INFORMATICA, 2011, Vol. 22, No. 3, 395–409

A Secure E-Cash Transfer System Based on the Elliptic Curve Discrete Logarithm Problem

Constantin POPESCU
Department of Mathematics and Computer Science, University of Oradea
Universitatii str. 1, Oradea, Romania
e-mail: cpopescu@uoradea.ro

Received: September 2010; accepted: April 2011

Abstract. Electronic commerce (e-commerce) is a relatively new, emerging and constantly changing area of business management and information technology. One of the technological innovations in banking, finance and e-commerce is the electronic cash (e-cash) transfer system. E-cash transfer systems refers to the technological breakthrough that enables us to perform financial transactions electronically. In this paper we propose a secure e-cash transfer system based on the elliptic curve cryptography. In order to protect the honest participants of the e-cash system we use an elliptic curve blind signature scheme and also we need a trusted third party to trace the criminals.

Keywords: elliptic curve cryptography, electronic commerce, e-cash system, digital signatures, wireless communications.

1. Introduction

Electronic commerce is one of the most important applications on the Internet. Customers' privacy must be protected if they are embedded in legal commercial transactions or payments. In last years, cryptographic protocols and network technologies have undergone rapid development (Xiong *et al.*, 2010; Raulynaitis *et al.*, 2010; Sakalauskas *et al.*, 2007; Liu and Huang, 2010). The smart card based remote user authentication scheme (Tseng*et al.*, 2008; Li and Hwang, 2010) and the electronic cash (e-cash) transfer system are the simplest and most convenient authentication mechanisms for insecure networks. Blind signature schemes (Pointcheval and Stern, 2000) have been widely used to protect the right of an customer's privacy in the untraceable electronic cash systems (Chaum, 1983). However, it is easy to make multiple copies of the electronic coin, which is in the form of number strings. Therefore, blind signature schemes are used in order to eliminate the possible abuse of unlinkability. If a user makes an abuse in the e-cash system, such as double spending, blackmailing or money laundering, then a trusted third party runs a specific protocol (tracing protocol) in order to reveal his/her identity.

Chaum (1983) proposed the first untraceable electronic cash system based on blind signatures in 1982. Also, various extended systems have been proposed, which provide functionalities such as anonymity, double spending prevention, unforgeability, untraceability and efficiency (Trolin, 2005; Lee *et al.*, 2002; Okamoto, 1995). In the field of e-cash systems, the notions on-line and off-line refer to a specific property of the payment protocol. On-line e-cash systems require constant and real time involvement of the bank in every payment transaction, resulting in excessive communication and computation costs. In contrast, off-line e-cash systems usually operate in dual mode, verifying high cost transactions on-line, while the rest of payments are processed in batch mode by the bank.

The first off-line electronic cash system has introduced by Chaum *et al.* (1990) and then developed further Popescu (2006, 2009), De Santis *et al.* (2007), Chou *et al.* (2009), Au *et al.* (2008). In these cases the bank has not involved in the payment transaction between a customer and a merchant. Customers withdraw electronic coins from the bank and use them to pay a merchant (a shop). Then, the merchant deposits the coins back to the bank.

In order to prevent criminal activities or to trace the criminals we need some anonymity revocation mechanisms. The off-line e-cash systems use a trusted third party (TTP) to trace the criminals in order to protect the honest participants of the e-cash system.

In 1985, Miller (1986) and Koblitz (1987) introduced Elliptic Curve Cryptography (ECC) which has attracted increasing attention in recent years due to its shorter key length requirement in comparison with other public key cryptosystems such as DSA (NIST, 2009), ElGamal (Elgamal, 1985) and RSA (Rivest *et al.*, 1978). For example, 160-bit elliptic curve version of DSA signature algorithm (ECDSA) has a security level equivalent to 1024-bit DSA signature algorithm. Such advantages make elliptic curve cryptography a better choice for public key cryptography.

In this paper we propose a secure e-cash transfer system based on the elliptic curve cryptography. Our e-cash transfer system does not use pairings (bilinear maps), which not only results in greater efficiency and ease of implementation, but also means that our system does not rely on the relatively new and untested hardness assumptions related to pairing based cryptography. The proposed off-line electronic transfer system can be used in the wireless networks with the limited bandwidth.

The rest of this paper is organized as follows. In the next section we review the model of an e-cash transfer system and the basic knowledge of the elliptic curves cryptography. Then we present our e-cash transfer system in the Section 3. Furthermore, we discuss some aspects of security in the Section 4. The Section 5 concludes the work of our paper.

2. Background

In this section, we briefly review the roles and functions in a secure electronic cash system and the basic knowledge of the elliptic curve cryptography.

2.1. *The Model of an E-Cash Transfer System*

Electronic cash (e-cash) is a new concept in online payment system because it combines computerized convenience with security and privacy that improve on paper cash. A typical electronic cash system has three participants: a bank, customers (users) and merchants (shops) and three main phases (procedures): withdrawal, payment and deposit. Customers and merchants open and maintain an account with the bank. The customer withdraws electronic coins from his account, by performing a withdrawal subprotocol with the bank over an authenticated channel. Then, the customer spends a coin in a payment subprotocol with a merchant over an anonymous channel and a merchant runs a deposit subprotocol with the bank in order to deposit the coin of the merchant into his account. E-cash systems are conventionally divided into those that are on-line and those that are off-line. In an online system, the bank must be involved for each transaction, although the anonymity of at least the customer remains protected. The e-cash system is off-line if during payment phase the merchant does not communicate with the bank. The bank can legally trace a dishonest customer with the help of the trusted third party (TTP). We summarize six important requirements (Lee *et al.*, 2002; Chou *et al.*, 2009) for a secure electronic cash transfer system. They are:

- Mutual authentication: Two parties can authenticate each other correctly.
- Correctness: One can ensure the correctness and integrity of messages transmitted by the other designated party.
- Anonymity: A bank cannot link a coin to the honest owner of the coin without the trustee's help.
- Unforgeability: Only the authorized bank can issue coins.
- Traceability: The bank can reveal the identity of customer (with the trustee's help) if the same e-coin is spent twice.
- Efficiency: The e-cash system must be efficient in terms of the storage space and the computation.

2.2. *Elliptic Curve Cryptography*

Elliptic Curve Cryptography (ECC) was firstly proposed in 1985 by Miller (1986) and Koblitz (1987). The elliptic curve cryptosystems are based on the elliptic curve logarithm problem over a finite field. Unlike other popular cryptosystems such as DSA, RSA or ElGamal, the elliptic curve cryptosystem is much more difficult to break at equivalent key lengths.

Table 1 compares the key sizes for three different cryptosystems to encryption for comparable levels of security against brute-force attacks. What makes this table all the more significant is that for comparable key lengths the computational cost of RSA and elliptic curve cryptosystems are comparable. Because of the much smaller key sizes, the elliptic curve cryptosystems can be implemented on smartcards without mathematical coprocessors.

An elliptic curve over a finite field F_p of characteristic greater than three can be constructed by choosing of two variables a and b within the field F_p.

DEFINITION 1. The elliptic curve is the set of *points* (x, y) *which satisfy the elliptic curve equation* $y^2 = x^3 + ax + b(\operatorname{mod} p)$*, where* $x, y \in F_p$*, together with a special point ("point at infinity") denoted* O *and* $4a^3 + 27b^2 \neq 0(\operatorname{mod} p)$.

Table 1

A comparison of key sizes with three different cryptosystems

Symmetric encryption (key size – in bits)	RSA and Diffie–Hellman (key size – in bits)	Elliptic curve (key size – in bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

The elliptic curve group is an additive abelian group with the point O which is the identity element. The formulas for addition of two points on an elliptic curve over a finite field F_p of characteristic greater than three are given as follows. Let $P(x_1, y_1)$ and $Q(x_2, y_2)$ be elements of the elliptic curve group. Then $P + Q = (x_3, y_3)$, where

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = \lambda(x_1 - x_3) - y_1, \end{cases}$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q, \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{if } P = Q. \end{cases}$$

Next, we give a definition of the elliptic curve discrete logarithm problem (Menezes, 1993).

DEFINITION 2. Let E be an elliptic curve defined over a finite field F_p and let $P \in E(F_p)$ be a point of order n. Given $Q \in E(F_p)$, the elliptic curve discrete logarithm problem is to find the integer l, $0 \leqslant l \leqslant n - 1$, such that $Q = l \cdot P$.

Elliptic curve cryptography is particularly useful in applications where the memory, the bandwidth and/or the computational power is limited (e.g., wireless communications, smartcards).

3. Our E-Cash Transfer System

The proposed e-cash transfer system consists of six phases: initialization phase, registration phase, withdrawal phase, payment phase, deposit phase, and tracing phase. In this paper we use the notations from Table 2.

Table 2

Notations used in our system

Notation	Description
U	The entity which is T – trusted third party, B – bank, C – customer
Q_U, R_U	Public keys of the entity U
x_U, k_U	Private keys of the entity U
Q'_C, R'_C	Public keys of the customer
x'_C, k'_C	Private keys of the customer
A_W	x – coordinate of the point W
P	Point on the elliptic curve
$l \cdot P$ (or lP)	Point multiplication operation
P+*Q*	Point addition operation
$H(), H'()$	One-way hash functions (SHA-1 or MD5)
p	A large prime number
q	A large prime number such that $q\|\#E(F_p)$.
$\|q\|$	The bit length of q
Z_p^*	The multiplicative group
$E(F_p)$	Elliptic curve defined over a finite field F_p
AMOUNT	The transfer funds amount
ACC	Customer's account in the bank
||	The concatenation operation

3.1. *Initialization Phase*

Let p be a large prime number. Choose an elliptic curve E defined over a finite field F_p of characteristic greater than three and calculate the order of the elliptic curve $\#E(F_p)$. Let q be a large prime number such that $q|\#E(F_p)$. Let P be a point of order q on the elliptic curve E.

Let H and H' be collision-resistant hash functions where:

$$H\colon \{0,1\}^* \rightarrow \{0,1\}^{|q|/2}, \tag{1}$$

and

$$H'\colon \{0,1\}^* \rightarrow Z_p^*. \tag{2}$$

The bank generates the following parameters:

1. Private key x_B: picks a random secret number x_B from the interval $[2, q-1]$.
2. Public key Q_B: computes the point $Q_B = x_B P$.
3. The private key of the bank is x_B.
4. The corresponding public key is Q_B.

The customer generates his parameters:

1. Static private key x_C: picks a random secret number x_C from the interval $[2, q-1]$.
2. Static public key Q_C: computes the point $Q_C = x_C P$.
3. The static private key of the customer is x_C.
4. The corresponding static public key is Q_C.

The trusted third party generates his parameters:

1. Static private key x_T: picks a random secret number x_T from the interval $[2, q-1]$.
2. Static public key Q_T: computes the point $Q_T = x_T P$.
3. The static private key of the customer is x_T.
4. The corresponding static public key is Q_T.

Also, the customer is identified by the bank and the bank opens customer's account ACC and sends it to the customer.

We assume that communication between parties (customer–trusted third party, customer–bank, customer–merchant, merchant–bank, bank–trusted third party) is secure, i.e., private and authentic.

3.2. *Registration Phase*

The registration phase involves the customer and the trusted third party. In this phase , we describe the transfer message between the customer and the trusted third party. The customer generates and registers a public key to the trusted third party.

1. The customer randomly generates a private key $x'_C \in [2, q-1]$. The corresponding public key is $Q'_C = x'_C P$. Customer also randomly generates the ephemeral keys pair:
   - an ephemeral private key $k'_C \in [2, q-1]$;
   - the corresponding public key $R'_C = k'_C P$.

   The customer generates the signature:

   $$\mathrm{SIG}_C = x_C H\big(A_{R'_C} \| \mathrm{ID}_C \| A_{Q'_C}\big) + k'_C,$$

   where ID_C is the identity of the customer, $A_{R'_C}$ is x – coordinate of the point R'_C and $A_{Q'_C}$ is x – coordinate of the point Q'_C. Then the customer sends SIG_C and ID_C to the trusted third party.
2. The trusted third party verifies the signature SIG_C:

   $$\mathrm{SIG}_C \cdot P = H\big(A_{R'_C} \| \mathrm{ID}_C \| A_{Q'_C}\big) Q_C + R'_C.$$

   If the above equation holds then the trusted third party accepts. The trusted third party also randomly generates keys pair:
   - an ephemeral private key $k_T \in [2, q-1]$;
   - the corresponding public key $R_T = k_T P$.

The trusted third party generates the signature:

$$\mathrm{SIG}_T = x_T H\big(A_{R_T} \| A_{Q'_C}\big) + k_T, \tag{3}$$

where A_{R_T} is x-coordinate of the point R_T.
The trusted third party sends SIG_T and R_T to the customer. Finally, the trusted third party stores ID_C, SIG_T, SIG_C and Q'_C in his database.

3. The customer verifies the signature SIG_T as follow:

$$\mathrm{SIG}_T \cdot P = H\big(A_{R_T} \| A_{Q'_C}\big) Q_T + R_T.$$

If the above equation holds then the customer accepts, otherwise rejects it.

3.3. *Withdrawal Phase*

The withdrawal subprotocol involves the customer and the bank. First, the customer proves his identity to the bank using the elliptic curve version of the signature scheme of Shao (2007). Then, the bank uses the blind signature scheme of the elliptic curve version of Shao's signature scheme (Shao, 2007) in order to withdraw a coin from the bank.

1. The customer sets his electronic cash requirement:

$$M = H'\big(\mathrm{ID}_C \| \mathrm{AMOUNT}\big), \tag{4}$$

where ID_C is the identity of the customer and AMOUNT is the transfer funds amount. Then, the customer chooses a random number k from the interval $[2, q-1]$ and computes:

$$V = kPH'(M),$$
$$h_C = H(M \| A_V),$$
$$s_C = \big(k - h_C x_C\big) \bmod q,$$

where A_V is x-coordinate of the point V.
The customer sends M and his signature (h_C, s_C) to the bank.

2. The bank verifies the signature (h_C, s_C) as follows: the bank computes $T_C = (h_C Q_C + s_C P)H'(M)$ and $h'_C = H(M \| A_{T_C})$, where A_{T_C} is x-coordinate of the point T_C. If $h'_C = h_C$ the bank accepts the signature, otherwise reject it.
3. The bank selects $k' \in [2, q-1]$, computes the point $R' = k'P$ and sends R' to the customer.
4. The customer randomly selects $\alpha, \beta \in [2, q-1]$ and computes:

$$R = \alpha P + \beta R', \tag{5}$$
$$h = H(M \| A_R), \tag{6}$$

where A_R is x-coordinate of the point R. The customer also computes $m' = h\beta^{-1} \bmod q$ and sends the value m' to the bank.

5. The bank computes:

$$s' = \big(k' - m'x_B\big) \bmod q, \tag{7}$$

and sends it to the customer.

6. The customer computes:

$$s = (s'\beta + \alpha) \bmod q, \tag{8}$$

$$R'' = hQ_B + sP, \tag{9}$$

$$h' = H(M \| A_{R''}), \tag{10}$$

where $A_{R''}$ is x-coordinate of the point R''. If $h = h'$, the blind signature (h, s) of the coin is valid, otherwise it is invalid.

7. The bank stores ID_C and m' and withdraws the coin value from the customer's account ACC.

3.4. *Payment Phase*

The payment phase involves the customer and the merchant and should be done through a secure channel (i.e., data privacy and integrity). The customer pays the withdrawn coin to the merchant in return for goods.

Step 1. The merchant sends challenge $C_m = H(\mathrm{ID}_m \| T_m)$ to the customer, where ID_m is the merchant's identity and T_m is the recorded time of the transaction.

Step 2. The customer chooses a random $k_C \in [2, q-1]$ and computes:

$$R_C = k_C P, \tag{11}$$

$$S_C = k_C - x'_C H\big(\mathrm{ID}_m \| C_m \| Q'_C \| h\big). \tag{12}$$

The customer sends S_C, SIG_T, (h, s) and R_C to the merchant.

Step 3. The merchant checks if (S_C, h, s) is already in its database for preventing the double spending. If not, the merchant verifies the signatures S_C, SIG_T, (h, s) as follows:

$$R_C = S_C P + H\big(\mathrm{ID}_m \| C_m \| Q'_C \| h\big) Q'_C, \tag{13}$$

$$\mathrm{SIG}_T \cdot P = H\big(A_{R_T} \| A_{Q'_C}\big) Q_T + R_T, \tag{14}$$

$$h = H\big((hQ_B + sP) \| h\big), \tag{15}$$

where A_{R_T} is x-coordinate of the point R_T and $A_{Q'_C}$ is x-coordinate of the point Q'_C.

If (13), (14) and (15) hold then the merchant accepts the signatures and stores S_C, SIG_T, (h, s), Q'_C, C_m in its database. The merchant accepts the e-cash and sends goods to the customer. Otherwise, the merchant rejects it.

3.5. *Deposit Phase*

The deposit phase involves the merchant and the bank. In this phase the merchant has to deposit the received e-coins to the bank.

Step 1. The merchant sends S_C, SIG_T, (h, s), R_C and C_m to the bank.

Step 2. The bank verifies the signature as given in (13), (14) and (15). If these equations are not valid, the bank terminates the transaction. If (13), (14) and (15) are valid, the bank checks if h and Q'_C obtained from the merchant exist in its database. If they exist, then the bank finds the signature S'_C for the deposited coin in its database and sends it to the merchant. In this case the bank detects a coin which is double-deposit or the coin is double spending.

Step 3. The merchant receives S'_C from the bank and verifies whether:

$$S'_C = S_C.$$

If the above equation holds, then the merchant rejects transaction. Otherwise, the merchant sends S_C, ID_m, C_m and T_m to the bank.

Step 4. The bank verifies the signature S_C as follows:

$$R_C = S_C P + H\big(\mathrm{ID}_m \| C_m \| Q'_C \| h\big) Q'_C.$$

If this signature is valid, the bank accepts the coin and deposits the cash to the merchant's account. The bank stores S_C, SIG_T, (h, s), Q'_C and ID_m in its database and the merchant sends the goods to the customer.

Step 5. If the bank finds out that h and S_C has been stored before but different T_m and C_m, then the coin has been double spending. In this case, the bank performs the tracing procedure in order to detect the identity of the double spender.

3.6. *Tracing Phase*

The tracing phase involves the bank and the trusted third party. In this phase the bank cooperates with the trusted third party in order to detect the identity of the double spender. The trusted third party reveals the owner (the double spender) of a paid coin from the payment phase.

Step 1. The bank sends R_C, C_m and the signatures S_C, SIG_T, (h, s) and S'_C from its database to the trusted third party.

Step 2. The trusted third party verifies the signatures S_C, SIG_T, (h, s) and S'_C as follows:

$$\begin{aligned}
&R_C = S_C P + H\big(\mathrm{ID}_m \| C_m \| Q'_C \| h\big) Q'_C,\\
&\mathrm{SIG}_T \cdot P = H\big(A_{R_T} \| A_{Q'_C}\big) Q_T + R_T,\\
&h = H\big((hQ_B + sP) \| h\big),\\
&R_C = S'_C P + H\big(\mathrm{ID}_m \| C_m \| Q'_C \| h\big) Q'_C,
\end{aligned}$$

where A_{R_T} is x-coordinate of the point R_T and $A_{Q'_C}$ is x-coordinate of the point Q'_C.
The trusted third party detects the identity of double spender, ID_C and Q'_C, and sends them to the bank.

Step 3. The bank has to freeze the double spender's account in order to prevent double spending.

4. Analysis and Discussions

In this section we discuss aspects of security and efficiency of our e-cash transfer system.

4.1. *Mutual Authentication*

The customer and the bank authenticate each other in the withdrawal phase. So, the bank give a valid ecoin to a legal customer. If the customer does not verify the validity of the ecoin received from the bank in the withdrawal phase, then the customer may therefore obtain a forged ecoin from an adversary. If an adversary wants to masquerade as the customer to send the signature (h'_C, s'_C) to the bank, the bank will reject it since $(h'_C, s'_C) \neq (h_C, s_C)$. Also, an adversary cannot compute k from the equation $V = kPH'(M)$ because he must solve the elliptic curve discrete logarithm problem described in Definition 2.

4.2. *The Anonymity of Our System*

In the withdrawal phase we use the blind signature scheme of the elliptic curve version of Shao's signature scheme (Shao, 2007). The blind digital signature scheme is a key tool for constructing various anonymous electronic cash systems. In an anonymous electronic cash system, even an all powerful agent that collaborates with the bank and any coalition of the customers cannot link payments to withdrawals.

Proposition 1 (Anonymity): The bank cannot link the blind e-cash with the identity of the customer.

Proof. The validity of the blind elliptic curve signature (h, s) for the coin follows from:

$$h = H\big((hQ_B + sP)\|h\big),$$

which is equivalent with

$$R'' = R.$$

The derivation of the verification is shown as follows:

$$\begin{aligned}
R'' &= hQ_B + sP \\
&= hQ_B + (s'\beta + \alpha)P \\
&= hQ_B + s'\beta P + \alpha P \\
&= hQ_B + \big(k' - m'x_B\big)\beta P + \alpha P \\
&= hQ_B + \big(k' - h\beta^{-1}x_B\big)\beta P + \alpha P \\
&= hQ_B + \beta k'P - h\beta^{-1}\beta x_B P + \alpha P \\
&= hQ_B + \beta R' + \alpha P - hQ_B \\
&= \beta R' + \alpha P \\
&= R\ (\text{see } (5)).
\end{aligned}$$

Because $R'' = R$ results that the signature (h, s) is a valid blind signature issued by the bank. Since H is one-way hash function, the bank can't recover the original message M from the following equation $m' = H(M\|A_R)\beta^{-1} \bmod q$. The original message M is protected by the one-way hash function H. It is computationally infeasible to derive M from the value $H(M\|A_R)$. Also, because H' is one-way hash function, it is computationally infeasible to obtain AMOUNT from the value $H'(\mathrm{ID}_C\|\mathrm{AMOUNT})$. If an adversary has the points P, R and R' he cannot compute α and β from the equation $R = \alpha P + \beta R'$ because also he must solve the elliptic curve discrete logarithm problem.

4.3. *The Correctness of the Signatures*

In the registration phase the customer generates the signature:

$$\mathrm{SIG}_C = x_C H\big(A_{R'_C}\|\mathrm{ID}_C\|A_{Q'_C}\big) + k'_C,$$

and the trusted third party generates his own signature:

$$\mathrm{SIG}_T = x_T H\big(A_{R_T}\|A_{Q'_C}\big) + k_T.$$

We prove that the both signatures SIG_C and SIG_T are valid.

Proposition 2 (Correctness): If the customer and the trusted third party follow the registration phase and accept it, then the signatures generates by the customer, respectively by the trusted third party, SIG_C and SIG_T, are correctly.

Proof. The verification equation for SIG_C is:

$$\mathrm{SIG}_C \cdot P = H\big(A_{R'_C}\|\mathrm{ID}_C\|A_{Q'_C}\big)Q_C + R'_C.$$

Obviously, the relation follows from:

$$\begin{aligned}\mathrm{SIG}_C \cdot P &= x_C H\big(A_{R'_C} \| \mathrm{ID}_C \| A_{Q'_C}\big)P + k'_C P\\ &= x_C P H\big(A_{R'_C} \| \mathrm{ID}_C \| A_{Q'_C}\big) + R'_C\\ &= H\big(A_{R'_C} \| \mathrm{ID}_C \| A_{Q'_C}\big)Q_C + R'_C.\end{aligned}$$

Also, the verification equation for SIG_T is:

$$\mathrm{SIG}_T \cdot P = H\big(A_{R_T} \| A_{Q'_C}\big)Q_T + R_T.$$

The derivation of the verification is described as follows:

$$\begin{aligned}\mathrm{SIG}_T \cdot P &= \big(x_T H\big(A_{R_T} \| A_{Q'_C}\big) + k_T\big) \cdot P\\ &= x_T P H\big(A_{R_T} \| A_{Q'_C}\big) + k_T P\\ &= H\big(A_{R_T} \| A_{Q'_C}\big)Q_T + R_T.\end{aligned}$$

4.4. *Non-Forgeability of the Coins*

The customers of the e-cash transfer system and the trusted third party, even in collaboration, should not be able to mint coins without express participation of the bank. The hardness of forgery in our system is determined by security parameters p and q. We let p be at least 512 bits and q be 160 bits.

Proposition 3 (Non-forgeability): The customer and other third parties can't generate the same signature of the bank if they follow the blind signing scheme in the withdrawal phase.

Proof. Because the signature scheme (Shao, 2007) is secure against existential forgery, this allows only the legal bank to generate the signature for the cash. For the one-way hash function H', when the value of M is given, it is easy to compute $H'(M)$. However, if the value of $H'(M)$ is given, computing M is very difficult. Also, because the hash function H' is collision-resistant, it is infeasible to generate two distinct inputs with matching outputs. So, the customer or an unauthorized person (criminal) cannot find a value $M' \neq M$ with $H'(M) = H'(M')$. If an adversary has the points Q_B, P and R' he cannot compute x_B and k' from the equations $Q_B = x_B P$ and $R' = k'P$ because also he must solve the elliptic curve discrete logarithm problem.

4.5. *Traceability*

If any customer uses the same coin twice, then with the help of the trusted third party, the bank can find out this illegal transaction by checking the double-spent ecoin stored in the database because the trusted third party can easily reveal the identity of the customer by verifying the signatures S_C, SIG_T, (h, s) and S'_C.

Table 3

Comparison of e-cash systems – Storage space

	Our	Au	Canard	Chou	Camenisch
Withdrawal phase	880 bits	8160 bits	6420 bits	2208 bits	1012 bits
Payment phase	720 bits	5188 bits	30740 bits	1184 bits	18432 bits
Deposit phase	820 bits	5164 bits	27648 bits	1184 bits	3098 bits

4.6. *Storage Space and Computational Time*

We evaluate the storage space and computational time of the costly operations. Tables 3 and 4 summarize the storage space and computation cost respectively, of different protocols of our e-cash transfer system and the payment systems (Chou *et al.*, 2009; Au *et al.*, 2008; Canard and Gouget, 2007; Camenisch *et al.*, 2005). The overall efficiency is improved in our e-cash transfer system compared to Chou *et al.*'s (2009) system Au *et al.*'s (2008) system, Canard *et al.*'s system (Canard and Gouget, 2007) and Camensich *et al.*'s (2005) system in terms of the storage space and the computation cost. Suppose that our e-cash transfer system has a point P of 160 bits, p of 512 bits, q of 160 bits and the size of each party's ID is 20 bits. For a moderate value $L = 10$ and $t = 40$, the payment protocol (Canard and Gouget, 2007) requires 1673 multi-based exponentiations and a total bandwidth of 30740 bits. The payment protocol (Au *et al.*, 2008) requires 34 multi-based exponentiations, 14 pairings and a total bandwidth of 5188 bits. In contrast, the payment protocol in our e-cash system requires 4 multi-based exponentiations and a total bandwidth of 720 bits.

For example, when a customer pays out to a merchant in the payment phase, the customer sends the following message to the merchant:

$$S_C,\ \mathrm{SIG}_T,\ (h, s),\ R_C.$$

The total size of this message is:

$$160 + 160 + (80 + 160) + 160 = 720\ \mathrm{bits}.$$

We note that one pairing has a cost about five 512-bit multi-based exponentiations.

5. Conclusions

In this paper we proposed a secure e-cash transfer system based on the elliptic curve discrete logarithm problem. We proved that our e-cash system satisfies the following security features: mutual authentication, non-forgeability of the coins, correctness, traceability, anonymity and efficiency. Consequently, we compared the proposed e-cash system with other existing payment systems, and the comparison results show that our system is more

Table 4

Comparison of e-cash systems – Computation cost

		Our	Au	Canard	Chou	Camenisch
Withdrawal phase	Multi-EXP	7	2156	5	12	18
	Pairing	0	22	0	8	0
Payment phase	Multi-EXP	4	34	1673	4	26
	Pairing	0	14	0	5	0
Deposit phase	Multi-EXP	3	10	14	4	16
	Pairing	0	0	0	5	0

efficient than others. Because the amount of communication between customer and merchant is about 720 bits, the proposed e-cash transfer system can be used in the wireless communications.

Acknowledgements. We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript.

References

Au, M. , Susilo, W., Mu, Y. (2008). Practical anonymous divisible e-cash from bounded accumulators. In: *Proceedings of Financial Cryptography and Data Security, Lecture Notes in Computer Science*, Vol. 5143. Springer, Berlin. pp. 287–301, 2008.

Camenisch, J., Hohenberger, S., Lysyanskaya, A. (2005). Compact e-cash. In: *Advances in Cryptology – EuroCrypt 2005*, pp. 302–321.

Canard, S., Gouget, A. (2007). Divisible e-cash systems can be truly anonymous. In: *Proceedings of EUROCRYPT 2007, Lecture Notes in Computer Science*, Vol. 4515, Springer, Berlin, pp. 482–497.

Chaum, D. (1983). Blind signature for untraceable payments. In: *Proceedings of Eurocrypt'82*, Plenum, New York, pp. 199–203.

Chaum, D., Fiat, A., Naor, M. (1990). Untraceable electronic cash. In: *Proceedings of the Crypto'88*, pp. 319–327.

Chou, J.S., Chen, Y.-L., Cho, M.-H., Sun, H.-M. (2009). A novel ID-based electronic cash system from pairings. In: *Cryptology ePrint Archive*, Report 2009/339. Available at `http://eprint.iacr.org/`.

De Santis, A., Ferrara, A.L., Masucci, B. (2007). An attack on a payment scheme. *Information Sciences*, 178, 1418–1421.

Elgamal, T. (1985). A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), 469–472.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48, 203–209.

Lee, M., Ahn, G., Kim, J., Park, J., Lee, B., Kim, K., Lee, H. (2002). Design and implementation of an efficient fair off-line e-cash system based on elliptic curve discrete logarithm problem. *Journal of Communications and Networks*, 4, 81–89.

Li, C.T., Hwang, M.S. (2010). An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, 33(1), 1–5.

Liu, J., Huang, S. (2010). Identity-based threshold proxy signature from bilinear pairings. *Informatica*, 21(1), 41–56.

Menezes, A. (1993). *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic, Dordrecht.

Miller, V. (1986). Uses of elliptic curves in cryptography. In: *Advances in Cryptology, Proceedings of Crypto'85, Lecture Notes in Computer Sciences*, Vol. 218, Springer, Berlin, pp. 417–426.

NIST (2009). *Federal Information Processing Standards*. Digital signature standard (DSS), Publication 186-3.

Okamoto, T. (1995). An efficient divisible electronic cash scheme. In: *Proceedings of Crypto'95*, pp. 302–318.

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

Popescu, C. (2006). An electronic cash system based on group blind signatures. *Informatica*, 17, 551–564.

Popescu, C. (2009). An anonymous mobile payment system based on bilinear pairings. *Informatica*, 20(4), 579–590.

Raulynaitis, A., Sakalauskas, E., Japertas, S. (2010). Security analysis of asymmetric cipher protocol based on matrix decomposition problem. *Informatica*, 21(2), 215–228.

Rivest, R.L., Shamir, A., Adelman, L. (1978). A method for obtain digital signatures and public-key cryptosystem. *Communication on ACM*, 21(2), 120–126.

Sakalauskas, E., Tvarijonas, P., Raulynaitis, A. (2007). Key agreement protocol (kap) using conjugacy and discrete logarithm problems in group representation level. *Informatica*, 18(1), 115–124.

Shao, Z. (2007). A provably secure short signature scheme based on discrete logarithms. *Informations Sciences*, 177, 5432–5440.

Trolin, M. (2005). A universally composable scheme for electronic cash. In: *Proceedings of Indocrypt*, pp. 347–360.

Tseng, Y.M., Wu, T.-Y., Wu, J.-D. (2008). A pairing-based user authentication scheme for wireless clients with smart cards. *Informatica*, 19(2), 285–302.

Xiong, H., Li, F., Qin. Z. (2010). A provably secure proxy signature scheme in certificateless cryptography. *Informatica*, 21(2), 277–294.

C. Popescu received the PhD degree in computer science (cryptography) at the Babes-Bolyai University, Cluj Napoca, Romania. Since 2005 he is a professor at the Department of Mathematics and Computer Science, University of Oradea, Romania. His research interests include cryptography, network security, group signatures, security protocols and electronic payment systems.

Saugi e-pinigų perlaidų sistema, naudojanti eliptinių kreivių diskretinių logaritmų problemą

Constantin POPESCU

Elektroninė prekyba esti palyginus nauja ir pastoviai kintanti informacinių technologijų bei verslo valdymo sritis. Straipsnyje pasiūlyta saugi e-pinigų perlaidų sistema, kurioje panaudota eliptinių kreivių kriptografija. E-pinigų perlaidos sistemos vartotojų apsaugos užtikrinimui panaudotas eliptinių kreivių aklojo parašo algoritmas bei reikalinga patikima trečioji pusė, kuri padeda nustatyti nusikalstamus veiksmus.