be waived if ordered by the 'judge'.

This means that the purchaser has no guarantees that he or she will remain anonymous if the 'judge' and the bank decide against that anonymity. So, if a form of 'judge' is to be implemented within the purchaser's secure store, or as it can now be termed a smart device, we are contending only with the problem to protect that 'judge'. In other words, the issue is how secure our 'judge' is (which we can more accurately refer to as our TTP); can it better assure the anonymity of the purchaser?

1.4 THE BEGINNING OF CRYPTO CURRENCY

At the end of 2008, Nakamoto et al. (2008) established decentralised currency based block chains and hash functions - a system which came to be known as Bitcoin. The decentralised currency later came to use this name to describe the entire concept. The Bitcoin payment system which uses a block chain can be referred to as an **off-line, transferable, divisible** digital cash system.

Nevertheless, not just Bitcoin, but also all crypto currencies are faced with the issues that are outlined below:

1. Money laundering (black market, weapons, narcotics, etc);
2. Unstable value (graph);
3. Decentralised currency;
4. 'Forking';
5. Hackers (Bitfinex Hack in September 2016, MT Gox Hack in 2014, MT Gox Hack in 2011, and Silk Road Hack in 2010).

Following this, many other forms of crypto currency also use hash algorithms (such as Scypt, ECDSA, etc.), while others use timestamps - known as POS. As of November 5, 2016, there are a total of 709 crypto currencies available for trade on the online markets, and more than 740 in total, but only 25 of them had market capital that exceeded $10 million.

The general objection towards ZeroCash when comparing it with Bitcoin is that it can facilitate money laundering by circumventing legally binding financial reporting requirements.

As proposed by Sasson et al. (2014), additional protocol modifications can allow users to maintain their anonymity and demonstrate compliance with reporting requirements - which is a definite advantage.

1.5 WHY OFF-LINE PAYMENTS LOSE OUT AGAINST ONLINE PAYMENTS

In their work, Srivastava & Mansell (1998) emphasized the importance of the use of electronic cash in the contemporary world. This does not mean that the banks will continue to play a similar role in the future e-money distribution system, and nor does it mean that their participation guarantees a broad recognition of innovation. This seems to be an attitude that has remained unchanged.

It was noted by Kreft & Adi (2006) when comparing the Mondex and DigiCash systems that Mondex seems to be the best of the Rothwell fifth generation innovation models. Unlike DigiCash, it has used its industry ties and seems to be determined to introduce as many different players and their types as possible. In addition, the product has been designed using a 'lead user' and, in this respect, more accurately predicts unmet needs and market opportunities. DigiCash did not have a 'lead user' in the development of its technology which can be treated as a disadvantage. The fact that Mondex was created by the 'lead user' does not necessarily mean that it will be the market leader. In this sense, if banks are 'lead users' also does not mean that they will

create a commercially successful product or will successfully implement any improvements after such a product has been rolled out.

So, from the user's perspective, there is no difference as to how researchers, bankers, or governments are imagining how e-cash will have to work. The main thing when it comes to making e-cash useful is to make the use of e-cash comfortable for the user.

Nowadays users see that online payments are faster and more accessible than they are in e-cash. Purchasers paying by card in shops (in Lithuania) only amount to about 40% of the total purchasing value.

1.6 TTP SECURITY

1.6.1 The TPM 1.2 chip is not secure from the physical perspective

As Tarnovsky (Messmer, 2010) 'The TPM 1.2 chip is not as secure as the vendor tries to tell you it is'. At the Black Hat conference, Tarnovsky claimed that he could recover the entire crypto engine if it was available in the RFID.

Wilson (2010) pointed out the figure of Tarnovsky who is a researcher at Flylogic Engineering and who has made a business of hacking 'unhackable' chip technology and other hardware. Tarnovsky published his achievements at the Black Hat conference, stating that he had hacked the 'Infineon SLE 66 CL PE' chip which is widely used in computers, gaming systems, identity cards, and other electronics.

Tarnovsky explained his work to overcome chip protection by demonstrating the use of an electron microscope. As he ponted out, 'I'm not saying it was easy, but this technology is not as secure as some vendors would like you to think' (Messmer, 2010).

1.6.2 Physical unclonable functions

The term 'Physical Unclonable Functions' (PUFs) refers to innovative chain primitives that unleash the secret of the physical properties of integrated circuits (ICs). When introducing PUF, the authors Suh & Devadas (2007) highlighted a design that utilised exclusive wiring and transistor delay characteristics which were very different from the systems that were being employed on microchips and they also described how PUF can enable low-cost single IC authentication to generate variable secret keys for cryptographic actions.

PUFs are unique in their physical microstructure. The PUF microstructure depends upon the random physical factors that begin in the production process. These factors are unpredictable and uncontrollable, making it almost impossible to virtually duplicate or clone their structure.

PUFs use random patterns to differentiate chips from one another. These physical unclonable functions also enable you to extract a unique identifier for the chip and to create a unique cryptographic key. With what has been done by Suh & Devadas (2007), and later between 2010 and 2013, it can be seen that PUFs were popular in the use of smartcards in relation to fingerprint-related applications. Now PUF is used in the e-cash systems, microchips, and other forms of technology.

These functions can also be used for generating and managing passwords and strong keys (weak keys as well). Because this technology has a low cost in terms of its implementation, it can replace existing technologies (such as Trusted Platform Modules) for several uses. So, this invention can be very useful for mobile devices which have limited resources, and which need to generate some strong keys for security.

The concept is similar to that of human fingerprints and human biometrics. Thanks to this each and every individual device can carry a unique identifier.

Various works (Cortez, Dargar, Hamdioui, & Schrijen, 2012; Herder, Yu, Koushanfar, & Devadas, 2014) have demonstrated that there are two primary applications for which PUFs are used:

1. low-cost authentication;
2. secure key generation.

So, there are two main PUF programs that are available: authentication and secure key generation. These categories are described as 'strong PUF' and 'weak PUF'. Authentication is usually used by strong PUFs while weak PUFs are used for crypto key storage.

1.7 CONCLUDING REMARKS

1. Electronic cash (e-cash) is a digital version of physical currency. In general, it has the same qualities as physical money (or as many as is possible) and is divisible, anonymous, off-line, untraceable, secure, and transferable.

2. Most divisible, anonymous, untraceable, off-line e-cash systems have a common issue - the data involved expands in terms of its data requirements when transferring e-cash. This is the main drawback of any e-cash system according to Chaum & Pedersen (1993). Additional information is required in order to prevent 'double spending' and to retain characteristics such as divisibility.

3. There is a high number of e-cash systems that sacrifice some properties in return for better performance and better characteristics in comparison to the others.

4. Any new e-payment system has to: be off-line and transferable, use divisible e-cash, theoretically be secure from a plain text attack, and be able to prevent money laundering and 'double spending', etc.

2 METHODOLOGY AND MATHEMATICAL BACKGROUND

In this section we introduce concepts of a digital signature message, authentication codes, the ElGamal signature scheme, the Schnorr identification scheme, and discrete logarithm assumption, the way it were given by (Boneh & Shoup, 2017; Diffie, Diffie, & Hellman, 1976; ElGamal, 1985; Rivest, Shamir, & Adleman, 1978; Schnorr, 1990).

2.1 DIGITAL SIGNATURES

A digital signature (an e-sign) is a mathematical technique that is used to ensure the authenticity and integrity of a digital document, a message, or software. It is the equivalent to a physical signature with some differences. An e-sign is most often used for crypto protocols to ensure security. Further, several instances of their usage are being discussed.

Using e-sign for software. Suppose that a software company (termed here 'the company' for ease of reference) issues software updates for its product. The buyer will download the software update file 'U' from the public site. Before installing 'U' on his or her computer, the buyer will want to check that the file 'U' has actually been issued by the company from which it has been downloaded. To make this easier, the seller adds a short tag to 'U' that is known as a digital signature. Only the company itself can generate a signature for 'U', but anyone can check if a public key is available to them. So, everyone can check the validity of the file 'U'. The short tag on the 'U' file is known as a digital signature. The overall procedure is referred to as a digital signature scheme. This scheme works as follows:

1. The company generates a secret key, PrK, and a public key, PuK, both of which are mathematically related. It keeps the secret key PrK to itself. The company will use the secret key PrK for coding the software it is selling.
2. Now the company can run a signing algorithm by adding (PrK, U) into data. Company's algorithm provides results in the form of σ. Now it can give the pair to the customer (U, σ). The signing algorithm is denoted as S.
3. The purchaser, when receiving the U file, the signature σ, and the public key Puk, checks the validity of U by using the signature σ and the public key PuK. We refer to this as the verification algorithm V. This algorithm outputs only two options - accept or reject.

The digital signature scheme is widely used in areas such as those mentioned by Boneh & Shoup (2017), Rosenberg (2010), and Stadlerl et al. (1995), and in every single software update. For security purposes, we must make it a requirement that a malicious attacker with a PuK cannot generate a valid signature using a fake one.

A digital signature bears some difference from a physical signature in the fact that an e-sign depends on the data that is supplied by 'U'. Unlike a physical signature, every e-sign is different and there are no similarities between e-signs of the same signee.

Message authentication. Suppose, for example, that Bob receives an email from his friend Alice. He really wants to guarantee that the email has been surely sent by her. Digital signatures provide a simple solution: firstly, Alice generates mathematically linked parameters - in the form of a public and secret key pair

(PuK, PrK). She makes PuK publicly available. By sending an email m to Bob, she generates a signature σ on m which is shuffled using her secret key PrK. Then Alice sends the message and the signature to Bob. He receives (m, σ) and ensures that m is from Alice in these steps:

Bob retrieves Alice's public key PuK which is publicly available and, after that, he runs the signature verification algorithm, entering these parameters: the public key, the message, and the signature, all of which were received from Alice. If the algorithm outputs its acceptance, then Bob is guaranteed that message m has been sent by Alice. In that way integrity is ensured.

In the available literature (Boneh & Shoup, 2017) it has been noted that there are more concrete examples of this process, e.g., the domain key-identified mail (DKIM) system which is widely used for every outgoing email that is sent by an organisation (with every email being signed) by placing PuK in the DNS records. DKIM can also be used to prevent spammers.

Certificate of authority. In this final example of the use of digital signatures, we look at their most widely-used case. Most often there is no public directory for the recipient of a message in which to place PuK for receiver of a message. Instead of creating a public directory for access by the recipient, Alice's public key PuK is certified by a TTP (also known as a certificate authority or CA). She first generates a public and private key pair $(PuK \ and \ PrK)$ for a certified public key. Then Alice sends her public key PuK to the CA. The CA has to confirm that the public key PuK belongs to her. After that, the CA signs the message m using its own secret key PrK_{CA} and sends the pair $Cert = (m, \sigma_{CA})$ to Alice. $Cert$ is a certificate for the public key PuK. To verify Alice's public key, Bob firstly obtains a certificate from Alice, and then verifies the CA's signature in the certificate. If the signature is authentic then Bob can be assured that PuK is Alice's public key. The fundamental principle behind the CA's e-signature is to prove to Bob that the message m was warranted by the CA. In order to verify the CA's signature, PuK_{CA} is needed from Bob. Most often the CA's PuK_{CA} are implemented in users' device.

Definition 1. A signature scheme $SiSc = (Gen, S, V)$ is a threesome of efficient algorithms, Gen, S, and V, with these being the generation, signing, and verification algorithms, respectively.

- Gen is a probabilistic algorithm that takes no input. It outputs a pair which consists of a public and a secret key (PuK, PrK), where PuK is referred to as a public key and PrK is referred to as a secret key.
- S is a probabilistic algorithm that outputs a signature $\sigma \overset{R}{\leftarrow} S(PrK, m)$, where PrK is a secret key and m is a message.
- V is a deterministic algorithm that is invoked as $V(PuK, m, \sigma)$. Acceptance or rejection is outputted.
- We require that a signature which is generated by S is always accepted by V. That is, for all of (PuK, PrK) that is outputted by Gen and all messages m, we have: $Pr[V\ (PuK, m, S(PrK, m)\ ) = accept] = 1$.

2.2 MESSAGE AUTHENTICATION CODES

It has been demonstrated in the available literature (Boneh & Shoup, 2017; ElGamal, 1985; Schnorr, 1990) that all of the main and well-known e-cash or crypto systems use a message authentication code. Even more than this, MAC is used in every single e-cash system (Bosselaers et al., 2012; Franklin, Yung, & Center, 1992; Muleravičius et al., 2016; Pailles, 2012). We begin by defining what a message integrity system is based on in terms of a common secret key between the sender and the recipient of the message. For historical reasons, such systems are briefly referred to as message authentication codes or MACs.

We define it the same way as Boneh and Shoup (Boneh & Shoup, 2017) .

Definition 1. A Message Authentication Code system $I = (S, V)$ is a pair of efficient algorithms (V is the verification algorithm and S is the signing algorithm).

- S is a probabilistic algorithm that outputs a hash $t \xleftarrow{R} S(k, m)$, where k is a key and m is a message.
- V is a deterministic algorithm that is invoked as $r \leftarrow V(k, m, t)$ where k is a key, t is a tag, m is a message, and the output r is either acceptance or rejection;
- We require that the hash that is generated by S is always accepted by V; So the MAC must satisfy the following accuracy property: for all keys k and all messages m, $\Pr\left[V\left(k, m, S(k, m)\right) = accept\right] = 1$.

We say that $I = (S, V)$ is defined over (K, M, T), where T is a finite tag space, K is a finite key space, and M is a finite message space.

For algorithm V outputs its acceptance for a message and a hash pair (m, t), we say that t is a valid hash for m under key k, or that (m, t) is a valid pair under k. Of course, we want Message Authentication Code systems that have a hash that is as short as possible so that hashing is minimal.

Then, signature algorithm *S* is probabilistic, the verification algorithm is defined as shown below:

$$V(k, m, t) = \begin{cases} accept & if\ S(k, m) = t \\ reject & otherwise \end{cases}$$

One probabilistic feature of a Message Authentication Code system is that it has unique tags: for a given k key and the specified message m, there is a unique valid hash for m by k. Not all MAC systems will have such a simple design, some have a random selection algorithm, so in a given k and m message, the output $S(k, m)$ can be one of many possible valid tags and the validation algorithm works in a different way. Such random-selection Message Authentication Code systems are not necessary for security, but they can increase efficiency.

We will require a highly hostile environment in order to create MACs that remain secure in various applications. Because most real-world systems which are using MACs work in less hostile settings.

Suppose that an adversary is attacking a MAC system $I = (S, M)$. Let k be a randomly selected MAC key which is unknown to the attacker. We let this user request tags $t = S(k, m)$ for an arbitrary message m of his or her choice. This attack is referred to as *a chosen message attack*. The malicious user is able to collect a lot of

valid message hash pairs. In this way we give the attacker considerable power to reduce the security of the encryption scheme.

2.3 ELGAMAL SIGNATURE SCHEME

In this section, we describe the ElGamal signature scheme as it was first published (ElGamal, 1985). Let m be a message to be signed, where $0 \leq m \leq p-1$, and where p is a strong prime number. The public file consists of a public key $PuK = a^s mod p$ for each user. In order to sign the message m , user 'A' should be able to use the secret key PrK_A to find a signature for m in such a way that all users will be able verify the authenticity of the e-signature using the public key PuK_A (together with a and p). Also, nobody can forge a signature without knowing the secret PrK_A.

The signature for m is the pair $(r, s), 0 \leq r, s < p-1,$ chosen so that the equation $a^m = PuK^r r^s mod p$ is satisfied.

Firstly, a random number q is selected, and is uniformly distributed between $zero$ and $p-1$, so that $\gcd\ (q, p-1) = 1$.

Secondly, the following is computed:

$$r = a^q mod\ p$$

And is written as:

$$a^m = a^{PrK \cdot r} a^{qs} mod\ p$$

Which can be solved for s using:

$$m = PrK \cdot r + qs\ mod(p-1)$$

This equation has a solution for s if q is chosen so that $\gcd(q, p-1) = 1$.

The verification procedure can be implemented by giving m, r, and s. In order to verify the authenticity of the signature the verification process must compute both sides of $a^m = PuK^r r^s mod\ p$ and check that they are equal.

2.4 SCHNORR IDENTIFICATION SCHEME

Here we describe an identification scheme which is known as Schnorr identification, named after its inventor, Schnorr (Schnorr, 1990). This protocol can be proved as being secure against eavesdropping attacks, assuming that the discrete logarithm problem is infeasible.

Let G_q be a cyclic group of prime numbers q with a generator $G \in G_q$. Suppose that the prover '*P'* has a secret key $PrK = \alpha \in Z_q$, and the mathematically linked public key is $PuK = u = G^\alpha \in G_q$. In order to prove his or her identity to the verifier '*V'*, '*P'* wants to establish with '*V'* that he or she knows α. The simplest way to do this would be for '*P'* to simply send α to '*V'*, but it cannot be done in such a way that is known only to '*P'*. The Schnorr protocol is essentially similar to the MAC protocol with a hash function. The main idea behind the Schnorr identification protocol is an intelligently designed interactive protocol that allows '*P'* to prove to '*V'* that he or she knows the discrete logarithm of u (by base of G) without sending this value to '*V'*.

Direct attacks: using no information from any sources other than that which is publicly available, the attacker must somehow impersonate the verifier by acting as the prover.

Let C be a subset of Z_q. Then Schnorr's identification protocol is $I_{Sch} = (G, 'P', 'V')$, where:

1. The key generation algorithm Gen runs as follows:

$\alpha \overset{R}{\leftarrow} Z_q, u \leftarrow G^{\alpha}$

The public key is $PuK = u$, and the secret key is $PrK = \alpha$

2. The protocol between *'P'* and '*V'* runs as follows, where the prover '*P'* is initialised with $PrK = \alpha$, and the verifier '*V'* is initialised with $PuK = u$:

'*P'* computes $\alpha_t \overset{R}{\leftarrow} Z_q, u_t \leftarrow G^{\alpha_t}$, and sends u_t to '*V'*;

3. *'V'* computes the challenge $c \overset{R}{\leftarrow} C$, and sends c to '*P'*;

4. *'P'* computes $\alpha_z \overset{R}{\leftarrow} \alpha_t + \alpha c \in Z_q$ and sends α_z back to '*V'*;

5. *'V'* checks whether $G^{\alpha_z} = u_t \cdot u^c$; if so then '*V'* outputs acceptance, otherwise '*V'* outputs rejection.

Figure 1 serves to illustrate the protocol:

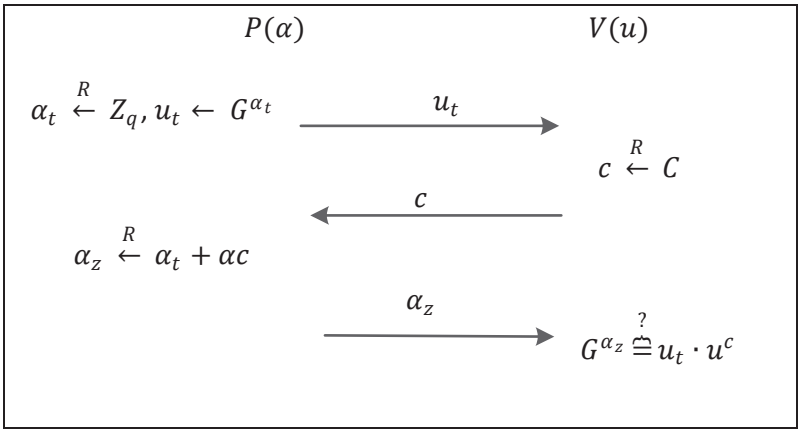

Figure 1 Schnorr's identification protocol

A communication between *'P'* and *'V'* generates a conversation $(u_t, c, \alpha_z) \in G_q \times C \times Z_q$. The conversation is accepted for u if the verifier's 'V' check passes, i.e., if $G^{\alpha_z} = u_t \cdot u^c$. Interaction between '*P'* and '*V* ' always generates an acceptance if $u_t = G^{\alpha_t}$ and $\alpha_z = \alpha_t + \alpha c$, then $G^{\alpha_t} = G^{\alpha_t + \alpha c} = G^{\alpha_t} \cdot (G^{\alpha_t})^c = u_t \cdot u^c$. The Schnorr's identification protocol satisfies the essential correctness requirement that any identification protocol must contribute as described in Boneh & Shoup (2017), Rosenberg (2010), and Schnorr (1990).

Various works (Boneh & Shoup, 2017; Rosenberg, 2010; Schnorr, 1990) proved that Schnorr's protocol is secure against eavesdropping attacks.

It is demonstrated in Boneh & Shoup (2017) that the Schnorr's identification protocol is secure against direct attacks by the fact that generation algorithm Gen holds the discrete logarithm assumption.

The Schnorr identification protocol is secure against direct attacks and it is called 'honest verifier zero knowledge'. Schnorr's signature scheme is also secure against eavesdropping attacks.

2.5 DISCRETE LOGARITHM AND OTHER SECURITY ASSUMPTIONS

We state below the DL and relevant assumptions, the way it were shown by Boneh & Shoup (2017) and Schnorr (1990).

Discrete logarithm definition. Let G be a cyclic group of prime order q generated by $g \in G$. For a given adversary '*A'*, define the following:

- The challenger computes: $\alpha \overset{R}{\leftarrow} Z_q, u \leftarrow g^{\alpha}$, and provides the value u to the adversary.
- The adversary outputs something like $\hat{\alpha} \in Z_q$

We define A's advantage in solving the discrete logarithm problem for G, denoted $DLadv[A, G]$, as the probability that $\hat{\alpha} = \alpha$.

Discrete logarithm assumption. If, for all efficient adversaries the quantity $DLadv[A, G]$ is negligible, then we state that the discrete logarithm assumption holds true for G.

α is a solution for the problem of the discrete logarithm problem that we state as g^{α}. The discrete logarithm assumption declares that there is no efficient algorithm that can effectively solve the discrete logarithm problem in a reasonable amount of time.

Note that the discrete logarithm assumption is defined by group G_q and the generator $G \in G_q$.

Now we can state the computational Diffie-Hellman assumption (Diffie et al., 1976).

Computational Diffie-Hellman assumption. Let G_q be a cyclic group of prime order q generated by $G \in G_q$. For a given adversary '*A'*, assumption runs as follows:

- The challenger computes

$$\alpha, \beta \overset{R}{\leftarrow} Z_q, u \leftarrow G^{\alpha}, v \leftarrow G^{\beta}, w \leftarrow G^{\alpha\beta}$$

and gives the pair (u, v) to the adversary.

- The adversary outputs something like $\hat{w} \in G_q$

We define A's advantage in solving the computational Diffie-Hellman problem for G_q, denoted as $CDHadv[A, G_q]$, as the probability that $\hat{w} = w$.

Computational Diffie-Hellman assumption. The computational Diffie-Hellman (CDH) assumption (Diffie et al., 1976) holds for G_q if, for all efficient adversaries '*A'*, the quantity $CDHadv[A, G_q]$ is negligible.

We call the pair (G^{α}, G^{β}) a 'Computational Diffie-Hellman' problem, and that $G^{\alpha\beta}$ is a solution to this problem by the authors (Diffie et al., 1976). We assume that the description of G_q includes its order q and a generator G. The CDH assumption asserts that there is no efficient algorithm that can effectively solve the CDH problem in a reasonable amount of time.

An interesting point of the 'Computational Diffie-Hellman' problem is the fact that there is no common and effective algorithm to even recognise solutions to the CDH problem, that is, given an instance (G, G^{β}) of the CDH problem, and a group element $\hat{w}$, to determine if $\hat{w}$ is a solution to the given problem instance.

We use these assumptions in our system security proof.

2.6 CONCLUDING REMARKS

1. In this section, we provide an analysis and an overview of the cryptographic primitives that are used later. These primitives include an electronic signature which is the main part of all of the cryptography elements that is required in order to ensure data integrity and authenticity. Message authentication codes are used to authenticate data and ensure integrity.

2. There are two main schemes that are being reviewed: the ElGamal signature scheme will be used as the primary cryptographic primitive for rapid information transfer, security, and anonymity, and the Schnorr identification scheme will be used to ensure the reliability and anonymity of information. Also, some security assumptions are reviewed.

3 A NEW E-CASH SCHEME

3.1 AN ABSTRACT E-CASH CIRCULATION SCHEME

The considered e-cash system consists of the following parties: the bank ($\boldsymbol{B}$), the purchaser ($\boldsymbol{P}$), the vendor ($\boldsymbol{V}$), the purchaser's observer ($\boldsymbol{O_P}$), the vendor's observer ($\boldsymbol{O_V}$) and the attacker ($\boldsymbol{Zoe}$).

The purchaser can withdraw e-coins from his or her observer ($\boldsymbol{O_P}$) and spend them with various vendors. The vendor also has his or her own observer ($\boldsymbol{O_V}$); the vendor deposits the e-coins that he or she gets from the purchaser into their own observer. In addition, following the deposit protocol, the vendor can transfer money between further users; in other words, the vendor can become a purchaser by achieving the property of transferability.

The observer is needed to achieve an off-line requirement. It acts as a bank by signing against funds that have been spent by the purchaser, there by legalising any related transactions.

Typically, we consider the purchaser and the vendor to be interchangeable since they both need to be able to carry out the process of making deposits and withdrawals.

The vendor can deposit his or her e-coins into the bank. If the vendor tries to deposit the same e-coin with the same time stamp, then 'double spending' can be detected. The bank uses the two e-coins to compute the identity of the user who is acting fraudulently.

All e-cash systems mainly consist of the same set of four protocols. Some protocols, such as registration, withdrawal, payment, and deposit, are universal. So, in other sections, we will take a deeper look into these protocols.

Before that, in Table 1 we present some notifications of system parameters and functions:

Table 1 Notifications for the e-cash system

Parameter/functions	Description
q, p	Large prime numbers so that p satisfies the strong prime property $p = 2q + 1$.
G_q	Cyclic subgroup where $\boldsymbol{G_q} = \{g^i \| i = \overline{1,q}\}$ and $g^q \equiv 1 \mathrm{mod} q$
G	A generator of multiplicative group $\mathbb{Z}_p^*$.
h_i	The value of a hash function.
$Sig_{ElG}^{X}(m)$	The ElGamal signature function, where m and X correspond to the message to be signed and the ElGamal private key of the signee.
$Ver_{ElG}^{A}(s, m)$	The ElGamal signature verification function, where m, s, and A correspond to: the message, the signature on the message, and the ElGamal public key of the signee.
i	The serial number for a transaction.
$PrK_P = x_P$, $PuK_P = \{G, A_P = G^{x_P}\}$	The 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, and these ensure the anonymity property for the proposed e-cash scheme.
$PrK_O = x_O$, $PuK_O = \{G, A_O = G^{x_O}, A_P^{Id_P}\}$	The observer's private and public keys.
Id_P, Id_V	The unique identification number for the purchaser's and vendor's observer chip, respectively
m_i	The amount of money to be spent by the purchaser.
$\widetilde{m}_i$	The actual price of the products to be bought by the purchaser.
t_i	The time instance for an e-cash withdrawal.
$m_i \| \| t_i$	The concatenation of the amount and the time instance.
t_{w0}, t_{p0}, t_{d0}	The time instance for a previous e-cash withdrawal protocol (payment and/or deposit).
m_{max}^P, m_{max}^V	The amount of money in the e-wallet of the purchaser and the vendor, respectively.
$\xi_i^{(1)}, \xi_i^{(2)}$	Random values for $\mathbb{Z}_q^*$ for the Schnorr interactive identification protocol.

3.1.1 Registration protocol

Let us assume that the purchaser is a new client of the bank and is willing to use the e-cash service that the bank provides. According to the ElGamal signature (ElGamal, 1985) and the Schnorr identification scheme (Schnorr, 1990), the bank will generate the following private and public keys for the purchaser:

PrK_P, PuK_P

The bank will also supply the purchaser - via a secure channel – his or her own observer (implemented in the chip) with the following keys hidden inside it:

PrK_O, PuK_O

The purchaser and the vendor use the same method to get their own keys. In other words, they establish an account with the bank by getting public and private keys.

Note that the observer (realised in the chip) is supplied from bank to the purchaser and the vendor, which can be implemented via a mobile phone, a tablet, a computer, or other mobile device.

3.1.2 Withdrawal protocol

This is an interactive protocol that allows a purchaser withdraw an e-coin from his or her bank account (the trustee that is implemented on the purchaser's device). Every withdrawal protocol begins with the purchaser's generation of a time instance and monetary amount and continues with sending these to the observer. After that, the observer checks the time instance and the balance left in the e-wallet. If there are enough funds, then a withdrawal can be made. The observer-generated random values are required to be able to carry out the Schnorr identification scheme. By signing these values, the observer provides the following: the purchaser's public key which is concluded from an amount of money that is concatenated with a time instance and multiplied with the Schnorr identification scheme parameter; the purchaser's public key which is get from an amount of money that is concatenated with a time instance; the purchaser's public key which is get from the purchaser's identifier; and the purchaser's public key which is concluded from an amount of money that is concatenated with a time instance that is multiplied by the purchaser's identifier and is multiplied with the Schnorr identification scheme's parameter. All of this is needed to prevent 'double spending' and to carry out other security preventions.

The observer may also log some information about the purchaser, the previous time instance, and the new monetary balance.

Note that we have assumed that no one other than the purchaser can make a request to their observer.

Figure 2 serves to illustrate the protocol:

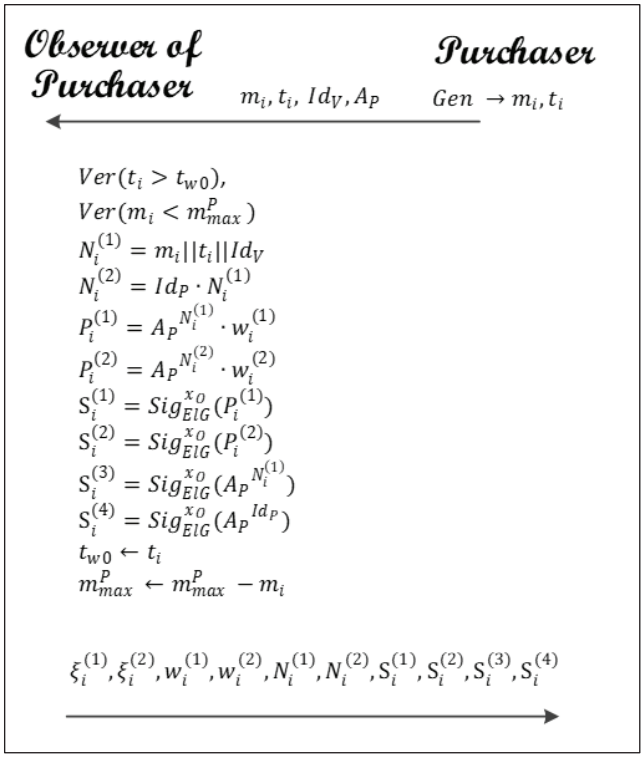

Figure 2 Withdrawal diagram

3.1.3 Payment protocol

This is a protocol in which the purchaser gives the vendor an e-coin and proves that it has been signed by the bank (or, on the other hand, the observer). As was previously noted regarding the withdrawal protocol, the required amount was taken from the observer in order to pay the vendor. After the purchaser gains his or her values from the trustee, the purchaser can begin the payment protocol process by sending the following values to the vendor: the amount of money, the time instance, the purchaser's public value, the purchaser's identifier shuffled together with his or her public key, the parameters for the Schnorr identification scheme, and all of the observer's signatures.

Firstly, the vendor attempts to check the sum of money and the time instance. When continuing the payment protocol, all four signatures are verified if they meet the requirements, and then the vendor generates a random value and sends it to the purchaser. Now the purchaser has to respond to the vendor's 'challenge' by computing certain values (see Figure 3 below for details) and sends those back along the same path. The vendor can verify these response values and can be guaranteed that the purchaser is not acting maliciously. The next stage involves the vendor having to prove that the payment has actually taken place. The vendor sends the sum of money and the time instance with the signature which was generated by the purchaser's observer to his or her own observer. The vendor's observer creates a unique signature of its own which is forwarded to the purchaser to ensure that the money that is now in

the vendor's wallet actually came from the purchaser. Finally, the vendor renews the time instance in his or her e-wallet.

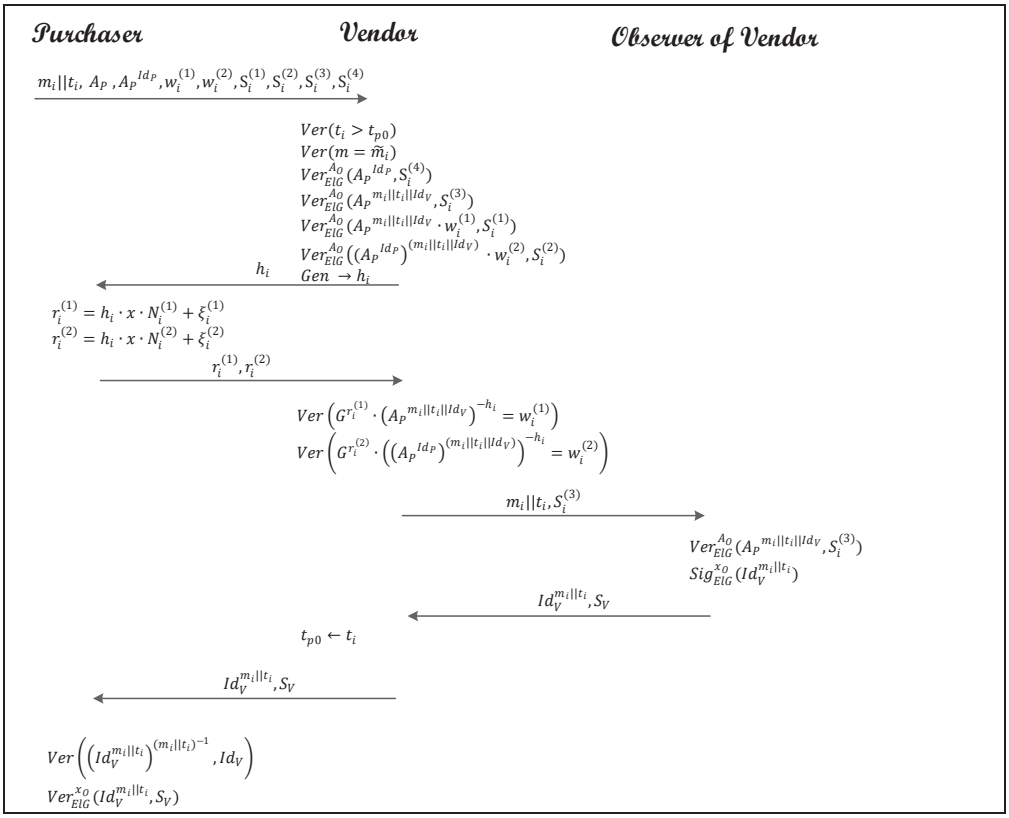

Figure 3 **Payment diagram**

3.1.4 Deposit protocol

After the vendor has received his or her values from the purchaser, he or she can begin the deposit protocol by sending the following values to the vendor's observer: the amount of money involved in the transaction, the time instance, the purchaser's public value, the purchaser's identifier shuffled together with his or her public key, the Schnorr identification schemes parameters, and all signatures which were received from the purchaser.

Firstly, the observer tries to check the time instance. Then, continuing the deposit protocol, the observer verifies all four signatures to see whether they meet the requirements, and then the observer updates the monetary balance and the time instance. Figure 4 serves to illustrate the protocol:

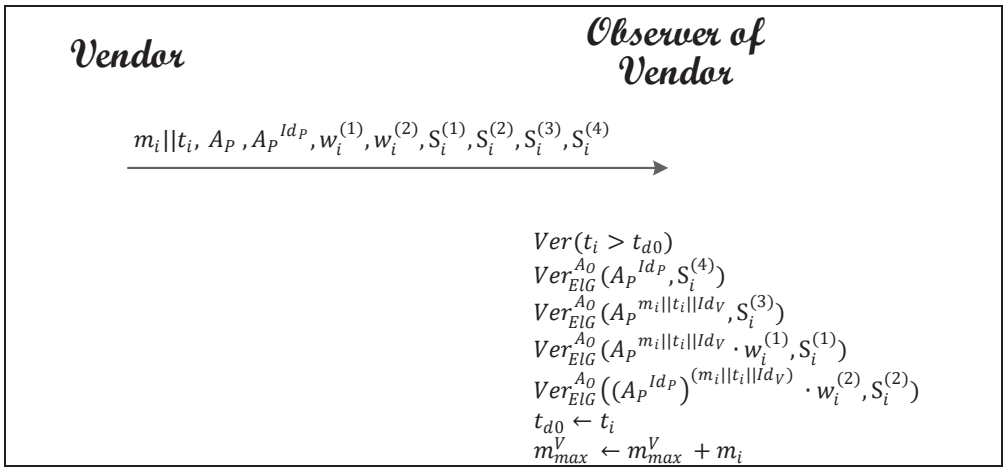

Figure 4 **Deposit diagram**

3.2 PROTOCOL REALISATION SCHEME

To start with, the bank generates a strong prime p, i.e. $p = 2q + 1$, where q is a prime number, and an element g, satisfies the congruence $g^q \equiv 1 \mathrm{mod} q$. The practical way of generating this element is to find a generator for the initial group $\boldsymbol{Z_p}$ and then to square it. The element g can be used to generate a cyclic subgroup $\boldsymbol{G_q} = \{g^i | i = \overline{1, q}\}$ which is called a Sylow subgroup. The bank also selects a hash function H such that $H: \{0,1\}^* \rightarrow \boldsymbol{G_q}$.

Let us assume that the purchaser is a new client of the bank and is willing to use the e-cash service that is provided by the bank. According to the ElGamal signature and the Schnorr identification schemes, the bank will generate the following private and public keys for the purchaser:

$PrK_P = x_P, PuK_P = \{G, A_P = G^{x_P}\}$

The bank also supplies the purchaser with his or her trustee observer and generates the following information for the purchaser's observer:

$PrK_O = x_O, PuK_O = \{G, A_O = G^{x_O}, A_P^{Id_P}\}$

In this, $1 < x_P, x_O < q - 1$ and Id_P is the purchaser's identifier, i.e. a unique integer which is assigned to each of the bank's clients. Note that A_P is a public parameter which is associated with the purchaser. The term $A_P^{Id_P}$ is certificated by the bank.

The signature on the message $m \in \boldsymbol{G_q}$ is computed using the ElGamal signature function $Sig_{ElG}^X(\,)$, where x denotes the private key of the signee:

$\mathrm{S}_m = Sig_{ElG}^x(m) = \{R, S\} = \{G^k mod\ p, k^{-1}(h(m) - xR) mod\ q\}$,

where k is a random secret non-zero integer that is less than q.

The verification of the signature S_m on the message, m is carried out using the verification function $Ver_{ElG}^A(\,)$, where A is a public key for a signee:

$$Ver_{ElG}^A(\mathrm{S}_m, m) = \begin{cases} True, if\ R \in \boldsymbol{G_q}\ and\ A^R R^S = G^{h(m)} mod\ p \\ False, \quad \text{otherwise} \end{cases}$$

The Schnorr interactive identification protocol is executed between the purchaser and the vendor and consists of four steps:

1. the purchaser chooses $\xi \in \boldsymbol{Z_q}$randomly and sends $W = G^{\xi}$ to the vendor;
2. the vendor sends a randomly-generated challenge $h \in \boldsymbol{Z_q}$ to the purchaser;
3. the purchaser sends the obtained response $r = \xi + xh$ to the vendor;
4. the vendor accepts the response if $G^r \equiv \mathrm{W}A^h \mathrm{mod\ p}$.

3.2.1 The e-cash withdrawal protocol

Assume that the purchaser intends to purchase goods from the vendor and wants to pay the sum of m_i in e-cash at the time instance t_i. The purchaser generates his or her temporary keys $PrK_P = x_P$ and $PuK_P = \{G, A_P = G^{x_P}\}$ and sends his or her public key A_P to the observer together with the required sum, the time at which the request is being made t_i, and the vendor's identity indicator Id_V.

1. The purchaser sends the sum m_i, which is the amount he or she intends to spend, along with the time instance t_i to his or her observer:

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

Note that we have assumed that only the purchaser can make a request of his or her own observer.

2. Upon receiving the required data from the purchaser, the observer carries out the following actions:

The accuracy of the received time instance t_i is verified and a check is made on whether the desired sum is available to spend, as shown here:

$Ver(t_i > t_{w0})$,
$Ver(m_i < m_{max}^P)$

where t_{w0} denotes a time instance of the last withdrawal and m_{max}^P is the currently-available amount in the e-wallet.

3. The observer generates random integers $\xi_i^{(1)}, \xi_i^{(2)} \in \boldsymbol{Z_q}$

4. And computes Schnorr identification protocol values $W_i^{(1)} = G^{\xi_i^{(1)}}$ and $W_i^{(2)} = G^{\xi_i^{(2)}}$.

5. This is followed by generating the values $N_i^{(1)}, N_i^{(2)}, P_i^{(1)}, P_i^{(2)}$, and signing the values $P_i^{(1)}, P_i^{(2)}, {A_P}^{N_i^{(1)}}, {A_P}^{Id_P}$:

$N_i^{(1)} = m_i || t_i || Id_V$
$N_i^{(2)} = Id_P \cdot N_i^{(1)}$
$P_i^{(1)} = {A_P}^{N_i^{(1)}} \cdot w_i^{(1)}$
$P_i^{(2)} = {A_P}^{N_i^{(2)}} \cdot w_i^{(2)}$
$\mathrm{S}_i^{(1)} = Sig_{ElG}^{x_O}(P_i^{(1)})$
$\mathrm{S}_i^{(2)} = Sig_{ElG}^{x_O}(P_i^{(2)})$
$\mathrm{S}_i^{(3)} = Sig_{ElG}^{x_O}({A_P}^{N_i^{(1)}})$
$\mathrm{S}_i^{(4)} = Sig_{ElG}^{x_O}({A_P}^{Id_P})$

where || denotes the concatenation of the sum m_i and time instance t_i. The result $N_i^{(1)}$ is represented by a single integer.

6. The observer saves the received time instance t_i as the time at which the previous withdrawal was made:

$t_{w0} \leftarrow t_i$

7. The observer saves the amount that has been received and subtracts that amount from the total funds in the purchaser's e-wallet:

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

8. Finally, the observer sends the following data to the purchaser:

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

Following the completion of the e-cash withdrawal protocol, the payment protocol can be executed.

3.2.2 The e-cash payment protocol

The Schnorr interactive identification protocol is embedded into the payment protocol in order that the purchaser can prove his or her identity to the vendor.

1. First of all, the purchaser sends the vendor the payment amount m_i, which is the amount that the purchaser intends to spend, along with the time instance t_i and the signatures $S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}$ which has been received from the observer, and the Schnorr protocol values $W_i^{(1)}, W_i^{(2)}$:

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

2. The vendor verifies the accuracy of the received time instance t_i, and checks that the amount received is equal to the actual amount m_i that was expected to be received:

$Ver(t_i > t_{p0})$

$Ver(m = \widetilde{m}_i)$

3. The vendor verifies the signatures to ensure that the received data has not been forged in any way:

$Ver_{ElG}^{A_O}({A_P}^{Id_P}, S_i^{(4)})$

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

$Ver_{ElG}^{A_O}({A_P}^{m_i||t_i||Id_V} \cdot w_i^{(1)}, S_i^{(1)})$

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

where t_{p0} is a time instance of the previous payment. The protocol is aborted if any failures occur at this step, since that will mean that the vendor has discovered a forgery in the data received. The purchaser will receive an error message that indicates the problem. The purchaser can no longer use the data for this transaction to execute any new payments.

4. The vendor wants to be sure that the user with whom he or she is communicating is the actual purchaser and hence initiates the Schnorr identification protocol. Firstly, the vendor will generate a random number $h_i \in \boldsymbol{Z_q}$:

$$Gen \rightarrow h_i$$

5. And afterwards, the vendor will send out a random challenge h_i to the purchaser:

$$V \xrightarrow{h_i} P$$

6. Upon receiving the challenge, the purchaser computes his or her Schnorr's protocol values:

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

the response values, $r_i^{(1)}$ and $r_i^{(2)}$, are forwarded to the vendor:

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

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

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

The protocol is aborted if any failures occur at this step. If that is the case, then the purchaser will receive an error message which indicates the identification problem. The purchaser may try to initialise the payment protocol again if the vendor allows this option. Otherwise, the data for this transaction can no longer be used.

If no failures occur during these steps, then the payment has been properly and fully completed. However, the vendor now has to confirm that the payment took place.

8. The vendor sends the payment amount m_i, the time instance t_i, and the signature $S_i^{(3)}$ to the vendor's observer:

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

9. The vendor's observer confirms the validity of the date received by verifying the signature $S_i^{(3)}$:

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

If this verification fails, then the observer blocks the transaction, i.e., the vendor is no longer able to deposit the amount m_i into the vendor's e-wallet.

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

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

11. The vendor sends the following data to the purchaser for verification:

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

12. The purchaser carries out the following actions to ensure that he or she is not dealing with a malicious vendor:

a. Raises $Id_V^{m_i||t_i}$ to power $(m_i||t_i)^{-1}$ and then compares the result to Id_V. In other words:

$$Ver\left(\left(Id_V^{m_i||t_i}\right)^{(m_i||t_i)^{-1}}, Id_V\right)$$

b. Verifies the time instance and signature S_V:

$Ver_{ElG}^{x_O}(Id_V^{m_i||t_i}, S_V)$.

If verification is successful, then the transaction is successfully completed, and both parties will receive a message confirming this result. Otherwise, the transaction is cancelled, and the purchaser may turn to the bank in electronic or physical form to restore the balance in his or her wallet. Both parties will receive error messages in this case.

13. The vendor saves the received time value as the time of the purchaser's most recent payment:

$t_{p0} \leftarrow t_i$

After the successful completion of the payment protocol, the vendor will have received the amount for the total value of goods and delivered them to the purchaser in electronic or physical form. Otherwise, if any errors have occurred, the purchaser is reported to the bank.

3.2.3 The e-cash deposit protocol

1. The vendor sends the following data which has been received from the purchaser to the vendor's observer:

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

2. Upon receiving the data, the observer verifies the accuracy of the received time instance t_i:

$Ver(t_i > t_{d0})$

Note that the observer O_V does not verify that the deposit is taking place at the current time since this protocol can be executed at any time. Any failure in this step results in an error message which indicates that the transfer of the funds has already taken place sometime before. The deposit protocol is aborted.

3. The vendor's observer O_V verifies the accuracy of the received signatures:

$Ver_{ElG}^{A_O}(A_P{}^{Id_P}, S_i^{(4)})$

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

$Ver_{ElG}^{A_O}(A_P{}^{m_i||t_i||Id_V} \cdot w_i^{(1)}, S_i^{(1)})$

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

4. The vendor's observer O_V renews a time instance for the most recent deposit:

$t_{d0} \leftarrow t_i$

5. The vendor's observer O_V updates the vendor's wallet balance:

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

3.2.4 Preventing 'double spending'

In the case of 'double spending', the purchaser's unique identification number Id_P can be revealed as explained below.

Since the purchaser has spent the same amount of money twice, the vendor will have been sent the following values:

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

$$r'_i{}^{(1)} = h'_i \cdot x_P \cdot N_i^{(1)} + \xi_i^{(1)}$$
$$r'_i{}^{(2)} = h'_i \cdot x_P \cdot N_i^{(2)} + \xi_i^{(2)}$$

where h_i and h'_i are random Schnorr protocol values (Schnorr, 1990) which have been generated by the vendor during the first and the second payment protocols, respectively.

Then the purchaser's identity Id_P can be computed in the following way:

$$\frac{r_i^{(2)} - r'_i{}^{(2)}}{r_i^{(1)} - r'_i{}^{(1)}} = \frac{(h_i - h'_i) \cdot x_P \cdot N_i^{(2)}}{(h_i - h'_i) \cdot x_P \cdot N_i^{(1)}} = \frac{Id_P \cdot N_i^{(1)}}{N_i^{(1)}} = Id_P$$

It is important to note that the latter identity is valid, since all actions are carried out as a prime modulus q, and hence a non-zero element $r_i^{(1)} - r'_i{}^{(1)}$ is invertible since the algebraic structure $\boldsymbol{Z_q}$ is a field.

3.3 CONCLUDING REMARKS

1. The ElGamal signature scheme (ElGamal, 1985) was used for providing authentification for the e-cash system by using the bank's implemented chip as a trustee. Additionally, it carries out a confidence check between the purchaser and the vendor. But its drawback is that it severely limits any anonymity between the purchaser and the bank.

2. Data integrity and authentication between the purchaser and the vendor is guaranteed by the Schnorr's identification scheme (Schnorr, 1990). Non-repudiation is carried out by combining Schnorr's and ElGamal systems.

3. The system is partially anonymous, i.e. anonymity for the purchaser against the vendor is provided anonymity for the purchaser against the bank is removed. However, in the case of any 'double spending', the identity of potential malicious purchasers will be revealed. Hence the purchaser cannot forge any data in order to carry out the act of 'double spending' and remain undetected. Analogously, the vendor cannot carry out 'double spending' in terms of e-cash funds that have been received during the deposit protocol process.

4. Any e-cash amount can be increased or decreased, and the size of any related information will not expand after payment has been made.

5. The system consists of registration protocol and three main protocols: withdrawal, which has eight steps of its own; payment, which has thirteen steps; and a further five steps which involve completing the deposit protocol.

6. The system satisfies some of the important properties: divisibility, in that payments do not require a return; off-line payment, in that no connection with the bank is required; prevention of 'double spending', so that if someone attempts to 'double spend' or even 'double deposit', their identity will instantly be revealed; untraceability of legal payments – they cannot be revealed unless they are fraudulent; no data expansion when transferred as the e-wallet's size is not dependent upon the payment amount for which the maximum size is 24 bits; transferability, which means that, following completion of the payment, the vendor can become a purchaser and make a payment of its own as is outlined in section 3.

7. The system does not support anonymity between the purchaser and the bank. Since the bank supplies the TTP and Id for users, this means that it can also track all the payments.

4 ANALYSIS, SECURITY AND A DIGITAL SIMULATION FOR THE E-CASH SYSTEM

4.1 ADVERSARY MODEL AND SECURITY ANALYSIS

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

1. An attack by a 'Malicious Purchaser' (**MP**):

(a) 'Double spending', i.e., using the same data to purchase goods from the vendor more than once;

(b) Forging transaction data, i.e., faking the payment amount, the time instance, and any data that is sent to the vendor. There are two alternatives available for such an attack: spend less money than is demanded by the vendor (thereby forging the payment amount) or presenting a previous transaction as a new one (thereby forging the time instance), this means, carrying out the act of 'double spending' by means of forgery.

2. A 'Man in the Middle' attack (**MitM**):

(a) The purchaser carries out the act of an impersonation by faking his or her identity Id_P, i.e., the act of using the e-wallet of a legitimate purchaser to acquire goods for oneself;

(b) Impersonating a vendor by faking one's identity Id_V , i.e., the act of acquiring and depositing funds that are meant for a legitimate vendor.

3. An attack by a 'Malicious Vendor' (**MV**):

(a) Carrying out a 'double deposit', i.e., the act of using the same data to increase the balance in the vendor's e-wallet more than once;

(b) A denial of payment and refusing to send out the goods, i.e,. the act of keeping the purchaser's money for oneself without delivering the goods by denying that the payment for the goods has been received;

(c) Forging the transaction data, i.e., the act of faking a payment amount, a time instance, and any data that is received from an honest purchaser. There are two alternatives that can take place as part of this form of attack: depositing more money than has been received from the purchaser (thereby forging the payment amount) or presenting a previous transaction as a new one (thereby forging the time instance), this means carrying out the act of producing a 'double deposit' by means of forgery.

4.1.1 Analysis of an attack by a 'Malicious Purchaser'

To start off our analysis, we first focus on the **MP** attack scenario, i.e., actions which can be executed by a dishonest purchaser so that he or she can benefit from a deal which is concluded with an honest vendor. Preventing 'double spending' is guaranteed by the Schnorr identification, that is, upon receiving the same transaction twice, the vendor can reveal the purchaser's identity by calculating the following expression:

$$\frac{r_i^{(2)} - r'_i{}^{(2)}}{r_i^{(1)} - r'_i{}^{(1)}} = Id_P \qquad (1)$$

where responses $r_i^{(1)}$ and $r_i^{(2)}$ were received during the first sending process for the transaction, whereas responses $r'_i{}^{(1)}$ and $r'_i{}^{(2)}$ were received during the second sending process for the same transaction.

The claim of proof, as in the case studied in (E. Sakalauskas, Timofejeva, Michalkovič, & Muleravičius, 2018), is due to the fact that signatures are secure against the 'chosen message attack'. The data cannot be forged since the purchaser has no access to the purchaser observer's private key x_O, i.e., the signatures $\mathrm{S}_i^{(1)} = Sig_{ElG}^{x_O}(P_i^{(1)})$, $\mathrm{S}_i^{(2)} = Sig_{ElG}^{x_O}(P_i^{(2)})$, $\mathrm{S}_i^{(3)} = Sig_{ElG}^{x_O}({A_P}^{N_i^{(1)}})$, as well as any data that has been signed to remain intact. Since A_P is a generator of the group G_q, no forgery of $N_i^{(1)}$ is possible, which also implies that the time instance t_i cannot be altered since the price of the desired goods m_i cannot be affected by the purchaser. Furthermore, forging $N_i^{(2)}$ is impossible due to the following facts:

1. $N_i^{(1)}$ and $N_i^{(2)}$ are mathematically linked;
2. The public key ${A_P}^{Id_P}$ is certificated and hence Id_P cannot be forged;
3. Z_q is a field and hence $N_i^{(1)}$ is invertible.

Non-forged values of $N_i^{(1)}, N_i^{(2)}, P_i^{(1)}, P_i^{(2)}$ now imply the correct values of $W_i^{(1)}, W_i^{(2)}$ since G_q is a group and hence ${A_P}^{N_i^{(1)}}$ and ${A_P}^{N_i^{(2)}}$ are both invertible.

To consider any forgery of the data that is provided by the **MP**, we recall the data that was sent to the vendor during the payment protocol process:

$$P \xrightarrow{m_i || t_i,\, A_P,\, {A_P}^{Id_P},\, w_i^{(1)},\, w_i^{(2)},\, \mathrm{S}_i^{(1)},\, \mathrm{S}_i^{(2)},\, \mathrm{S}_i^{(3)},\, \mathrm{S}_i^{(4)}} V$$

Since this data involves signatures, in order to fake his or her identity, the purchaser may try to forge all of the signatures that are sent during this step. However, since the purchaser is not able to generate signatures by himself or herself (only the purchaser's observer can do this), any forgery of signature requires the purchaser to deal with the discrete logarithm problem as stated in Theorem 20 of Pointcheval & Stern (2000), while considering the modified ElGamal signature scheme's security levels which help to prevent an adaptive adversary. Based on this fact we can state the following:

PROPOSITION 1. If the purchaser can forge any signature during the payment protocol process, then he or she has to recover the purchaser's observer private ElGamal key x_O in a reasonable amount of time.

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

Formerly the security of the purchaser's identity relied on the uniqueness of his or her signature:

$$\mathrm{S}_i^{(4)} = Sig_{ElG}^{x_O}({A_P}^{Id_P})$$

as stated in Pointcheval & Stern (2000). To be able to prove this, allow us to consider the data that is being signed, i.e.:

$$A_P{}^{Id_P} = (G^{x_P})^{Id_P} = (G^{Id_P})^{x_P}$$

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 also unique and, hence, if $A_P{}^{Id_P} = A_P{}^{Id_{P\prime}}$ where $Id_{P\prime}$ is a forged identity, then $Id_P = Id_{P\prime}$. Furthermore, if $A_P{}^{Id_P} = A_{P\prime}{}^{Id_{P\prime}}$ then $x_P \cdot Id_P = x_{P\prime} \cdot Id_{P\prime}$ where data with index P' is fake. However, for the randomly chosen values $x_P, Id_P, x_{P'}, Id_{P\prime}$, the probability is as follows:

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

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

$$A_P{}^{Id_P} = (G^{Id_{P\prime}})^{x_{P\prime}} \quad (2)$$

for an unknown value of $x_{P\prime}$ which is a private key from the fake purchaser's P' observer. Hence, we are able to obtain the DLP as stated above.

The accuracy of the time instance t_i, payment amount m_i and the vendor's identity Id_V follows from the structure of $N_i^{(1)}$ and the signature:

$$S_i^{(3)} = Sig_{ElG}^{x_O}(A_P{}^{N_i^{(1)}})$$

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

$$A_P{}^{N_i^{(1)}} = (G^{A_P{}^{N'_i{}^{(1)}}})^{x_{P\prime}} \quad (3)$$

for an unknown value of $x_{P\prime}$, where $N'_i{}^{(1)}$ is meaningless data.

Hence the vendor will discover any alteration of data on the purchaser's side of the transaction by verifying the signatures in step 3 of the payment protocol process.

Valid signatures $S_i^{(1)}$ and $S_i^{(2)}$ ensure correct values for $w_i^{(1)}$ and $w_i^{(2)}$, which are required for successful Schnorr 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:

$$w_i^{(1)} = P_i^{(1)} \cdot (A_P{}^{N_i^{(1)}})^{-1} \quad (4)$$

$$w_i^{(2)} = P_i^{(2)} \cdot (A_P{}^{N_i^{(2)}})^{-1} \quad (5)$$

Since these identities hold true, the vendor will discover any forgery of these values in step 8 of the payment protocol. We can now claim the following:

PROPOSITION 2. The purchaser cannot forge any data that is sent during the payment protocol process. Hence, we can claim that the following corollary is true:

COROLLARY 1. All actions that are carried out by an **MP** attacker will be discovered by the vendor.

4.1.2 Analysis of a 'Man in the Middle' attack

We can now consider scenarios involving **MitM** attacks. Let us assume that an inside **MP** attacker has intercepted the payment protocol and has acquired the data that has been sent by another legitimate purchaser. The **MP**'s objective is to obtain the goods in question by using the potential victim's e-wallet. In order to achieve this goal, the **MP** has to forge the personal data of the potential victim by replacing it with his or her own. However, in this case, the **MP** has to deal with the following discrete logarithm problem:

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

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

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

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

$$w_i^{(1)} = G^{\xi^{(1)}}; \quad (8)$$

$$\tilde{w}_i^{(2)} = G^{\tilde{\xi}^{(1)}}. \quad (9)$$

Based on these facts we can claim the following:

PROPOSITION 3. If the **MP** can purchase goods using a legitimate purchaser's e-wallet then that **MP** has to solve the discrete logarithm problems (6), (8) and (9) in a reasonable amount of time.

Let us now assume that an inside **MV** attacker has intercepted the payment protocol and has acquired the data that has been sent by an honest purchaser. The **MV**'s objective is to deposit funds that are meant for another legitimate vendor. In order to achieve this goal, the potential victim's identity has to be forged by switching it with the **MV**'s own. This is not possible since the data that is sent to the observer does not have this information. Furthermore, the **MV**'s observer can use only the identity Id_{MP} and the **MV** cannot affect this this in any way. Hence the observer discovers that the hijacked transaction is not meant for the **MV** in step 3 of the deposit protocol by verifying the signature $S_i^{(3)}$ and therefore the observer blocks the deposit.

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

PROPOSITION 4. The system is resistant against **MitM** attack scenarios which involve purchaser or vendor impersonation.

4.1.3 Analysis of an attack by a 'Malicious Vendor'

To complete our analysis, we consider a scenario which involves an attack by an **MV**, i.e., one which exhibits actions that could be executed by a dishonest vendor in order that he or she might benefit from a transaction with an honest purchaser.

A 'double deposit' is prevented by the fact that the vendor is not able to forge a time instance due to Proposition 2, which also remains valid for the vendor. For this reason alone, the **MV** will discover such an attempt in step 2 of the deposit protocol process.

Any denial of payment is prevented by carrying out a check during steps 8-11 of the payment protocols since, in steps 9 and 10, the observer verifies the signature $S_i^{(3)}$ and therefore confirms that the payment has taken place by generating a signature S_V. Due to Proposition 1, which is also valid for the vendor, any honest purchaser will discover a fake verification check $(Id_V^{m_i||t_i}, S_V)$ in step 12 of the payment protocol.

The main goal is to prove that there is no manipulation of data taking place where that data has been received by the vendor which will disproportionately increase the balance of an e-wallet by affecting the payment amount. Such manipulations may also involve forging other parameters, such as A_P. Note, however, that the vendor is incapable of affecting any of the signatures that are received due to Proposition 1, which is also valid for the vendor. Any attempts to forge the value A_P will result in the creation of a 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, any forgery of the payment amount would imply the following equation:

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

for an unknown x, where $\tilde{m}_i$ is the forged payment amount and $\tilde{t}_i$ is the forged time instance. Hence, we can state the following:

PROPOSITION 5. If the vendor can manipulate the payment amount, then he or she has to solve the discrete logarithm problem (10) in a reasonable amount of time.

REMARK 1. The latter proposition is also valid for the purchaser.

Due to the acknowledged validity of the signatures that have been received, the vendor's observer will discover any attempted forgery by the vendor in step 3 of the deposit protocol.

Based on the results that have been presented here we can state the following:

PROPOSITION 6. Any actions which can be seen to be unfair or illegal and which are taken by the **MV** will be discovered.

Hence the reliance on Propositions 1, 4, and 6 helps us to reach the following conclusion:

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

4.2 AN ANALYSIS OF THE E-CASH SYSTEM'S TRUSTWORTHINESS

The trustworthiness of the proposed e-cash system is analysed using Burrows-Abadi-Needham logic (or BAN logic). BAN logic was first presented in Burrows, Abadi, & Needham (1989), and over time in Boyd & Mao (2007), 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 information being exchanged between different parties is trustworthy and that it has not been adversely affected by potential malicious insiders such as a malicious bank, vendor, purchaser, or others. BAN logic

begins with a set of goals that are meant to be proven, and the logic relies upon those assumptions which should be made and can be used as a basis for proof.

BAN logic is often used in the following areas:

✓ To describe the knowledge and beliefs of any parties who are involved in an authentication process in a formal manner

✓ To formally analyse the changing knowledge and the beliefs of the parties

✓ The logic behind authentication allows final protocol states to be made available

✓ To provide trust amongst communicating parties

BAN logic is meant for reasoning to be carried out in regard to cryptographic protocols. Any proof which uses BAN logic is a good proof of accuracy, based on the given assumptions.

The main BAN logic notations are presented in Table 2:

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 this
$P \overset{k}{\leftrightarrow} V$	A shared key k between P and V
$\#X$	X is refreshed
$A \lhd X$	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)
$< X >_k$	X is or 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

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

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

The purchaser's observer's parameters: $PrK_O = x_O,\ PuK_O = \{G, A_O = G^{x_O}\}$

We are going to keep the original notation for the 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 accuracy and security of our payment protocol, we will set the following goals:

Goal 1: $V \equiv m_i$

Goal 2: $V \equiv P$

Goal 3: $P \equiv V$

The following assumptions will be used as a basis for supplying proof for those goals:

Assumption 1: the vendor trusts that the purchaser's public parameters $G, A_P, A_P^{Id_P}$ as well as the public key A_O for his or her observer O_P are not forged in any way:

$P, V \mid\equiv G, A_P, A_O, A_P^{Id_P}$

Assumption 2: the vendor trusts observers as they represent the banks:

$V \mid \equiv O_P, O_V$

Assumption 3: the vendor receives correct public information from the purchaser:

$$A_P, A_O, G, A_P^{Id_P} \rightarrow V$$

Assumption 4: the purchaser receives correct public information from his or her observer:

$$A_O, G, A_P^{Id_P} \rightarrow P$$

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

Message 1: data which is generated by the purchaser's observer is sent to the purchaser:

$$O_V \rightarrow P: \begin{array}{c} \xi_i^{(1)}, \xi_i^{(2)},\ g^{\xi_i^{(1)}},\ g^{\xi_i^{(2)}}, \\ (m_i||t_i||Id_V),\ (m_i||t_i||Id_V) \cdot g^{\xi_i^{(1)}}, \\ < A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} >_{x_O}, \\ < A_P^{Id_P} \cdot A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} >_{x_O}, \\ < a^{(m_i||t_i||Id_V)} >_{x_O}. \end{array}$$

Message 2:

$$P \rightarrow V: \begin{array}{c} (m_i||t_i||Id_V), < A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} >_{x_O}, \\ < A_P^{Id_P} \cdot A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} >_{x_O}, \\ < A_P{}^{(m_i||t_i||Id_V)} >_{x_O}. \end{array}$$

Message 3:

$$V \rightarrow P:\ h_i$$

Message 4:

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

Message 5: the vendor sends response parameters to the purchaser in order to authenticate himself or herself:

$$V \rightarrow P: \left[Id_V, \{Id_V{}^{m_i||t_i}\}_{x_O}\right]$$

It follows from Message 2 that:

$$P \to V: \quad (m_i||t_i||Id_V), < A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} >_{A_O},$$
$$< A_P^{Id_P} \cdot A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} >_{x_O}, < a^{(m_i||t_i||Id_V)} >_{A_O}.$$

The application of the 'message seeing rule' results in the fact that the vendor sees the data that has been received from the purchaser:

$$V \lhd (m_i||t_i), < A_P{}^{(m_i||t_i||Id_V)} >_{A_O}, < A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(1)}} >_{A_O}, < A_P^{Id_P} \cdot A_P{}^{(m_i||t_i||Id_V)} \cdot g^{\xi_i^{(2)}} >_{A_O}$$

The application of the message meaning and belief rules and the use of the purchaser's observer's public key results in:

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

It follows from the belief rule that:

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

Since a is a public key, we get:

$$V| \equiv O_P| \sim m_i||t_i||Id_V$$

Subsequently, the vendor believes in the validity of these parameters:

$$V| \equiv O_P| \equiv m_i||t_i||Id_V$$
$$V| \equiv O_P| \equiv m_i$$

The application of a non-verification rule, plus jurisdiction and control, and the assumption that the observer is trusted by all parties results in acceptable proof that the first goal has been achieved.

$$V| \equiv m_i$$

Now we consider the second goal. The vendor sees the following information that has been received from the purchaser:

$$V \lhd \left[< (h_i', (m_i||t_i||Id_V)') >_{A_P, \xi_i^{(1)}}, < (h_i', Id_p', (m_i||t_i||Id_V') >_{A_P, \xi_i^{(2)}}\right]$$

The application of the message meaning rule and Assumption 3 results in:

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

So, the vendor believes that it was the purchaser who sent them the specified data. Moreover, it follows from Assumption 3 and concatenation rules that, due to the values of the total price m_i remaining non-forged along with the time instance t_i, it is in fact the purchaser who is interested in acquiring the relevant goods:

It follows from Assumption 1, Assumption 3, Assumption 4, and concatenation rules that:

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

Any non-verification rule can be applied as follows:

$$V| \equiv P| \Rightarrow m_i||t_i||Id_V$$
$$V| \equiv O_P \Rightarrow Id_p$$

and hence the vendor trusts that the purchaser has 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 jurisdiction over the purchaser via his or her representative (the observer O_P). Hence the vendor trusts the identity of the purchaser:

$V \mid\equiv P \lhd Id_p$

Finally, using jurisdiction, control, and referencing the rules, we obtain the following result:

$V \mid\equiv O \mid \Rightarrow Id_p$ and $V \mid\equiv O \mid \equiv Id_p$ result in:

$V \mid\equiv Id_p$

The second goal $V \mid\equiv P$ follows from the proven results $V \mid\equiv Id_p$ and $V \mid\equiv m_i$, since the vendor trusts the purchaser's identity and honesty (the sum m_i is not forged).

Now we consider the **third goal**. Due to Message 5, the purchaser sees the following information that has been received from the vendor:

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

Note, that the vendor has received this data from his or her observer, implying that:

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

By applying the message meaning rule, the concatenation rule, and Assumption 4 we obtain:

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

Thus, the purchaser believes that it was the vendor's observer, who generated the signature. Moreover, it follows from Assumption 4 and concatenation rules that, due to correct values of the total price m_i and the time instance t_i, the purchaser is dealing with an honest vendor:

$P \mid\equiv O_V \mid\sim\ Id_V$

We now apply the nonce-verification rule:

$P \mid\equiv O_V \Rightarrow Id_V$

Hence the purchaser believes that vendor's observer has jurisdiction over the vendor. Furthermore, due to this fact, the purchaser trusts that the vendor knows his or her identity since his or her observer possesses this information:

$P \mid\equiv V \lhd Id_V,\ P \mid\equiv O_V \lhd Id_V$

Finally, using jurisdiction and checks, and referencing the rules above, the purchaser trusts the identity of the vendor:

$P \mid\equiv Id_V$

Hence, the validity of the third goal $P \mid\equiv V$ now follows from the proven results.

4.3 A DIGITAL SIMULATION OF THE PROPOSED E-CASH SYSTEM

Since a considerable volume of the work involved in any payment operation is carried out by the ***observer*** with its somewhat restricted computational resources, the effectiveness of the proposed e-wallet system depends upon an estimation of the operational time involved in the process.

Computational time is directly related to the processor's clock frequency. If the processor is running at a clock frequency of 1 GHz then its clock cycle takes $10^{-9}\ s = 1ns$ of time.

The list of operations that are required to be performed as part of any payment protocol involves multiplication, addition, shifting, and modulus p. We can categorise these operations as elementary operations.

We assume that a 32-bit microprocessor is used in the ***observer***. This is far less than the bit length of the variables that are used in the payment protocol as represented by 2048-bit integers (although in some cases another bit length can be produced). Without stepping into any specifics, we can assume that all elementary operations take one clock cycle.

For the assessment of computation time, firstly we must estimate the number of elementary operations that are required for the calculation of the modular exponent function $r = g^k \bmod p$.

According to Hwang, Su, Yeh, & Chen (2005) and Knuth (1998) the modular exponent function is computed using the *addition chain method* (Knuth, 1998). The formulas to find the number of such operations are as follows:

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

where:

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

$A(w) = w/32$

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

- $\boldsymbol{MOD_E(y,z)}$ - denotes an operation of modular exponentiation $r = g^k \bmod p$;
- $\boldsymbol{M(w), A(w), Mod(w)}$ - denotes operations of multiplication, additions and modulus with a bit length of operand being w;
- $\boldsymbol{l(w)}$ - denotes the bit length of w;
- $\boldsymbol{S}$ - denotes the shift operator.

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

Table 3 Bit lengths of variables

Variable	**Bit length**
$p, q, x_P, x_O\ A_P, A_O, G, Id_P, Id_V, h_i, R$	$2048\ bits$
$\xi_i^{(1)}, \xi_i^{(2)}, w_i^{(1)}, w_i^{(2)}, N_i^{(1)}, N_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, r_i^{(1)}, r_i^{(2)}, S_V$	$2048\ bits$
m, t	$\sim 24\ bits$
$m_i \| \| t_i \| \| Id_V$	$\sim 2072\ bits$
$H(m)$	$\sim 256\ bits$

By default, we take 82 clock cycles for the SHA-2 algorithm computation (Guilford & Yap, 2012).

After the number N of clock cycles is found, the operational time can be estimated in the following way: $Time = N \cdot T$, where $T = 1/F$ and F is a clock

frequency so that we have a figure of 1.6 GHz in further sections for the demonstration of the calculations.

Appendix 1 presents the VBA code for the computation of operations $MOD_E(y,z)$ and also $M(w), A(w), Mod(w)$.

In terms of an example, we will show here how to calculate the duration of the process of the signature calculation.

$S_m = Sig^X_{ElG}(m) = \{R,s\} = \{G^k mod\ p, k^{-1}(h(m) - xR) mod\ q\}$,

Firstly, we calculate $G^k mod\ p$ and, thanks to that, we have $MOD_E(2048,2048) = 117903360$ in terms of clock cycles. After that we can calculate $h(m)$ where we get 82 clock cycles. Then xR is calculated only by multiplying $M(2048) = 8107$. Through the operation of $h(m) - xR$ we can carry out the process of addition so that we get $A(2048) = 64$. When calculating $k^{-1}(h(m) - xR)$ we gain only a multiplication of what is shown between the brackets, plus k^{-1}, so that we get $M(2048) = 8107$. Finally we get $k^{-1}(h(m) - xR) mod\ q$ where we have no exponentiation, and only the modulus operator $MOD_E(1,2048) = 57570$.

$$MOD_E(2048,2048) + 82 + M(2048) + A(2048) + M(2048) + MOD_E(1,2048)$$
$$= 117903360 + 82 + 8107 + 64 + 8107 + 57570$$
$$= 117977290$$

All together it takes a clock cycle of 117,977,290 to be able to generate a signature. This figure will be used for the further computation of all protocols.

For another example, we show how to calculate a thorough verification of a signature.

$$Ver^A_{ElG}(S_m, m) = \begin{cases} True, if\ R \in \boldsymbol{G_q}\ and\ A^R R^S = G^{h(m)} mod\ p \\ False, \quad \text{otherwise} \end{cases}$$

Firstly, as in the case of a signature calculation, we calculate A^R and R^S, and get $2 \cdot MOD_E(2048,2048) = 2 \cdot 117903360$ in terms of clock cycles. After that we calculate $h(m)$ where we get 82 clock cycles. For $G^{h(m)} mod\ p$ we get $MOD_E(256,2048) = 14737920$.

For all of the verification functions we gain the following:

$$MOD_E(k,p) = MOD_E(2048,2048) + MOD_E(2048,2048) + MOD_E(256,2048)$$
$$+ 82 = 117903360 + 117903360 + 14737920 + 82$$
$$= 250544722$$

All together a clock cycle delay of 250,544,722 clock is endured in order to be able to complete a verification operation. This figure will be used in the further computation of all protocols.

Further on in terms of calculating the withdrawal, payment, and deposit protocols it can be seen that some of the functions have approximately the same calculation times, such as the ElGamal signature function and the exponentiation of the modulus function. This happens because all calculations depend mostly upon the dimensions of the exponential and modulus divider number, and there is a negligible impact of operation such as sum, multiplication or only modulus functions.

As is shown by Eligijus Sakalauskas, Muleravicius, & Timofejeva (2017), we will present every step in the calculation of clock cycles and the elapsed time in all withdrawal protocols in Table 4:

Table 4 Calculation steps of Withdraw protocol

No	*Calculation step*	*Clock cycles*	*Operational time in ms*
1.	***P*** sends m_i, t_i, Id_V, A_P to O_p: $P \xrightarrow{m_i, t_i, Id_V, A_P} O_P$		
2.	O_p verifies the sum m_i and the time instance t_i:$Ver(t_i > t_{w0})$, $Ver(m_i < m_{max}^P)$		
3.	O_p generates random values: $Gen \rightarrow \xi_i^{(1)}, \xi_i^{(2)}$		
4.	O_p computes values: $w_i^{(1)} = G^{\xi_i^{(1)}}, w_i^{(2)} = G^{\xi_i^{(2)}}$	235806720	148 *ms*
5.	O_p calculates values:		
6.	$N_i^{(1)} = m_i \| \| t_i \| \| Id_V$	66	*0 ms*
7.	$N_i^{(2)} = Id_P \cdot N_i^{(1)}$	115140	*0.07 ms*
8.	$P_i^{(1)} = {A_P}^{N_i^{(1)}} \cdot w_i^{(1)}$	117960930	74 *ms*
9.	$P_i^{(2)} = {A_P}^{N_i^{(2)}} \cdot w_i^{(2)}$	117960930	74 *ms*
10	$S_i^{(1)} = Sig_{ElG}^{x_O}(P_i^{(1)})$	117977290	74 *ms*
11	$S_i^{(2)} = Sig_{ElG}^{x_O}(P_i^{(2)})$	117977290	74 *ms*
12	Calculate ${A_P}^{m_i \| \| t_i \| \| Id_V}$	117960930	74 *ms*
13	$S_i^{(3)} = Sig_{ElG}^{x_O}({A_P}^{m_i \| \| t_i \| \| Id_V})$	117977290	74 *ms*
14	Calculate ${A_P}^{Id_P}$	117960930	74 *ms*
15	$S_i^{(4)} = Sig_{ElG}^{x_O}({A_P}^{Id_P})$	117977290	74 *ms*
16	O_P updates a time instance: $t_{w0} \leftarrow t_i$		
17	O_P updates ***P***'s wallet balance: $m_{max}^P \leftarrow m_{max}^P - m_i$		
18	O_P sends computed values to ***P***: $O_P \xrightarrow{\xi_i^{(1)}, \xi_i^{(2)}, w_i^{(1)}, w_i^{(2)}, N_i^{(1)}, N_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} P$		

As has been previously described, generation of values where the exponential modulus is 2048 bits in length was most time consuming.

It can be seen from Table 4 that the number of clock cycles that are required for signature verification is 1,179,674,806, which corresponds to an operational time of 740 milliseconds.

Now we can calculate the clock cycles and elapsed time for the entire payment protocol process in Table 5:

Table 5 Calculation steps of Payment protocol

No	*Calculation step*	*Clock cycles*	*Operational time in ms*
1.	***P*** sends to ***V***: $P \xrightarrow{m_i \| t_i, A_P, {A_P}^{Id_P}, w_i^{(1)}, w_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)},} V$		
2.	***V*** verifies the sum m_i and the time instance t_i:$Ver(t_i > t_{p0})$, $Ver(m = \widetilde{m}_i)$		
3.	***V*** verifies signatures:		
4.	$Ver_{ElG}^{Ao}({A_P}^{Id_P}, S_i^{(4)})$	250544722	157 *ms*
5.	Calculate ${A_P}^{m_i \| t_i \| Id_V}$	117903360	74 *ms*
6.	$Ver_{ElG}^{Ao}({A_P}^{m_i \| t_i \| Id_V}, S_i^{(3)})$	250544722	157 *ms*
7.	Calculate ${A_P}^{m_i \| t_i \| Id_V} \cdot w_i^{(1)}$	57570	0.07 *ms*
8.	$Ver_{ElG}^{Ao}({A_P}^{m_i \| t_i \| Id_V} \cdot w_i^{(1)}, S_i^{(1)})$	250544722	157 *ms*
9.	Calculate $\left({A_P}^{Id_P}\right)^{(m_i \| t_i \| Id_V)}$	117969037	74 *ms*
10.	$Ver_{ElG}^{Ao}\left(\left({A_P}^{Id_P}\right)^{(m_i \| t_i \| Id_V)} \cdot w_i^{(2)}, S_i^{(2)}\right)$	250544722	157 *ms*
11.	***V*** generates a random challenge h_i :$Gen \rightarrow h_i$		
12.	***V*** sends a challenge h_i to ***P***: $V \xrightarrow{h_i} P$		
13.	***P*** calculates values $r_i^{(1)}$ and $r_i^{(2)}$:		
14.	$r_i^{(1)} = h_i \cdot x_P \cdot N_i^{(1)} + \xi_i^{(1)}$	172710	0.14 *ms*
15.	$r_i^{(2)} = h_i \cdot x_P \cdot N_i^{(2)} + \xi_i^{(2)}$	172710	0.14 *ms*
16.	***P*** forwards values $r_i^{(1)}$ and $r_i^{(2)}$ to ***V***:$P \xrightarrow{r_i^{(1)}, r_i^{(2)}} V$		
17.	***V*** verifies the Schnorr identification values:		
18.	$Ver\left(G^{r_i^{(1)}} \cdot \left({A_P}^{m_i \| t_i \| Id_V}\right)^{-h_i} = w_i^{(1)}\right)$	132641280	83 *ms*
19.	$Ver\left(G^{r_i^{(2)}} \cdot \left(\left({A_P}^{Id_P}\right)^{(m_i \| t_i \| Id_V)}\right)^{-h_i} = w_i^{(2)}\right)$	132641280	83 *ms*

No	*Calculation step*	*Clock cycles*	*Operational time in ms*
20.	$\boldsymbol{V}$ sends $m_i \| \| t_i, S_i^{(3)}$ to $\boldsymbol{O_V}$: $V \xrightarrow{m_i \| \| t_i, S_i^{(3)}} O_V$		
21.	$\boldsymbol{O_V}$ checks $Ver_{ElG}^{Ao}(A_P{}^{m_i \| \| t_i \| \| Id_V}, S_i^{(3)})$	250544722	157 *ms*
22.	$\boldsymbol{O_V}$ calculates $Id_V^{m_i \| \| t_i}$	1842240	1.15 *ms*
23.	$\boldsymbol{O_V}$ calculates $S_V = Sig_{ElG}^{xo}(Id_V^{m_i \| \| t_i})$	117977290	74 *ms*
24.	$\boldsymbol{O_V}$ sends to $\boldsymbol{V}$ the values $Id_V^{m_i \| \| t_i}, S_V$ $O_V \xrightarrow{Id_V^{m_i \| \| t_i}, S_V} V$		
25.	$\boldsymbol{V}$ sends to $\boldsymbol{P}$ the values $Id_V^{m_i \| \| t_i}, S_V$. V $\xrightarrow{Id_V^{m_i \| \| t_i}, S_V} P$		
26.	$\boldsymbol{P}$ verifies $Ver\left(\left(Id_V^{m_i \| \| t_i}\right)^{(m_i \| \| t_i)^{-1}}, Id_V\right)$	3684480	2.3 *ms*
27.	$\boldsymbol{P}$ verifies $Ver_{ElG}^{xo}(Id_V^{m_i \| \| t_i}, S_V)$	250544722	157 *ms*
28.	$\boldsymbol{V}$ renews a time instance: $t_{p0} \leftarrow t_i$		

The overall number of clock cycles that are required for the payment protocol is 2,128,330,289 and the time needed to carry out the protocol is 1,333 milliseconds.

Now we can calculate clock cycles and the elapsed time for all steps of deposit protocols in Table 6:

Table 6 Calculation steps of Deposit protocol

No	***Calculation step***	***Clock cycles***	***Operational time in ms***
1.	**V** sends to $\mathbf{O_V}$: $V \xrightarrow{m_i\|\|t_i,\, A_P,\, {A_P}^{Id_P}, w_i^{(1)}, w_i^{(2)}, S_i^{(1)}, S_i^{(2)}, S_i^{(3)}, S_i^{(4)}} O_V$		
2.	$\mathbf{O_V}$ verifies the time instance t_i:$Ver(t_i > t_{d0})$		
3.	$\boldsymbol{O_V}$ verifies the signatures:		
4.	$Ver_{ElG}^{A_O}({A_P}^{Id_P}, S_i^{(4)})$	250544722	157 *ms*
5.	Calculate ${A_P}^{m_i\|\|t_i\|\|Id_V}$	117903360	74 *ms*
6.	$Ver_{ElG}^{A_O}({A_P}^{m_i\|\|t_i\|\|Id_V}, S_i^{(3)})$	250544722	157 *ms*
7.	Calculate ${A_P}^{m_i\|\|t_i\|\|Id_V} \cdot w_i^{(1)}$	57570	0.07 *ms*
8.	$Ver_{ElG}^{A_O}({A_P}^{m_i\|\|t_i\|\|Id_V} \cdot w_i^{(1)}, S_i^{(1)})$	250544722	157 *ms*
9.	Calculate $\left({A_P}^{Id_P}\right)^{(m_i\|\|t_i\|\|Id_V)}$	117903360	74 *ms*
10.	$Ver_{ElG}^{A_O}\left(\left({A_P}^{Id_P}\right)^{(m_i\|\|t_i\|\|Id_V)} \cdot w_i^{(2)}, S_i^{(2)}\right)$	250544722	157 *ms*
11.	$\boldsymbol{O_V}$ renews a time instance: $t_{d0} \leftarrow t_i$		
12.	$\mathbf{O_V}$ updates **V**'s wallet balance: $m_{max}^V \leftarrow m_{max}^V + m_i$		

The overall number of clock cycles that are required for the deposit protocol is 1,238,043,178 and the time needed to carry out the protocol is 776 milliseconds.

The comparison of our system with those of Brands and CHL is presented in Table 7.

Thanks to Hinterwälder, Riek, & Paar (2015), all protocols in the Brands e-cash require about 2966 ms in all protocols that are generated in cards. As to Au, Susilo, & Mu (2007), the computational time required for the CHL e-cash protocol in a single payment is thirty modular exponentiations which takes about 2111 ms according to approximations that were published by Juang (2010), but the cost of each operation is somehow hard to compute because this depends upon how many transactions have previously been made and how many e-coins were used in those previous transactions. A comparison of our system with those of Brands and CHL is presented in Table 7.

Table 7 Computational time comparisons in ms.

Protocol	**Our system**	**Brands**	**CHL**
Withdrawal	740	-	-
Payment	1,333	-	-
Deposit	776	-	-
Total	2,849	2,996	2,111

Computational time analysis is carried out for the purchaser observer's e-wallet and the vendor's computational device. Analysis has shown that the realisation of

cryptographic functions provides an acceptable operational time when microprocessors are running at a clock frequency of 1.6 GHz.

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

4.4 ANALYSIS AND COMPARISON OF E-CASH SYSTEMS

The e-payment process is widely used for e-commerce and has attracted many mobile or computer users in particular. The currently-available solutions have usually been designed to protect the customer's funds in terms of security without necessarily protecting their privacy.

Modern e-commerce still lacks a widely available and acceptable e-money system. This is a problem which in general results from the actions (or inaction) of the banks, governments, sellers, community conflicts, and the attractiveness of e-cash.

The suitability of electronic retail payments for surveillance purposes is high, and a fully traceable e-cash system would be a boon for intelligence agencies, allowing many more profiles than ever before to be collected in regard to people's behaviour.

Defined as an electronic payment transfer from payer to payee, electronic payments have already been realised in a variety of ways, and well-known companies such as Revolut, MoQ, PayPal, Google Wallet, Apple Pay, Alipay, and others use them in mobile devices.

With the widespread use of mobile e-payment, buyers face a risk of their privacy being exposed.

For our system, we have employed the ElGamal signature scheme (ElGamal, 1985), in order to provide authentication for the e-cash system by using the bank's implemented chip as a trustee. In addition, the scheme carries out a confidence check between the purchaser and the vendor.

Data integrity and authentication between the purchaser and the vendor is guaranteed by Schnorr's identification scheme (Schnorr, 1990) and also by MAC. The process of ensuring non-repudiation is achieved by combining these two systems.

This new scheme, which is combined with two cryptographic algorithms, give us an e-cash system that is secure, there is also no data expansion when transfers are made. In addition, it is divisible, anonymity between buyer and receiver of the e-money is ensured, and there is an opportunity to conclude off-line payments. These crypto systems, which have been chosen to integrate them into any mobile device, can work better than any others that are currently available. When combined with the physical unclonable function that was described in the first section very good results can be obtained.

In this thesis, we have shown that this new system is simpler than the presently-available ones and that it has more potential to be more widely used.

It is important to mention that a better system can be put in place if we eliminate the trustee from the e-payment system. This is something that has not yet been created or implemented by anyone.

In Table 8 we present a comparison of some of the available e-cash systems, all of which satisfy such properties as off-line payment, transferability, anonymity from

the vendor, and etc. However, each of these systems possesses the flaw which sees the data related to funds expand in size with every transfer. Furthermore, any previously presented e-cash system which eliminates this flaw also loses the function of ensuring anonymity from the vendor or the important off-line payment properties. Another possible alternative is crypto currencies such as Bitcoin.

Table 8 Comparisons of functional characteristics with other systems

Author	Year	E-money systems	Off-line	Transferable	Untraceable	Data expansion	Prevents 'double spending'	Over-spending prevention	Anonymity from vendor	Anonymity from bank
Chaum et al	1988	CFN	Online	Non-transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
O'Mahony et al	1990	Mondex	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Revealed
Brands	1994	Brands	Off-line	Transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
D'Amiano and Crescenzo	1994	S D'Amiano	Off-line	Transferable	Untraceable	No	Prevented	Prevented	Revealed	Anonymous
Boly et al	1994	CAFE	Off-line	Non-transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Okamoto	1995	Okamoto	Off-line	Transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Peterson and Poupard	1996	FOLC	Off-line	Transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Revealed
Tsiounis	1997	FOLC	Off-line	Transferable	Untraceable	Yes	No	Prevented	Anonymous	Anonymous
Camenisch et al	2005	CHL	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Kreft and Adi	2006	fairCASH	Off-line	Transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Camenisch et al	2007	Endorsed	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Canard et al	2008	Canard e-cash	Online	Transferable	Traceable	Constant size	Prevented	Prevented	Anonymous	Revealed
Canard and Gouget	2008	Canard e-cash	Online	Transferable	Traceable	No	Prevented	Prevented	Anonymous	Anonymous
Nakamoto	2009	Bitcoin	Off-line	Transferable	Untraceable	Yes	No	No	Anonymous	Anonymous
Tiwari	2009	Secret splitting	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Blazy et al	2011	GS proof e-cash	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Baldimtsi et al	2015	Baldimtsi	Off-line	Transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Canard	2015	Canard e-cash	Off-line	Non-transferable	Untraceable	Yes	Prevented	Prevented	Revealed	Revealed
Märtens	2015	Märtens	Off-line	Non-transferable	Traceable	Yes	Prevented	Prevented	Anonymous	Anonymous
Canard	2015	Scalable e-cash	Off-line	Transferable	Untraceable	Yes	Prevented	Prevented	Anonymous	Anonymous
E-wallet proposed in this dissertation	2018	Dissertation	Off-line	Transferable	Untraceable	No	Prevented	Prevented	Anonymous	Revealed

This system stands out from other comparable systems in terms of parameters that are either roughly the same or which are demonstrably better (such as being divisible when compared to Brands' system, and not expanding the data size when compared to the CHL system), and our system has protocol computational times that are nearly the same as those of the competition or which are demonstrably better than them.

As can be seen, our e-cash system has the following functional advantages: anonymity from the vendor, off-line payments, divisibility, transferability, and the 'double spending' prevention requirement and most important of all, the fact that data does not expand in size when transferred. So, we can have a very similar payment system to that of physical cash.

4.5 CONCLUDING REMARKS AND RESULTS

1. Our new e-cash system is resistant against an attack by a 'Malicious Purchaser'. The purchaser cannot forge any data in order to carry out 'double spending' and remain undetected. The purchaser is also unable to forge transaction data, in addition, he or she has to resolve a discrete logarithm problem.

2. Our new e-cash system is resistant against an attack by 'Malicious Vendor'. The vendor is unable to forge transaction data without solving the discrete logarithm problem. In addition, the vendor cannot make a 'double deposit' and remain undetected. Furthermore, if the vendor attempts a fraudulent action by receiving funds in his/her e-wallet and then refusing to hand over the goods to the purchaser by claiming that the payment has not been correctly executed, such a fraud would be detected.

3. Our new e-cash system is resistant against a 'Man in the Middle' attack. Impersonation is not possible, i.e., if an insider tries to do that, he or she has to solve a discrete logarithm problem.

4. The use of the e-cash can be trusted under this new e-cash scheme. The vendor will have no reason to mistrust the purchaser or the amount of money that is being transferred between them. In addition, the purchaser will have no reason to mistrust the vendor.

5. The theoretical calculation time in the e-cash system is 2849 ms, which consists of withdrawal, payment, and deposit protocols. Realisation provides an acceptable operational time when microprocessors are running at a clock speed of 1.6 GHz.

6. Based on the presented comparison of other systems in literature (Baldimtsi, Chase, Fuchsbauer, & Kohlweiss, 2015; Blazy et al., 2011; Bosselaers et al., 2012; S. Brands, 1994; Camenisch et al., 2005, 2007; Canard & Gouget, 2008; Canard, Pointcheval, Sanders, & Traoré, 2015; Chaum et al., 1988; D. O'Mahony M. Peirce, 1997; D'Amiano & Di Crescenzo, 2006; Fuchsbauer et al., 2009; Kreft & Adi, 2006; Märtens, 2015; Nakamoto & others, 2008; Okamoto, 1995; Petersen & Poupard, 2005; E. Sakalauskas et al., 2018; Tiwari, 2009; Tsiounis, 1997), we can conclude that our system stands out from other systems since it possesses similar characteristics to those schemes whilst also avoiding the drawback of data expansion. Furthermore, it has a computational time that is either approximately the same as other systems have or is, in fact, better if compared to the other schemes that are presented in Table 8.

CONCLUSIONS

The basic tasks that were the goal of this dissertation were accomplished:

1. The analysis of existing the e-cash systems and their main properties is presented in Section 1 and their comparison is given in subsection 4.5.

The system for mobile devices was constructed, it is advanced and attractive in comparison to the other e-cash systems (subsection 4.4); secure (subsection 4.1), and suitable in terms of the time required for payment (subsection 4.3). The use of new e-cash system required a much-reduced total of calculation resources.

2. The trustworthiness of BAN logic was checked under the new e-cash scheme in subsection 4.2 to determine its functionality in terms of three main goals: the vendor being able to trust the received payment amount, the vendor being able to trust the purchaser, and the purchaser being able to trust the vendor.

3. Digital simulation was performed which estimated the theoretical processing time of the new e-cash scheme for withdrawal, payment, and deposit protocols, with that processing time being 740, 1,333, and 776 milliseconds, respectively (subsection 4.3). The total processing time equals 2,849 milliseconds.

4. The security requirements of the proposed e-cash system have been proved in certain ways: during the attacks by a 'Malicious Purchaser', a 'Man in the Middle', and a 'Malicious Vendor'.

An overview of novel features:

The proposed e-cash system does not produce data expansion during transfers because it revokes anonymity between the bank and the purchaser. This scheme still has the following main properties: it is providing off-line and divisible payments, it ensures anonymity between the purchaser and the vendor, it is also preventing 'double spending', ensuring untraceability, while at the same time remaining secure.

Conclusions:

1. The new e-cash system ***does not produce data expansion*** when transfers are made between users, at the same time having all the other main properties: off-line payments; divisible e-cash; legal payment's untraceability; transferability between users; and the purchaser retaining anonymity from the vendor.

2. The new e-cash system is secure from an attacks by a 'Malicious Purchaser', a 'Man in the Middle', and by a 'Malicious Vendor'. In addition, it continues to prevent 'double spending'.

3. The new e-cash system can be fully trusted when using BAN logic. There is no reason for the vendor not to trust the purchaser and, vice versa, there is no reason for the purchaser not to trust the vendor. In addition, the vendor can also trust a payment amount which is received from the purchaser.

4. The new e-cash system can effectively be used on mobile devices. All of the processing time that is involved in withdrawal, payment, and deposit procedures amounts to 2,849 milliseconds. In comparison to other e-cash systems, this one is approximately equal to them or better than them, while also maintaining important properties.

5. The new e-cash system we can call more attractive and advance, to the fact that it has better features.

6. We can say that the new e-cash system is more attractive and advanced, because it has improved features.

ACKNOWLEDGMENTS

This thesis would have been impossible without the support and contributions from my advisor, Professor Eligijus Sakalauskas. I would like to extend my thanks to Inga Timofejeva and Aleksejus Michalkovic for helping with security analysis and other techniques that are involved in the cryptographic primitives.

I would like to thank my friends for accepting nothing less than excellence from me. Lastly, I would like to thank my family: my parents, my brother, and my fiancée for supporting me spiritually throughout the process of writing this thesis and in my life in general.

REFERENCES

Abe, M., Haralambiev, K., & Ohkubo, M. (2010). Signing on Elements in Bilinear Groups for Modular Protocol Design. *IACR Cryptology EPrint Archive*, *2010*, 133.

Au, M. H., Susilo, W., & Mu, Y. (2007). Practical Compact E-Cash. In *Information Security and Privacy*. https://doi.org/10.1007/978-3-540-73458-1_31

Au, M. H., Susilo, W., & Mu, Y. (2011). Electronic cash with anonymous user suspension. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-22497-3_12

Baldimtsi, F., Chase, M., Fuchsbauer, G., & Kohlweiss, M. (2015). Anonymous transferable e-cash. In *Public-Key Cryptography -- PKC 2015*. https://doi.org/10.1007/978-3-662-46447-2_5

Baseri, Y., Takhtaei, B., & Mohajeri, J. (2013). Secure untraceable off-line electronic cash system. *Scientia Iranica*. https://doi.org/10.1016/j.scient.2013.05.002

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *Proceedings - IEEE Symposium on Security and Privacy*. https://doi.org/10.1109/SP.2014.36

Blazy, O., Canard, S., Fuchsbauer, G., Gouget, A., Sibert, H., & Traoré, J. (2011). Achieving optimal anonymity in transferable e-cash with a judge. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-21969-6_13

Boneh, D., & Shoup, V. (2017). A Graduate Course in Applied Cryptography. *Crypto.Stanford.Edu*.

Bosselaers, A., Schunter, M., Michelsen, R., Cramer, R., Vallée, L., Waidner, M., … Mjølsnes, S. (2012). The ESPRIT project CAFE —High security digital payment systems. https://doi.org/10.1007/3-540-58618-0_66

Boyd, C., & Mao, W. (2007). On a Limitation of BAN Logic. In *Advances in Cryptology — EUROCRYPT '93*. https://doi.org/10.1007/3-540-48285-7_20

Brands, S. (1994). Untraceable Off-line Cash in Wallets with Observers (Extended abstract). *Advances in Cryptology—CRYPTO'93*. https://doi.org/10.1007/3-540-48329-2_26

Brands, S. (2012). Off-line electronic cash based on secret-key certificates. https://doi.org/10.1007/3-540-59175-3_86

Brands, S. ~A. (1993). An efficient off-line electronic cash system based on the representation problem. *NASA STI/Recon Technical Report N*, *94*.

Burrows, M., Abadi, M., & Needham, R. (1989). A logic of authentication. *ACM Transactions on Computer Systems*. https://doi.org/10.1145/77648.77649

Camenisch, J., Hohenberger, S., & Lysyanskaya, A. (2005). Compact E-Cash. https://doi.org/10.1007/11426639_18

Camenisch, J., Lysyanskaya, A., & Meyerovich, M. (2007). Endorsed e-cash. In *Proceedings - IEEE Symposium on Security and Privacy*. https://doi.org/10.1109/SP.2007.15

Canard, S., & Gouget, A. (2008). Anonymity in transferable e-cash. In *ACNS'08 Proceedings of the 6th international conference on Applied cryptography and network security*. https://doi.org/10.1007/978-3-540-68914-0_13

Canard, S., Pointcheval, D., Sanders, O., & Traoré, J. (2015). Scalable divisible E-cash. In *Applied Cryptography and Network Security*. https://doi.org/10.1007/978-3-319-28166-7_14

Chan, A., Frankel, Y., & Tsiounis, Y. (1998). Easy come — easy go divisible cash. In *Advances in Cryptology — EUROCRYPT'98*. https://doi.org/10.1007/BFb0054154

Chaum, D., Fiat, A., & Naor, M. (1988). Untraceable Electronic Cash. In *Advances in*

Cryptology — CRYPTO' 88. https://doi.org/10.1007/0-387-34799-2_25
Chaum, D., & Pedersen, T. P. (1993). Transferred cash grows in size. In *Advances in Cryptology — EUROCRYPT' 92*. https://doi.org/10.1007/3-540-47555-9_32
Clarke, R. (1996). The Mondex Value-Card Scheme A Mid-Term Report.
Common Criteria. (2017). Common Criteria for Information Technology Security Evaluation - Part 3 : Security assurance components. *Common Criteria*.
Cortez, M., Dargar, A., Hamdioui, S., & Schrijen, G. J. (2012). Modeling SRAM start-up behavior for physical unclonable functions. In *Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. https://doi.org/10.1109/DFT.2012.6378190
Cramer, R., & Shoup, V. (2004). Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing*. https://doi.org/10.1137/s0097539702403773
D. O'Mahony M. Peirce, H. T. (1997). Electronic Payment Systems.
D.Everett. (2016). *Digital Cash - A Short Briefing Note*.
D'Amiano, S., & Di Crescenzo, G. (2006). Methodology for digital money based on general cryptographic tools. https://doi.org/10.1007/bfb0053432
de Solages, A., & Traorè, J. (1998). An efficient fair off-line electronic cash system with extensions to checks and wallets with observers. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/BFb0055489
Diffie, W., Diffie, W., & Hellman, M. E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.1976.1055638
ElGamal, T. (1985). A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, *196 LNCS*, 10–18. https://doi.org/10.1007/3-540-39568-7_2
Eng, T., & Okamoto, T. (2006). Single-term divisible electronic coins. https://doi.org/10.1007/bfb0053446
Eslami, Z., & Talebi, M. (2011). A new untraceable off-line electronic cash system. In *Electronic Commerce Research and Applications*. https://doi.org/10.1016/j.elerap.2010.08.002
Fan, C. I., Huang, V. S. M., & Yu, Y. C. (2013). User efficient recoverable off-line e-cash scheme with fast anonymity revoking. *Mathematical and Computer Modelling*. https://doi.org/10.1016/j.mcm.2012.07.012
Franklin, M., Yung, M., & Center, T. J. W. (1992). *Towards Provably Secure E cient Electronic Cash*.
Fuchsbauer, G. (2009). Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. *IACR Cryptology EPrint Archive*, *2009*, 320.
Fuchsbauer, G., Pointcheval, D., & Vergnaud, D. (2009). Transferable constant-size fair E-cash. In *Cryptology and Network Security*. https://doi.org/10.1007/978-3-642-10433-6_15
Herder, C., Yu, M. D., Koushanfar, F., & Devadas, S. (2014). Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2014.2320516
Hinterwälder, G., Riek, F., & Paar, C. (2015). Efficient e-cash with attributes on MULTOS smartcards. In *Radio Frequency Identification*. https://doi.org/10.1007/978-3-319-24837-0_9
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 *Lecture Notes in Computer Science (including subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-39077-7_3

Hwang, R. J., Su, F. F., Yeh, Y. S., & Chen, C. Y. (2005). An efficient decryption method for RSA cryptosystem. In *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*. https://doi.org/10.1109/AINA.2005.97

J. Guilford K. Yap, V. G. (2012). Fast SHA-256 Implementations on Intel. *Architecture Processors, White Paper*.

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*. https://doi.org/10.1016/j.jss.2009.11.006

Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching*. *Computer*. https://doi.org/10.2307/2283757

Kreft, H., & Adi, W. (2006). fairCASH - A digital cash candidate for the proposed GCC gulf dinar. In *2006 Innovations in Information Technology, IIT*. https://doi.org/10.1109/INNOVATIONS.2006.301916

Märtens, P. (2015). Practical Divisible E-Cash. *IACR Cryptology EPrint Archive*, *2015*, 318.

Messmer, E. (2010). Black Hat: Researcher claims hack of chip used to secure computers, smartcards. *DG Communications, Inc.*

Muleravičius, J., Sakalauskas, E., & Timofejeva, I. (2016). *On methodology of e-wallet construction for partially off-line payment system*. *Communications in Computer and Information Science* (Vol. 639). https://doi.org/10.1007/978-3-319-46254-7_61

Muleravicius, J., Timofejeva, I., Mihalkovich, A., & Sakalauskas, E. (2019). Security, Trustworthiness and Effectivity Analysis of an Offline E-Cash System with Observers. *Informatica*, *30*(2), 327–348.

Nakamoto, S., & others. (2008). Bitcoin: A peer-to-peer electronic cash system.

Okamoto, T. (1995). An efficient divisible electronic cash scheme. In *CRYPTO '95 Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*. https://doi.org/10.1007/3-540-44750-4_35

Okamoto, T., & Ohta, K. (2007). Universal Electronic Cash. In *Advances in Cryptology — CRYPTO '91*. https://doi.org/10.1007/3-540-46766-1_27

Pailles, J. C. (2012). New protocols for electronic money. In *Advances in Cryptology — AUSCRYPT '92*. https://doi.org/10.1007/3-540-57220-1_68

Petersen, H., & Poupard, G. (2005). Efficient scalable fair cash with off-line extortion prevention. https://doi.org/10.1007/bfb0028503

Pfitzmann, A., & Köhntopp, M. (2007). Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology. https://doi.org/10.1007/3-540-44702-4_1

Pointcheval, D., & Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology*. https://doi.org/10.1007/S001450010003

Rabin, M. (1978). Digitalized Signatures and Public-Key Functions as Intractable as Factorization. *Foundations of Secure Computations*. https://doi.org/10.1080/09720529.2013.858478

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. https://doi.org/10.1145/359340.359342

Rosenberg, B. (2010). *Handbook of financial cryptography and security*. *Handbook of Financial Cryptography and Security*. https://doi.org/10.1201/9781420059823

Sakalauskas, E., Timofejeva, I., Michalkovič, A., & Muleravičius, J. (2018). A simple off-line e-cash system with observers. *Information Technology and Control*, *47*(1). https://doi.org/10.5755/j01.itc.47.1.18062

Sakalauskas, Eligijus, Muleravicius, J., & Timofejeva, I. (2017). Computational resources for

mobile E-wallet system with observers. In *Proceedings of the 21st International Conference on Electronics*. https://doi.org/10.1109/ELECTRONICS.2017.7995226

Schellhorn, G., Grandy, H., Haneberg, D., & Reif, W. (2006). The Mondex Challenge: Machine Checked Proofs for an Electronic Purse. https://doi.org/10.1007/11813040_2

Schmitt, P. H., & Tonin, I. (2007). Verifying the mondex case study. In *Proceedings - 5th IEEE International Conference on Software Engineering and Formal Methods, SEFM 2007*. https://doi.org/10.1109/SEFM.2007.47

Schnorr, C. P. (1990). Efficient identification and signatures for smart cards. In *Advances in Cryptology — CRYPTO' 89 Proceedings*. https://doi.org/10.1007/3-540-46885-4_68

Srivastava, L., & Mansell, R. (1998). *Electronic Cash and the Innovation Process: A Use Paradigm*. University of Sussex, SPRU.

Stadlerl, M., Piveteau, J. M., & Camenisch, J. (1995). Fair blind signatures. In *Advances in Cryptology — EUROCRYPT '95*. https://doi.org/10.1007/3-540-49264-X_17

Stalder, F. (2002). Failures and successes: Notes on the development of electronic cash. *Information Society*. https://doi.org/10.1080/01972240290074968

Stepney, S., Cooper, D., & Woodcock, J. (2000). An electronic purse: Specification, refinement and proof.

Suh, G. E., & Devadas, S. (2007). Physical unclonable functions for device authentication and secret key generation. In *Proceedings - Design Automation Conference*. https://doi.org/10.1109/DAC.2007.375043

Tam, K. Y., & Ho, S. Y. (2011). A Smart Card Based Internet Micropayment Infrastructure: Technical Development and User Adoption. *Journal of Organizational Computing and Electronic Commerce*. https://doi.org/10.1080/10919390701294095

Tiwari, K. (2009). Transferable E-Cash Without Observer. *A Thesis for Master of Technology, Department of Computer Science and Engineering, Indian Institute of Technology Kanpur*.

Tsiounis, Y. S. (1997). *Efficient Electronic Cash: New Notions and Techniques*.

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

Wilson, T. (2010). Researcher Cracks Security Of Widely Used Computer Chip.

Yan Liang, Z. X., & Zhi-ming, Z. (2016). An Electronic Cash System Based on Certificateless Group Signature. *International Journal of Security and Its Applications*, 237–300.

LIST OF PUBLICATIONS

Papers in the Master list of journals of the Institute for Scientific Information (ISI)

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.

Muleravičius, J., Timofejeva, I, Sakalauskas, E., &. Mihalkovich, A. (2019). Security, Trustworthiness and Effectivity Analysis of an Off-line E-Cash System with Observers. Informatica, Vol. 30, No. 2, 327–348.

Papers in other reviewed scientific editions

Muleravičius, J., Sakalauskas, E., & Timofejeva, I. (2016, October). On Methodology of E-wallet Construction for Partially Off-line Payment System. In International Conference on Information and Software Technologies (pp. 753-765). Springer, Cham.

Papers in the proceedings list

Sakalauskas, E., Muleravicius, J., & Timofejeva, I. (2017, June). Computational resources for mobile E-wallet system with observers. In Electronics, 2017 (pp. 1-5). IEEE.

APPENDIX 1

```
 VBA kodas
Function A(w As Long)
Dim temp As Long
 temp = w / 32
A = temp
End Function
Function M(w As Long)
Dim temp, S As Long
Dim w0 As Long
S = 1
w0 = w
k = Log(w) / Log(2) - Log(32) / Log(2) + 1
k = Round(k, 0)
'MsgBox (k)
  For i = 1 To k
    If i <> k Then
                  temp = temp + (5 * A(w0) + 2 * S) *
(3 ^ (i - 1))
           w0 = w0 / 2
          Else:
            temp = temp + (3 ^ (i - 1))
            w0 = w0 / 2
    End If
  Next i
M = temp

End Function
Function Modw(w As Long)
Dim temp, S As Long
Dim w0 As Long
S = 1
w0 = w
k = Log(w) / Log(2) - Log(32) / Log(2) + 1
k = Round(k, 0)
'MsgBox (k)
  For i = 1 To k
    If i <> k Then
           temp = temp + (4 * M(w0 / 2) + 1.5 * A(w0) +
3 * S)
           w0 = w0 / 2
    End If
  Next i
Modw = temp + 1
End Function
Function MODyz(y As Long, z As Long)
```

```
'y - laipsnis, z mod argumentas
Dim temp
  temp = 1.5 * y * (M(z) + 2 * Modw(z) + 1)
  MODyz = temp
End Function
Function op_time(f As Double)
'y - laipsnis, z mod argumentas
Dim temp
  temp = 1 / (f * (10 ^ 9))
  op_time = temp
End Function
```

SL344. 2019-10-04, 8,75 leidyb. apsk. l. Tiražas 14 egz. Užsakymas 219.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas