An Efficient Off-Line E-Cash System based on Signcryption without Bilinear Pairings

Hassan M. Elkamchouchi
Elec. Eng. Dept, Fac. of Eng,
Alexandria Univ.
Alexandria
Egypt

Eman F. Abou Elkheir
Elec. Eng. Dept, Fac. of Eng,
Kafr Elsheikh Univ.
Kafr Elsheikh
Egypt

Yasmine Abouelseoud
Eng. Math. Dept, Fac. of Eng,
Alexandria Univ.
Alexandria
Egypt

ABSTRACT
There is an increased activity in research and development conducted to improve current payment systems in parallel with the progress of Internet. Signcryption combines the functionalities of encryption and digital signing in a single logical step. It is a cryptographic primitive that provides confidentiality, integrity, authentication and non-repudiation. Identity-based cryptography serves as an efficient alternative to the traditional certificate-based cryptosystems. This paper introduces an efficient electronic payment system based on signcryption without bilinear pairings. This e-cash system is more efficient than other schemes employing bilinear pairings and involves less computational cost. In the proposed protocol, the token is issued and authenticated by the bank to prevent double spending problem. The customer delegates his signing capability to the merchant. The bank verifies the identities of both the original signer (customer) and the proxy signer (merchant) and ensures the originality of the transaction. Unlike the existing e-payment systems, the problem of double spending of e-cash does not arise because each transaction is made uniquely identifiable. Hence, no separate protocol is needed to check double-spending. The performance and the security analysis of the proposed e-cash system are discussed revealing its strength from the viewpoint of security and efficiency with regard to computations required.

General Terms
Cryptography, Security.

Keywords
E-cash payment system, Signcryption , Bilinear Pairings, E-Commerce Security

1. INTRODUCTION
In the last years, cryptographic protocols and network technologies have experienced rapid development [1, 2, 3, 4]. Electronic commerce is one of the most important applications over the Internet. Smart card based remote user authentication [5, 6] is the simplest and most convenient authentication mechanism for insecure networks. Customers' privacy must be protected if they are involved in legal commercial transactions or payments. Blind signature schemes [7] have been widely used to protect the right of customer's privacy in untraceable electronic cash systems [8].However, it is easy to make multiple copies of the electronic coin, which is in the form of number strings. Therefore, anonymity revocation mechanisms 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 [8] proposed the first untraceable electronic cash system based on blind signatures in 1982. Several extensions have been proposed, which provide functionalities such as anonymity revocation and double spending prevention [9, 10, 11].

In the field of e-cash systems, the notions of on-line systems and off-line systems 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 computational costs. In contrast, off-line e-cash systems usually operate in dual mode; verifying high cost transactions on-line, while the rest of the payments are processed in batch mode by the bank.

The first off-line electronic cash system was introduced by Chaum in 1990 [12] and then further developed by Popescu in 2006 and 2009 [13,14], by De Santis in 2007 [15], by Chou in 2009 [16], and by Au in 2008 [17]. In these papers, the bank is 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. 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.

Miller [18] and Koblitz [19] introduced Elliptic Curve Cryptography (ECC), which has increasingly attracted the attention of researchers in recent years due to its shorter key length requirement in comparison with other public key cryptosystems based on finite fields; such as DSA [20], ElGamal [21] and RSA [22]. For example, the 160-bit elliptic curve version of the 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. An E-cash transfer system using blind signatures based on the elliptic curve discrete logarithm problem has been recently proposed by Popescu in 2009 [13]. Another scheme employing elliptic curves is that proposed by Debasis Giri and Arpita Mazumdar in 2013 [23].

This paper introduces a new e-cash system employing identity-based signcryption without bilinear pairings. The proposed e-cash protocol is based on the signcryption scheme in [24] and the proxy signcryption scheme in [25] by which the customer delegates his signing rights to the merchant.

The rest of the paper is organized as follows. In the next section, the requirements for a secure e-cash system are summarized. The basic steps of the proposed e-cash system are presented in Section 3 and a more detailed description of the system appears in Section 4. The performance and security of the proposed system are analyzed in Section 5. Finally, Section 6 concludes the paper.

2. Security Requirements for any E-cash System

The most important requirements [14, 16] for a secure electronic cash transfer system are summarized below. They are:

2.1 Mutual authentication

Two parties can authenticate each other correctly.

2.2 Correctness

One can ensure the correctness and integrity of messages transmitted by the other designated party.

2.3 Unforgeability

Only the authorized bank can issue coins.

2.4 Traceability

The bank can reveal the identity of customer (with the trustee's help) if the same e-coin is spent twice.

2.5 Efficiency

The e-cash system must be efficient in terms of storage requirements and computations.

3. The Proposed E-cash system

An e-cash system is a set of entities with their interactions, exchanging e-cash and goods. Our system has three entities:

Customer(C): purchases goods or acquires services from the merchant using the e-cash.

Merchant(M): sells goods or services to the customer, and deposits the e-cash to the bank.

Bank(B): issues the e-cash and manages the accounts for customers and merchants.

There are also three protocols in the system: withdrawal, payment and deposit. The customer withdraws a token from the bank and pays the token to the merchant. The token is readable only. The merchant gets the token from the customer and deposits it in the bank. The bank manages customer accounts, issues and updates tokens. No separate protocol is required to trace a dishonest customer. The proposed protocol is shown in Figure 1. In a transaction, the following events take place:

At first, setup is done by a central authority (CA).

1. (C sends to B) The customer requests the withdrawal of an electronic coin (token) from his account in the bank.
2. (B sends to C) Bank issues a signed token to C and securely sends it to C and deducts the appropriate amount from the customer's account.
3. (C sends to M) C receives the token, chooses an item from M's home page and sends the signcrypted order information (OI) to M. C also sends the hash form of the token information signed which later helps B to verify C.
4. (M sends to C) M sends the hash value of SEQNO of the token signed as an acknowledgement to C. M also delivers the products to the customer. Transfer of products may be immediate (for intangible goods) or delayed (for tangible goods).
5. (C sends to M) The customer delegates his signing rights to the merchant and the bank acts as the verifier.
6. (M sends to B) Merchant appends price details, own account number (MAC) to OI and forwards the signcrypted (with proxy signature) modified OI to B along with the signed hashed token.
7. (Verification by B) Bank verifies both the (proxy signer) merchant as well as the (original signer) customer and ensures that OI is genuinely placed by the customer. The bank retrieves the customer token details by matching (TOKEN _ID||SEQNO). It also performs hashing and checks whether the sent hashed token value is identical to the hash value of the stored token.
8. (B sends to C) After verification, the bank sends update information to C via e-mail. Updated token information is kept in a server. A valid customer after a certain period of time can download the updated token details (Token value, SEQNO, TS).
9. (B sends to M) Bank credits the account of the merchant and sends an acknowledgement to it.

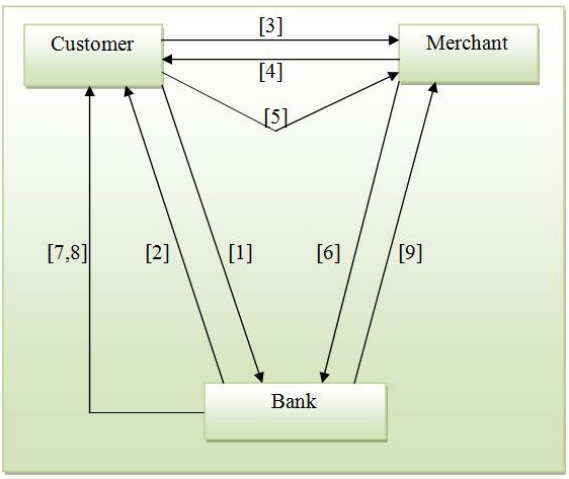

Fig. 1The proposed e-cash protocol steps

In the proposed protocol, the format of the token information (TI) is shown below.

A/C NO	TOKENID	SEQNO	EXPIRES	TS	VALUE	PROPOS

A/C NO: Account number of customer

TOKEN_ID: Unique identification number of a token

SEQNO: Unique sequence number for each transaction

EXPIRES: Lifespan of a token.

TS: Time stamp of a transaction.

VALUE: Monetary value.

Moreover the format of Order Information (OI) is shown below.

TOKENID	ITMCODE	SEQNO

where ITMCODE is a unique code of an item for sale.

4. The Detailed Description of the Proposed Protocol

4.1 Setup

Given a security parameter k, the key generator center (KGC) chooses q a large prime number, where $q > 2^{160}$, (a, b) two integer elements which are smaller than q and satisfy $(4a^3 + 27b^2) \bmod q \neq 0$. E is the selected elliptic curve over a finite field $F_q : y^2 = (x^3 + ax + b) \bmod q$. P is the base point of order n of the group G (the set of points on E generated by P). Also, O is the point at infinity and n is the order of the point P , where n is a prime, $n.P = O$ and $n > 2^{160}$. The KGC selects a cryptographic one way hash function $H : \{0,1\}^k \times Z_q \rightarrow Z_q$.

4.2 Key Generation

The KGC selects a random number s as the master key and computes the master public key $R = s.P$. The KGC keeps s secret and publishes the system parameters $params: \{k, G, P, R, H\}$. The KGC generates the secret and public keys of the customer, the merchant and the bank then sends the secret keys to their owners through a secure channel and publishes the public keys and the identities. The KGC calculates the customer, the merchant and the bank secret keys respectively as follows: $d_c = (h(ID_c).s) \bmod n$, $d_M = (h(ID_M).s) \bmod n$ and $d_B = (h(ID_B).s) \bmod n$ .The KGC calculates the public keys as follows: $Q_C = x_C.R$ (the customer public key), $Q_M = x_M.R$ (the merchant public key) and $Q_B = x_B.R$ (the bank public key).

4.3 Payment Protocol

Step 1: The customer C sends a request for a token indicating the amount and his account number with the bank B (token issuing authority). This information is sent securely by signcrypting the message, say $req = (A/cno, Amt)$

The customer C chooses a random number w and computes:

- $k_C = ID_B.w.Q_B = (\rho, \upsilon)$
- Split ρ into k_{1C}, k_{2C}
- $c = E_{k_{1C}}(req)$
- $r = hash(req, k_{2C}, ID_C, ID_B)$
- $s = (ID_B.w - r.x_C.ID_C) \bmod n$
- C sends (r, c, s) to the bank.

Step 2: After receiving it, the bank verifies the authenticity of the received message. It performs the following steps.

The bank recovers the key k_C as follows:

$$k_C = s.Q_B + r.ID_C.x_B.Q_C = ID_B.w.Q_B = (\rho, \upsilon)$$

- Split ρ into k_{1C}, k_{2C}
- $\bar{req} = D_{k_{1C}}(c)$
- $\bar{r} = hash(req, k_{2C}, ID_C, ID_B)$
- If $\bar{r} = r$, then the bank accepts the signature.

The correctness of the key recovery equation is demonstrated below:

$$k_C = s.Q_B + r.ID_C.x_B.Q_C = ID_B.w.Q_B = (x, y)$$

$$= s.Q_B + r.ID_C.x_B.Q_C$$

$$= ID_B.w.Q_b - r.x_C.ID_C.Q_B + r.ID_C.x_B.Q_C = ID_B.w.Q_B = (\rho, \upsilon)$$

If C is a valid account holder, then B issues a token (TI) and sends it to C securely by the following way:

- The bank B chooses a random number w_1 and computes $k_{1B} = ID_C.w_1.Q_C = (\rho_1, \upsilon_1)$
- Split ρ_1 into k_{11B}, k_{21B}
- $c_1 = E_{k_{11B}}(TI)$
- $r_1 = hash(TI, k_{21B}, ID_C, ID_B)$
- $s_1 = (ID_C.w_1 - r_1.x_B.ID_B) \bmod n$
- B sends (r_1, c_1, s_1) to the customer.

Step 3: Upon receiving this tuple, the customer verifies the authenticity of the received message. The customer proceeds as follows.

It recovers the key by computing

$$k_{1B} = s_1.Q_C + r_1.ID_B.x_C.Q_B = ID_C.w_1.Q_C = (\rho_1, \upsilon_1)$$

- Split ρ_1 into k_{11B}, k_{21B}
- $\bar{TI} = D_{k_{11B}}(c_1)$
- $\bar{r_1} = hash(TI, k_{21B}, ID_C, ID_B)$
- If $\bar{r_1} = r_1$, the customer accepts the token TI.

After verification, the customer (C) signcrypts the token and OI, then sends them to the merchant. Generation of the above information by C is described in details below.

The customer signcrypts the OI by carrying out the following steps:

- The customer C chooses a random number w_2 and computes $k_{2C} = ID_M.w_2.Q_M = (\rho_2, \upsilon_2)$
- Split ρ_2 into k_{12C}, k_{22C}
- $c_2 = E_{k_{12C}}(OI)$
- $r_2 = hash(OI, k_{22C}, ID_C, ID_M)$
- $s_2 = (ID_M.w_2 - r_2.x_C.ID_C) \bmod n$

Also, the customer signs the token information TI and sends them securely to the merchant as follows:

* The customer chooses a random number w_3 and computes $r_3 = (w_3 + x_C).R = (u_1, v_1)$
* computes $s_3 = (u_1 + h(TI).x_M) mod\ n$
* sends $(r_2, c_2, s_2, u_1, s_3, h(TI))$ to the merchant M.

Step 4:

The merchant checks the authenticity of the customer and retrieves the value of OI.

* The merchant M recovers the key k_{2C} by computing $k_{2C} = s_2.Q_M + r_2.ID_C.x_M.Q_C = ID_M.w_2.Q_M = (\rho_2, \upsilon_2)$
* Split ρ_2 into k_{12C}, k_{22C}
* $\bar{OI} = D_{k_{12C}}(c_2)$
* $\bar{r}_2 = hash(OI, k_{22C}, ID_C, ID_M)$
* If $\bar{r_2} = r_2$, the merchant verifies that the OI has been issued by a valid customer.

Also, the merchant authenticates that the signature has been issued by the customer as follows:

$h(TI).Q_C = (S_3 - u_1).R$

The correctness of the equation: $h(TI).Q_C = (S_3 - u_1).R$

Starting from the right hand side: $(S_3 - u_1).R = (u_1 + h(TI).x_C - u_1).R$

$= h(TI).x_C.R = h(TI).Q_C = LHS$

After successful verification, M sends the signature of SEQNO (part of token information OI) to C in the following manner:

* The merchant M chooses a random number w_4 and computes $r_4 = (w_4 + x_M).R = (u_2, v_2)$
* computes $s_4 = (u_2 + SEQNO.x_M) mod\ n$
* sends $(u_2, s_4, SEQNO)$ to the customer C as an acknowledgement

Step 5: The customer C verifies the merchant as follows: $SEQNOQ_M = (S_3 - u_2).R$

Then, the customer delegates his signing rights to the merchant.

The original signer (C) chooses a random number $d \in [1, q-1]$ and computes:

1- $T = ID_M d.R = (\alpha, \beta)$

2- $\sigma = (d.ID_M - x_C.h(\alpha, m_w.ID_C)) \bmod n$

C sends (α, σ, m_w) to the proxy agent (merchant M).

Step 6: The proxy (M) authenticates the original signer (customer) as follows:

If $\sigma.R + ID_C.h(\alpha, m_w).Q_C = T$, then the proxy computes the secret proxy key $skp \equiv (x_M + \sigma) \bmod n$. Otherwise, the proxy requests a new (α, σ, m_w)-tuple.

The correctness of the verification of equation is demonstrated below:

$RHS = \sigma.R + ID_C.h(\alpha, m_w).Q_C$

$= (ID_M.d - x_C.h(\alpha, m_w).ID_C).R + h(\alpha, m_w).ID_C.Q_C$

$= ID_M.d.R - x_C.h(\alpha, m_w).ID_C.R + ID_C.x_C.h(\alpha, m_w).R$

$= ID_M.d.R = T = LHS$

The merchant M appends its own account number (MAC), price with OI, i.e., MOI=(OI| MAC||price) and sends the bulk of the information to the bank. M creates a proxy signature on the individual transaction information on behalf of his customer and sends it securely as follows:

* The merchant M chooses random number w_5 and computes $k_p = ID_B.w_5.Q_B = (\rho_3, \upsilon_3)$
* Split ρ_3 into k_{13M}, k_{23M}
* $c_p = E_{k_{13M}}(MOI)$
* $r_p = hash(MOI, k_{23M}, ID_B, ID_M)$
* $s_p = (ID_B.w_5 - r_p.skp.ID_M) \bmod n$

At the end of the day, M forwards signed token (sent by customer) to the bank B.

* The merchant sends token information and the proxy signature $((u_1, s_3, h(TI)), \alpha, \sigma, m_w, c_p, r_p, s_p)$ to the bank B.

Step 7: The bank B verifies the customer signature on the token information (TI) that the merchant forwarded as follows:

The bank checks if $h(TI).Q_C \stackrel{?}{=} (S_3 - u_1).R$, otherwise the bank rejects the signature. Then, the bank verifies the proxy signer (M) as well as the original signer (C) and ensures that OI has been genuinely placed by the customer.

The bank recovers the key as follows: $k_p = s_p.Q_B + r_p.ID_M.x_B.(T + Q_M - ID_C.h(\alpha, m_w).Q_C)$

$= ID_B.w_5.Q_B = (\rho_3, \upsilon_3)$

* Split ρ_3 into k_{13M}, k_{23M}
* $\bar{MOI} = D_{k_{13M}}(c_p)$
* $\bar{r_p} = hash(m, k_2, ID_s, ID_r, ID_p)$
* If $\bar{r_p} = r_p$, the bank accepts the signature.

The correctness of the key recovery equation is shown below.

$$k_p = s_p.Q_B + r_p.ID_M.x_B.(T + Q_M - ID_C.h(\alpha, m_w).Q_C)$$

$$= (ID_B.w_5 - ID_M.r_p.skp).Q_B + ID_M.r_p.x_B.T + ID_M.r_p.x_B.Q_M$$
$$- ID_M.r_p.x_B.ID_C.h(\alpha, m_w)Q_C)$$
$$= (ID_B.w_5.Q_B - ID_M.r_p.(x_M + \sigma).Q_B + ID_M.r_p.x_B.T$$
$$+ ID_M.r_p.x_B.Q_M - ID_M.r_p.x_B.ID_C.h(\alpha, m_w)Q_C)$$
$$= ID_B.w_5.Q_B - ID_M.r_p.(x_M + ID_M.d - ID_C.x_C.h(\alpha, m_w)).Q_B$$
$$+ ID_M.r_p.x_B.T + ID_M.r_p.x_B.Q_M$$
$$- ID_M.r_p.x_B.ID_C.h(\alpha, m_w)Q_C)$$
$$= ID_B.w_5.Q_B - ID_M.r_p.x_M.Q_B - ID_M.r_p.ID_M.d.Q_B$$
$$+ ID_M.r_p.ID_C.x_C.h(\alpha, m_w).Q_B + ID_M.r_p.x_B.T$$
$$+ ID_M.r_p.x_B.Q_M - ID_M.r_p.x_B.ID_C.h(\alpha, m_w)Q_C)$$
$$= ID_B.w_5.Q_B - ID_M.r_p.x_M.x_B.P - ID_M.r_p.ID_M.d.x_B.P$$
$$+ ID_M.r_p.ID_C.x_C.h(\alpha, m_w).x_B.P + ID_M.r_p.x_B.ID_M.d.P$$
$$+ ID_M.r_p.x_B.x_M.P - ID_M.r_p.x_B.ID_C.h(\alpha, m_w)x_C.P)$$
$$= ID_B.w_5.Q_B = (\rho_3, \upsilon_3)$$

The bank retrieves the customer's token details by matching (TOKEN ID||SEQNO). Also, it performs hashing and checks whether the hashed token value sent by customer is the same as the hash value of the stored token.

Step 8: After verification, the bank sends update information to C via e-mail. Updated token information is kept in a server. A valid customer after a certain period of time can download the updated token details: Token value, SEQNO, TS.

Step 9: The bank credits the account of merchant and sends an acknowledgement to it.

5. Security and Performance Analysis

The security of the proposed e-cash protocol is based on the elliptic curve discrete logarithm problem (ECDLP) [26]. Up till now, the ECDLP is considered to be hard.

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined as follows. Let G and Q be two points on an elliptic curve and G is of order n and n is a prime. The point $Q = k.G$, where $k < n$. Given these two points G and Q, find the discrete logarithm of Q to the base G; that is, k.

5.1 Security Analysis

5.1.1 Mutual Authentication

The proposed protocol uses the signcryption primitive in which the message is efficiently signed and encrypted in a single logical step, so the communicating parties can authenticate each other easily. For example, in the withdrawal phase between the customer and the bank; the customer sends a signcrypted request (r, c, s) to the bank for the withdrawal of electronic coins from his account. If an adversary wants to masquerade as the customer to send an encrypted message to the bank, the bank will reject it since it will recover a wrong key and it cannot decrypt the ciphertext to obtain any meaningful information and surely authentication fails. Moreover, only the bank can recover the sent message as it is the only entity in possession of the secret key required.

5.1.2 Unforgeability

The unforgeability property holds in all exchanged messages due to the employment of secure (proxy) signature and (proxy) signcryption schemes [24,25].

5.1.3 Traceability:

In the proposed protocol, the bank verifies the original signer (C) and the proxy signer (M) and checks the validity of token. A token cannot be double spent as it is updated. In other e-payment protocols, detection of double spending involves a database search and the tracing phase involves the bank and a trusted third party in order to detect the identity of a double spender.

5.2 Comparative study

Table I shows the time abbreviations that will be used in the comparison table. Table II shows a comparison between the performance of the proposed protocol to that of the protocol in [23]. The comparison in table II shows that the proposed protocol is more efficient and has a lower computational cost than the protocol in [23] with bilinear pairings. Also, using the proposed protocol saves bandwidth, where the signcryption module involved in the proposed protocol reduces signaling overhead.

Table 1 Time Abbreviations

Symbol	Operation
$\mathbf{T_{EC\text{-}mult}}$	time required for executing multiplication operation on elliptic curve *E*
$\mathbf{T_{EC\text{-}add}}$	time required for executing addition operation on elliptic curve *E*
$\mathbf{T_{mult}}$	time required for executing modulus multiplication in a finite field
$\mathbf{T_h}$	time required for executing one way dispersed row function operation
$\mathbf{T_{encr}}$	time required by the system for executing encryption operation
$\mathbf{T_{decr}}$	time required by the system for executing decryption operation
$\mathbf{T_{exp}}$	time required for executing modulus exponential operation
$\mathbf{T_{pairings}}$	time of executing a bilinear pairing operation

6. Conclusion

In this paper, a new e-cash payment protocol based on signcryption has been introduced. It involves no bilinear pairings. The proposed protocol is presented together with its security analysis. It is also compared with another e-cash protocol in [23]. The comparative study shows that the proposed protocol is more efficient reducing the signaling overhead and the computational cost while achieving the same security properties of the protocol in [23].

Table 2 the comparison between the proposed protocol and the protocol in [23]

Step	The protocol in [23](with bilinear pairings)	The proposed protocol(without bilinear pairings)
1(Signcryption)	$3T_{EC\text{-}mult}$ $+2T_h$ $+3T_{pairings}$ $+1T_{EC\text{-}add}$ + $1T_{enc}$	$1T_{EC\text{-}mult}$ $+1T_h$ $+4T_{mult}$ + $1T_{enc}$
2(verifying Signcryption +signcryption)	$3T_{EC\text{-}mult}$ $+4T_h$+ $6T_{pairings}$+ $1T_{EC\text{-}add}$ + $1T_{enc}$+ $1T_{dec}$	$3T_{EC\text{-}mult}$ $+2T_h$ $+6T_{mult}$ $+1T_{EC\text{-}add}$ + $1T_{enc}$+$1T_{dec}$
3(verifying Signcryption +Signcryption +signature)	$6T_{EC\text{-}mult}$ $+6T_h$ $+8T_{pairings}$+ $2T_{EC\text{-}add}$ + $1T_{enc}$+ $1T_{dec}$	$4T_{EC\text{-}mult}$ $+3T_h$ $+7T_{mult}$ $+1T_{EC\text{-}add}$ + $1T_{enc}$+$1T_{dec}$
4(verifying signature +verifying Signcryption +signature)	$3T_{EC\text{-}mult}$ $+4T_h$+ $7T_{pairings}$+$1T_{EC\text{-}add}$ +$1T_{dec}$	$5T_{EC\text{-}mult}$ $+3T_h$ $+3T_{mult}$+ $1T_{EC\text{-}add}$+ $1T_{dec}$
5 (verifying signature +delegation signing rights)	$2T_{EC\text{-}mult}$ $+2T_h$+ $3T_{pairings}$+ $1T_{EC\text{-}add}$	$3T_{EC\text{-}mult}$+$1T_h$+$3T_{mult}$
6(verifying signature +proxy key +proxy Signcryption)	4TEC-mult +3Th +4Tpairings +3TEC-add + 1Tenc	3TEC-mult +1Th +5Tmult + 1Tenc
7(verifying signature +verifying proxy Signcryption	4Th+5Tpairings+1Tdec	5TEC-mult +3Th +3Tmult+ 3TEC-add + 1Tdec
Total	21TEC-mult +25Th +36Tpairings+ 9TEC-add+ 4Tenc+ 4Tdec	24TEC-mult +14Th +31Tmult+ 6TEC-add+ 4Tenc+4Tdec

7. REFERENCES

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

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

[3] 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.

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

[5] 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.

[6] 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.

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

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

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

[10] 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.

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

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

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

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

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

[16] 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/.

[17] 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.

[18] 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.

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

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

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

[22] 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.

[23] D. Giri and A. Mazumdar: A Secure Off-line Electronic Payment System Based on Bilinear Pairings and Signcryption. International Journal of Soft Computing and Engineering (IJSCE),ISSN: 2231-2307, Volume-2, Issue-6, January, 2013

[24] Hassan Elkamchouchi, Eman Abou El-kheir, and Yasmine Abouelseoud. An Efficient Identity-Based Signcryption Scheme Without Bilinear Pairings. Appear in the ninth international conference on computer engineering and systems(ICCES 2013) November, 2013

[25] Hassan Elkamchouchi, Eman Abou El-kheir, and Yasmine Abouelseoud. An Efficient Proxy Signcryption Scheme Based on the Discrete Logarithm Problem. International Journal of Information Technology, Modeling and Computing (IJITMC) Vol.1, No.2, May 2013

[26] D. Johnson, A. Menezes, and S. Vanstone: The elliptic curve digital signature algorithm (ECDSA) . International Journal of Information Security 1 (1) (2001) 36–63.