ECGSC: Elliptic Curve based Generalized Signcryption Scheme

Yiliang Han, Xiaoyuan Yang

Key Lab. on Network and Information Security of Armed Police Force
Department of Electronic Technology, Engineering College of Armed Police Force
Wujing Road, Xi'an, 710086, China
Yilianghan@hotmail.com

Abstract: Signcryption is a new cryptographic primitive that simultaneously fulfills both the functions of signature and encryption. The definition of generalized signcryption is proposed in the paper firstly. Generalized signcryption has a special feature that provides confidentiality or authenticity separately under the condition of specific inputs. So it is more useful than common ones. Based on ECDSA, a signcryption scheme called ECGSC is designed. It will be equivalent to an AtE(OTP$_\$$,MAC) encryption scheme or ECDSA when one of party is absent. A third party can verify the signcryption text publicly in the method of ECDSA. Security properties are proven based on Random Oracle mode: confidentiality (CUF-CPA), unforgeability (UF-CMA) and non-repudiation. Compared with the others, ECGSC presents a 78% reduction in computational cost for typical security parameters for high level security applications.

Keywords Signcryption; Generalized Signcryption; Digital Signature; Authenticated Encryption; ECDSA

1 Introduction

One of the essential problems for any computer systems is the confidential and authenticated message delivery and storage. Traditionally, the composition of authentication and encryption are used to avoiding the forgery and ensuring confidentiality of the contents of a letter [1]. The method is used in some famous security protocols, such as SSL, IPSec. In public key setting, the method can be achieved by sign-then-encrypt also. Unfortunately, the method is not practical for two reasons. Firstly, it has low efficiency and the cost is the sum of the authentication and encryption. Secondly, arbitrary schemes cannot guarantee the security. WEP(Wired Equivalent Privacy) is well known for its bad design.

Zheng proposed a novel cryptographic primitive *signcryption* which achieved the combined functionality in a single step and kept higher efficiency [2]. It is an active area in past years. Most of recent researches focus on two objects: (1) Trying to design practical signcryption schemes based on common cryptosystems [2, 3, 4, 5, 6, 7, 8, 9]; (2) Trying to prove the security of abstract structures that can be used in signcryption designing [1, 10, 11, 12, 13, 14, 15, 16]. Though many results were published, there is no scheme based on standard elliptic curve scheme until our work.

Not all messages require both confidentiality and integrity. Though traditional signcryption is efficient to provide both signature and encryption functions at the same time, it will not be feasible when only one function is requested. Zheng suggest that signcryption is replaced with other signature/ encryption algorithms to resolve the problem. So, applications must contain at least three cryptographic primitives, which will be infeasible in some space restricted applications such as embedded systems and ubiquitous computing.

This paper is motivated by above results. Our contribution lies in two aspects: (1) *Generalized Signcryption* which will provide three functions by using a universal primitive was defined. (2) A generalized signcryption scheme, ECGSC, which based on standard signature scheme ECDSA is proposed in section 3. Using the hypothesis that ECDSA is secure against UF-CMA, the formal proof based on Random Oracle mode[17]for ECGSC is shown. Compared with other schemes, efficiency of ECGSC is shown also.

2 ECDSA

A signcryption scheme usually comes from a signature scheme. Our ECGSC is based on the standard elliptic curve signature scheme (ECDSA). ECDSA (Elliptic Curve Digital Signature Algorithm), proposed by Vanstone [19], is the only signature standard based on elliptic curve and the elliptic curve version of DSS (Digital Signature Standard). Brown has given a precise security proof [20]. It is secure against all of known attacks except for duplicate signature [21]. ECDSA is one of the most famous schemes because of its security and efficiency. It has been accepted as signature standard by most of the standard organizations: ISO 15946-2, ANSI X9.62, FIPS 186.2, IEEE1363-2000, SECG and 3278. ECDSA will take the place of DSA and RSA in the future.

3 Signcryption and Generalized Signcryption

3.1 Definition of Signcryption

Signcryption is a two-party protocol as well as other public key schemes. Message *m* will be signcrypted by asender *S* and sent to a specific recipient *R* who will designcrypt and verify. The trusted party has the right to settle Alice's repudiation without recipient's private key. Namely, a signcrypted text must be verified publicly. The definition of signcryption is given as follows.

Definition 1 (Signcryption). A signcryption scheme Σ=(Gen, SC, DSC) consists of three algorithms: Keys generation algorithm Gen generates a pair of keys for user U. (SDK_U, VEK_U)$\leftarrow$Gen(U, T), where T is a security parameter, SDK is a secret key, VEK is a public key. Signcryption algorithm SC is a probabilistic algorithm. For any $m\in M$, $w\leftarrow$SC(m, SDK_S,VEK_R), w is a signcryption text. Decryption algorithm DSC is a deterministic algorithm. For any signcryption text w, $m\cup\{\perp\}$ $\leftarrow$DSC(w, SDK_R, VEK_S), where $\perp$ denotes invalid.

Definition 2 (Correctness). A signcryption scheme Σ=(Gen, SC, DSC) is correct only if**:** " S, R and $m\in M$, \$ DSC(SC(m,SDK_S,VEK_R),SDK_R,VEK_S)=m.

The security notions of signcryption were given by Zheng. A signcryption scheme is secure, if the following conditions are satisfied [2]:

Unforgeability: it is computationally infeasible for an adaptive attacker, who may be a dishonest Bob and allowed to query Alice's signcryption algorithm, to masquerade Alice in creating an authentic signcrypted text.

Non-repudiation: it is computationally feasible for a third party to settle a dispute between Alice and Bob in an event where Alice denies the fact that she is the originator of a signcrypted text with Bob as its recipient.

Confidentiality: it is computationally infeasible for an adaptive attacker to gain any partial information on the contents of a signcrypted text. The adaptive attacker may be any party other than Alice and Bob.

The first formal mode and proof for signcryption were given by Baek, Steinfeld and Zheng[18].

3.2 Generalized signcryption

Not all messages require both confidentiality and integrity. Some messages may need to be signed only, while others may need to be encrypted only. The later two cases can only provide one of specific parties, while traditional signcryption requires both of them. Traditional signcrytion (in Definition 1) will halt because one party which is specific and has key pairs is absent. Zheng suggest that applications may switch to SCS signature and ElGamal encryption to solve this problem. Namely, applications must implement at least three primitives: signature, encryption and signcryption. But the approach is impossible in some space restricted applications such as embedded systems and ubiquitous computing. To fix the above problem, we will propose the definition of generalized signcryption.

Generalized signcryption is a signcryption with more flexibility and practicability. It provides double functions when confidentiality and authenticity are required simultaneously, and provides single encryption/signature function when confidentiality/ authenticity is required only without any amended and additional computation. Namely, a generalized signcryption scheme will be equivalent to a signature scheme or an encryption scheme in special cases.

So, it exists three cases: signcryption, signature-only and encryption-only. How to identify the three cases is an important problem. In public key settings, performing the authentication operation requires the information about specific sender (public key and private key). Performing the encryption operation requires the information about specific recipient (public key and private key). Performing the signcryption operation requires the information about both parties. Thus, identity of operator can be used to distinguish the three cases. It is signcryption operation when both specific parties exist. It is signature/encryption when one of specific parties exists.

Definition 3 (Generalized Signcryption). A Generalized signcryption scheme Σ=(Gen, SC, DSC) consists of three algorithms: Gen is a keys generate algorithm as above. SC is a probabilistic signcryption algorithm. For any $m\in M,w\leftarrow$SC(m, SDK$_S$,VEK$_R$). When $R\in F$, SC(m, SDK$_S$, VEK$_R$)=Sig(m, SDK$_S$), DSC(w, SDK$_R$, VEK$_S$) =Ver(t, VEK$_S$). DSC is a deterministic designcryption algorithm. For any signcryption text w, $m\cup\{\perp\}\leftarrow$DSC(w, SDK$_R$, VEK$_S$). When $S\in F$, SC (m, SDK$_S$, VEK$_R$) =Enc(m, VEK$_R$), DSC(w, SDK$_R$, VEK$_S$)=Dec(e, SDK$_R$). Where, ENC= (Gen,Enc,Dec) is an encryption scheme: $e\leftarrow$Enc(m, VEK$_R$), $m\leftarrow$Dec (e, SDK$_R$). SIG=(Gen,Sig,Ver) is a signature scheme: $t\leftarrow$ Sig(m, SDK$_S$), $\{$T$,\perp\}\leftarrow$Ver(t, VEK$_S$). T denotes a valid signature. $\perp$ denotes an invalid signature.

4. ECGSC: Elliptic Curve based Generalized Signcryption

4.1 ECGSC

A generalized signcryption scheme ECGSC will be proposed in this section. It is the first signcryption scheme based on the standard signature scheme ECDSA.

Parameters

According to SEC1, an elliptic curve E(Fp)over finite field Fp is a sextuple: $T=(p, a, b, G, n, h)$, where G is base point, prime n is the order of G, O is the point at infinity, $[n]G=O$. Notice that, for any point $P\in E$(Fp), $P+O=P$.

$Q=[x]G$ denotes multiple double additions on elliptic curve. $\in_R$ denotes choose an element randomly. *Bind* is some information about Alice and Bob. $\{0, 1\}^l$ denotes a l bits binary string. *Kenc*, *Kmac* and *Ksig* are binary strings.

Four hash functions are used in ECGSC. H:$\{0,1\}^*\rightarrow Z_p^*$. K: $Z_p^*\rightarrow\{0,1\}^{Z+*}$. LH(.):$\{0,1\}^*\rightarrow\{0,1\}^{l+z}$. A hash functions with long outputs. Eg. SHA-256, SHA-384 and SHA-512. MAC_k:$\{0,1\}^l\times\{0,1\}^t\rightarrow\{0,1\}^z$, is a keyed message authentication function with k, where $|k|=t$. $|m|=l,l+|MAC(.)|=|LH(x_2)|$, $|.|$ indicates the number of bits in the binary representation of an integer. $H(0)\rightarrow0,K(0)\rightarrow0,LH(0)\rightarrow0,MAC_0\rightarrow0$.

Presentation of ECGSC

Message $m\in\{0,1\}^l$ will be signcrypted by a sender Alice and designcrypted by a recipient Bob.

ECGSC (Gen, SC, DSC)

Keys generation

```
Gen(Alice, $T$)
  $d_A \in_R \{1,\ldots,n\text{-}1\}; Q_A=[d_A]G$;
  Return $(d_A, Q_A)$.
$(0, O) \leftarrow$ Gen($U$, $T$), $U \in F$.
```

```
Gen(Bob, $T$)
$d_B \in_R \{1,\ldots,n\text{-}1\}; Q_B=[d_B]G$;
Return $(d_B, Q_A)$.
```

Signcryption

```
SC($m$, $d_A$, $Q_B$)
  1. $k \in_R \{1,\ldots,n\text{-}1\}$;
  2. $R \leftarrow [k]G=(x_1, y_1); r \leftarrow x_1 \bmod p$;
  3. $[k]P_B=(x_2, y_2)$;
  4. $Kenc \leftarrow LH(x_2); (Kmac, Ksig) \leftarrow K(y_2)$;
  5. If $d_A=0$, $s \leftarrow j$;
  else $s \leftarrow k^{-1}(H(m||Bind||Ksig)+rd_A) \bmod n$;
  6. $e \leftarrow MAC_{Kmac}(m||s)$;
  7. $c \leftarrow (m||e) \oplus Kenc$;
  Return $w=(c, R, s)$.
```

Designcryption

```
DSC($w$, $d_B$, $Q_A$)
  1. $r \leftarrow R$;
  2. $(x_2,y_2)=[d_B]R$;
  3. $Kenc \leftarrow LH(x_2); (Kmac, Ksig) \leftarrow K(y_2)$;
  4. $(m||e) \leftarrow c \oplus Kenc$;
  5. $e' \leftarrow MAC_{Kmac}(m||s)$;
  If $e \neq e'$, return $\perp$; else if $s=j$, return $m$;
  6. $u_1 \leftarrow s^{-1}H(m||Bind||Ksig); u_2 \leftarrow s^{-1}r$;
  7. $R' \leftarrow [u_1]G+[u_2]Q_A$;
  If $R' \neq R$, return $\perp$;else return $m$.
```

Verified Publicly.

The trusted party will settle Alice's repudiation. Namely, Alice has no chance to deny a signcrypted text sent by her. The trusted party performs the following algorithm after $w'=(H(m||Bind||Ksig), R, s)$ has been published by Bob.

```
VP($w'$, $Q_A$)
  1.$u_1 \leftarrow s^{-1}H(m||Bind||Ksig); u_2 \leftarrow s^{-1}r$;
  2.$R' \leftarrow [u_1]G+[u_2]Q_A$; if $R' \neq R$, return $\perp$;else return T.
```

ECGSC is the unique standard verifiable scheme. In ECGSC, doublet (R, s) is an ECDSA signature on message $H(m||Bind||Ksig)$ which can be verified in a standard mode. $H(m||Bind||Ksig)$ will not release any information about message because the padding key $Ksig$ is a random number. Bao&Deng's scheme[3] is the first publicly verifiable scheme which has two shortcomings: (1)based on a non-standard signature scheme; (2)published $H(m)$ which will release the partial information. Though SC-DSA[5] is based on DSA, its verification operation is not a standard one.

Signature mode and encryption mode.

Let the recipient $R=j$, ECGSC will become ECDSA scheme. $(m, R, s) \leftarrow$SC(m, d_A, O), $\{T, \perp\} \leftarrow$DSC(w, 0, Q_A).

Let the sender $S=j$, ECGSC will become an encryption scheme.(c, R) $\leftarrow$SC(m, 0, Q_B), $m \cup \{\perp\} \leftarrow$DSC($w$, d_B , O).

4.2 Correctness

(1) In the case of S, $R \notin F$. We denote Alice by S, and Bob by R. ECGSC is correct only if the equation 1 is satisfied.

$$\mathrm{DSC}(\mathrm{SC}(m,d_A,Q_B),\, d_B,Q_A)=m \tag{1}$$

Proof. The left of e.q.1 is a designcryption algorithm:

$\mathrm{DSC}(\mathrm{SC}(m,\, d_A,Q_B),\, d_B,Q_A)=\mathrm{DSC}((c,\, R,\, s),\, d_B,Q_A)$.

We denote the parameters in designcryption operation by (x'_2,y'_2), $Kenc'$, $Kmac'$ and $Ksig'$.

$[d_B]R \rightarrow [d_Bk]G \rightarrow [k]Q_B=(x'_2,y'_2) \Rightarrow x'_2=x_2, y'_2=y_2 \Rightarrow Kenc'=Kenc, Kmac'=Kmac, Ksig'=Ksig \Rightarrow e'=e,\, m'=m$

Thus, step1-4 is correct. $s^{-1}= k\ (H(m\|Bind\|Ksig)+rd_A)^{-1}$

Let $h=H(m\|Bind\|Ksig)$

$u_1=k(h+rd_A)^{-1}h, u_2= k(h+rd_A)^{-1}r \Rightarrow R'=[u_1]G+ [u_2]Q_A=[\, k]G \Rightarrow R= R'$

Thus, step5-6 is correct.

So, $\mathrm{DSC}(\mathrm{SC}(m,\, d_A,Q_B),d_B,Q_A)=m$ is satisfied.

(2) Work with signature-only mode. $R \in F, d_A=0, Q_A=O$. ECGSC will became ECDSA.

Alice signs a message as follows:

```
SC($m$, $d_A$, $O$)
  1. $k \in_R \{1,\ldots,n\text{-}1\}$;
  2. $R=[k]G=(x_1, y_1)$; $r = x_1 \bmod p$;
  3. $[k]O = O$;$0 \leftarrow LH(0)$;$(0,\ 0) \leftarrow K\ (0)$;
  4. $s \leftarrow [k^{-1}](H(m||0||0)+rd_A) \bmod n$;
  5. $0=MAC_0\,(m)$;
  6. $m \leftarrow (m||0) \oplus 0$;
  Return $w=(m,\, R,\, s)$.
```

Any recipient can verify as follows:

```
DSC($w$, 0, $Q_A$)
1. $O \leftarrow [0]R$;$0 \leftarrow LH(0)$;$(0,\ 0) \leftarrow K\ (0)$;
2. $m \leftarrow m \oplus 0$;
3. $0 \leftarrow MAC_0\,(m)$;
4. $u_1 \leftarrow s^{-1}H(m||0||0)$;$u_2 \leftarrow s^{-1}r$;
5. $R'= [u_1]G+[u_2]Q_A$;
6. If $R' \neq R$, return $\perp$; else return T.
```

The above algorithm is ECDSA if we bypass all of the null operations. The correctness of ECDSA is known.

(3) Work with encryption-only mode. $R \in F, d_B=0, Q_B =O$. ECGSC will become a encryption scheme. Any one who knows Bob's public key can encrypt a message.

```
SC($m$, 0, $Q_B$)
  1. $k \in_R \{1,\ldots,n\text{-}1\}$;
  2. $R \leftarrow [k]G=(x_1, y_1)$;$r \leftarrow x_1 \bmod p$;
  3. $[k]P_B=(x_2, y_2)$;
  $Kenc \leftarrow LH(x_2)$;$(Kmac,Ksig) \leftarrow K\ y_2)$;
  4. $e \leftarrow MAC_{Kmac}\,(m)$;
  5. $c \leftarrow (m||e) \oplus Kenc$;
  Return $w=(c, R)$.
```

```
DSC($w$, $d_B$, $O$)
  1. $(x_2,y_2)=[d_B]R$;
  $Kenc \leftarrow LH(x_2)$;$(Kmac,\ Ksig) \leftarrow K\ (y_2)$;
  2. $(m||e) \leftarrow c \oplus Kenc$;
  3. $e' \leftarrow MAC_{Kmac}(m)$;
  If $e' \neq e$, return $\perp$; else return $m$.
```

The algorithm is correct if $\mathrm{DSC}(\mathrm{SC}(c,\, R),\, d_B,\, O)=m$ is satisfied.

The correctness of it can be found in the proof of step1-4 in the first case.

4.3 Security of ECGSC

Except for security notions mentioned in section 3.1, it can defined insider security and outsider security according to whether Bob is an attacker [12].Obviously, insider security is stronger. We will give reduction proofs based on known results by using above notions.

Definition 4 (*ECDLP*). (Elliptic curve discrete logarithm problem). Compute $x \leftarrow \mathrm{ECDLP}\ (G, Y)$, where $Y=[x]\ G$, G is a base point, $x \in [1,\ldots, n\text{-}1]$ and $Y \in \langle G \rangle$.

Hypothesis 1 (*ECDLP is hard*). Given secure parameter T, $x=\mathrm{Adv}_{\mathrm{ECDLP}}(T,t¢¢)$ denotes the probability of resolving ECDLP in time $t¢¢$. x is a negligible quantity at present.

Hypothesis 2 (*Random Oracle*). Hash function has the property of Random Oracle. Namely, the outputs of hash function are random and uniform.

4.3.1 Unforgeability. In the sense of insider security, the forgers are Bob and others. Dishonest Bob is the most powerful attacker to forge a signcryption, because he is the only person who knows the private key d_B which is required to directly verify a signcryption from Alice. Then the problem will turn into the verification of the normal ECDSA signature. Brown has proved the security of ECDSA [21]: if hash function is idealized as a random oracle, then ECDSA has active existential unforgeability.

Using the hypothesis that ECDSA is secure against UF-CMA, unforgeability of ECGSC will be proved based on Random Oracle mode[17].

Theorem 1. Assume that there exists an adversary *ASC* that wins the UF-CMA game against ECGSC in time t, using q_{sc} queries to its signing oracle and (q_h,q_m) queries to its random oracles. Then there exists an algorithm *AS* that wins the UF-CMA game against the ECDSA signature scheme such that

$$\begin{aligned} &\mathrm{Adv}^{\text{UF-CMA}}_{\text{ASC}}(T,t,q_{sc},q_{h,}q_m) \\ \le\ &A\mathrm{dv}^{\text{UF-CMA}}_{\text{AS}}(T,t',q_{sc},q_h+q_{sc}) + {2q_h + q_{sc}(q_{sc}-1)} \big/ {2n} \end{aligned} \qquad (2)$$

The algorithm *AS* asks q_{sc} queries to its signing oracle and q_{sc} + q_h queries to its random oracle.It runs in time t'.

Proof. The following listing specifies the initial UF-CMA game against ECGSC in its entirety. Note the usage of random oracles in place of the cryptographic hash functions *H* and *MAC*. The oracles are simulated by lazy evaluation using lists L_H and L_{MAC} to maintain state between queries. A signing oracle Oracle_*SC* provides signcryption service for input message.

Game 0:

```
    $(d_A, Q_A) \leftarrow$ Gen($A$, $T$);
    $(d_B, Q_B) \leftarrow$ Gen($B$, $T$);
    $Bind \leftarrow A \| B$;
    $(m^*, w^*) \leftarrow ASC(T, Q_A, Q_B,$ Oracle_$SC$, Oracle_$H$, Oracle_$MAC$ );
    $ASC$ wins, If $m^* \leftarrow$ DSC($w^*, Q_A, d_B$) and $m^*$ was never a query to Oracle_$SC$.

  Oracle_$SC(m)$
    Return SC($m$, $d_A$, $Q_B$).
  Oracle_$H(m \| Bind \| Ksig)$
    If $(m \| Bind \| Ksig, h)$ in $L_H$, return $h$; else $h \in_R \{0,1\}^{|p|}$, Add $(m \| Bind \| Ksig, h)$ to $L_H$;
    Return $h$.
  Oracle_$MAC(m)$
    If $(m, e)$ in $L_{MAC}$, return $e$; else $e \in_R \{0,1\}^z$, add $(m, e)$ to $L_{MAC}$;
    Return $e$.
```

Next, consider the following algorithm *SC*, which plays the UF-CMA game against ECDSA and uses *ASC* as a subroutine.

Game 1:

```
  $AS(T, Q_A,$ Oracle_$Sign$, Oralce_$H$)
    $(d_B, Q_B) \leftarrow$ Gen($R$, $T$);
    $Bind \leftarrow A \| B$;
    $(m^*, w^*) \leftarrow ASC(T, Q_A, Q_B,,$ Sim_$SC$, Sim_$H$, Sim_$MAC$ );
    If $m^* \leftarrow$ DSC($Q_S$, $d_B$, $w^*$) and $m^*$ was never a query to Oracle_$Sign$
    $(c^*, R^*, s^*) \leftarrow w^*$; return ( $m^*$, $R^*$, $s^*$);
    Else return $\perp$.
```

Signcryption will be forged by the assistance of signing oracle Oracle_*Sign* which provides signing service. A random oracle Oracle_*H* with list L_H will in place of hash function *H*.

```
  Sim_$SC(m)$
    $(r, s) \leftarrow$ Oracle_$Sign(m)$;
    $R \leftarrow r$;
    $(x_2, y_2) = [d_B]R$;
    $Kenc \leftarrow LH(x_2)$; $(Kmac, Ksig) \leftarrow K(y_2)$;
    $e \leftarrow$ Sim_$MAC_{Kmac}(m)$ ;
    $c \leftarrow (m \| e) \oplus Kenc$;
    $h \leftarrow$ Oracle_$H(m)$, add $(m, h)$ to $L_H$;
    $h' \leftarrow$ Oracle_$H(m \| Bind \| Ksig)$;
    Add $(m \| Bind \| Ksig, h')$ to $L_H$;
    $k \leftarrow$ ECDLP($T$, $R$) ;
    $s' \leftarrow k^{-1}(h' + k\,s - h)$ mod $n$;
    $w \leftarrow (c, R, s')$;
    Return $w$.
  Sim_$H(m \| Bind \| Ksig)$
    $m \leftarrow m \| Bind \| Ksig$;
    $h \leftarrow$ Oracle_$H(m)$, add $(m, h)$ to $L_H$;
```

h'←Oracle_$H(m\|Bind\|Ksig)$;

Add ($m\|Bind\|Ksig$, h') to L_H;

Return *h*¢.

Sim_*MAC*(*m*)

If (*m*, *e*) in the list L_{MAC}, return *e*; else $e\in_R\{0,1\}^z$,add (*m*, *e*) to L_{MAC}, return *e*.

The only event that may cause Game 0 and Game 1 to differ, is that Sim_*SC* returns a value *h* whose preimage has already been assigned a value in L_H. To bound the probability of this occurring, consider the worst scenario that may occur: *ASC* asks q_h queries before querying Sim_*SC* with the same *m*, q_{sc} times. The total probability that no errors have occurred can be denoted by e.q.-3.

$$\text{Prob(h)} \;=\; 2q_h + q_{sc}(q_{sc}-1) \Big/ 2n \tag{3}$$

Where, notation h denotes the collision, ¬h denotes no collision.

ASC wins denotes the event that *ASC* wins. AS wins denotes the event that *AS* wins. ECDLP wins denotes the event that ECDLP is solved. Thus, ECDLP wins imply AS wins. The following equation is satisfied when h is not happen:

$$\text{Prob(ASCwins}\,|\,\neg\text{h}) = \text{Prob(ECDLP wins} \vee (\text{ASwins}\,|\,\neg\text{h})) = \text{Prob(ASwins}\,|\,\neg\text{h})$$

So, the probability of AS wins:

$$\begin{aligned}
&\text{Prob(ASC wins)} = \text{Prob(ASC wins} \wedge \neg\,\text{h}) + \text{Prob(ASC wins} \wedge \text{h})\\
&\le \text{Prob(ASC wins} \mid \neg\,\text{h})\text{Prob}(\neg\,\text{h}) + \text{Prob(h)}\\
&= \text{Prob(AS wins} \mid \neg\,\text{h})\text{Prob}(\neg\,\text{h}) + \text{Prob(h)}\\
&= \text{Prob(AS wins} \wedge \neg\,\text{h}) + \text{Prob(h)}\\
&= A\,\text{dv}_{\text{AS}}^{\text{UF -CMA}}\,(T, t', q_{sc}, q_h + q_{sc}) + 2q_h + q_{sc}(q_{sc}-1) \Big/ 2n
\end{aligned}$$

In the same time, *ASC* wins the UF-CMA game against ECGSC in time *t*, using q_{sc} queries to its signing oracle and (q_h,q_m) queries to its random oracles. The probability of *ASC* wins as follows:

$$\text{Adv}_{\text{ASC}}^{\text{UF-CMA}}\,(T, t, q_{sc}, q_{h,} q_m) = \text{Prob(ASC wins)}$$

ECGSC is secure against UF-CMA, if ECDSA is secure against UF-CMA.

4.3.2 Non-repudiation. As well as signature schemes, unforgeability implies non-repudiation if there is no duplication of the signcryption. If the signcryption scheme is malleable or forgeable, Alice will have opportunity to deny. Non-repudiation of ECGSC can be achieved only if no repudiation signcryption exits because of its unforgeability.

Stern, Pointcheval and Malone-Lee found that ECDSA is a *duplicate signature*, because the map *f*: *R*→*r* is not unique [21]. The two symmetrical point has the same *x*-coordinate: *R* =(x_R, y_R), -*R* =($x_{R,}$, -y_R), so the same signature (*r*, *s*) can be got by (m_1,*R*,*s*) and (m_2,-*R*, *s*). The flaw is fixed in ECGSC by using *f*: *R*→*R* instead of *f*: *R*→*r*. Thus, ECGSC is not a duplicate signcryption because the map *f*: *R*→*R* is unique.

Non-repudiation of ECGSC is achieved through verification of the triplet ($H(m\|Bind_{A,B}\|Ksig)$, *R*, *s*). Thus, ECGSC is non-repudiation.

4.3.3 Confidentiality. A new provable security encryption scheme will be constructed which have the same confidentially as ECGSC. Krawczyk has proved the AtE mode is CUF-CPA (chosen plaintext attacks) with the CBC (Cipher Block Chaining with a secure underlying block cipher) or OTP (One Time Padding, stream ciphers that xor data with a (pseudo) random pad)[1].

Definition 5（CUF-CPA). An encryption is ciphertext unforgeable, and denote it CUF-CPA, if is infeasible for any attacker F that has access to an encryption oracle Oracle_E with key k to produce a valid ciphertext under k not generate by Oracle_E as response to one of the query by F. Namely, we quantify cipher unforgeability by function $E(q, Q, t)$ defined as the maximal probability of success for any cipher forger F that queries q plaintexts totaling Q bits and spend time t in the attack. $E(q, Q, t)$ is negligible.

Definition 6 (OTP(*F*)). The OTP encryption under $f\in F$ of plaintext x is performed by choosing $r\in_{\mathrm{R}}\{0,1\}^{l}$ and computing $c=f(r)\oplus x$, where $F=\{f \mid f{:}\{0,1\}^{l}\rightarrow\{0,1\}^{l'}\}, x\in M$. The ciphertext is the pair(r, c). If f is chosen at random and there are no repetitions in the value r, OTP schemes will be noted as $\mathrm{OTP}_{\$}$.

Definition 7. AtE($\mathrm{OTP}_{\$}$, MAC) composition: (i) computes $t=MAC_k(x)$; (ii) appends t to x; (iii) output $c=f(r)\oplus(x\|t)$. Where, $MAC_k{:}\{0,1\}^{*}\times\{0,1\}^{t}\rightarrow\{0, 1\}^{n}$, $|k|=t$.

Lemma 1. AtE(OTP$,MAC) is secure against CUF-CPA, if message authentication function MAC is secure against IND-CMA (Indistinguishability – chosen message attacks). Proof can be found in [1].

We will construct an encryption scheme ENC in AtE($\mathrm{OTP}_{\$}$,$MAC$) manner which works on inputs message m. $LH(.)$ is a hash function with $l'+|n|$ bits outputs. Hash function H with l bits outputs.

```
Alice Encrypt as follows:
  1. $k\in_{\mathrm{R}}\{1,\ldots,n-1\}$;
  2. $(x_1, y_1)=R\leftarrow[k]G$;
  3. $(x_2, y_2)=[k]Q_B$;
  4. $Kenc\leftarrow LH(x_2),(Kmac, Ksig)\leftarrow K(y_2)$;
  5. $e\leftarrow H(m\|Kmac)$;
  6. $c\leftarrow(m\|e)\oplus Kenc$;
Return $(c, R)$.
```

```
Bob Decrypt as follows:
  1. $[d_B]R=(x_2, y_2)$;
  2. $Kenc\leftarrow LH(x_2),(Kmac, Ksig)\leftarrow K(y_2)$;
  3. $(m\|e)\leftarrow c\oplus Kenc$;
  4. $e'\leftarrow H(m\|Kmac)$
  If $e\neq e'$, return $\perp$; else, return $m$.
```

Lemma 2. ENC is secure against CUF-CPA.

Proof. Defining two functions: (i) $x(R)=R_x$ denotes the operation of computing x-coordinate of a point R; (ii) $E(x)=R$ denotes the operation of embedding x into an elliptic curve as a point R.

Let $r=x(R)=x_1$ and $R=[k]G$. The value of r is random because of the same property of k. Let $f(.)=LH(x([d_B]E(.)))$. Function $f(.)$ is private and selected randomly, because d_B is private and selected randomly.

While $f(r)=LH(x([d_B]E(r)))=LH(x([d_B]E(x_1)))=LH(x([d_B]R))=LH(x_2)=Kenc$

Kmac is the authentic key that can be computed by both the sender and recipient. Hence, *ENC* is a composition in AtE($OTP_{\$}$,*MAC*) manner. $H(.)$ is a secure hash function which achieves the IND-CMA security.

Then by Lemma 1, ENC is CUF-CPA and implements secure channels.

Theorem2. ENC and ECGSC have the same security property of confidentiality.

Proof. All of public data for ECGSC as follows: (T, Q_A, Q_B, R, c, s). The attacker can compute $[H(m||Bind||Ksig)]G=[r]Q_A-[s]R$, where $r=x_1=x(R)$. Let $h=H(m||Bind_{A,B}||Ksig)$.

An adaptive attacker to ENC can obtain the following public data: (T, Q_B, R, c). Giving the value of $[h]G$ will not reduce the complexity of attacking ENC, because $[h]G$ hides all of the information of message m under the assumption of *Random Oracle*.

Suppose that *AENC* is an adversary for ENC which works on input $(T, Q_B, R, c, [h]G)$ and outputs partial information $\tilde{m}$. *ASC* is an adversary for ECGSC which works on input (T, Q_A, Q_B, R, c, s) and outputs partial information $\tilde{m}$. We can reduce the two algorithm form each other.

There is a deterministic polynomial time algorithm *ASC* (T, Q_A, Q_B, R, c, s):

1. Computes$[h]G=[r]Q_A-[s]R$;
2. $\tilde{m} \leftarrow AENC(T, Q_B, R, c, [h]G)$;
3. Return $\tilde{m}$.

If *AENC* gets any partial information of message m, so does *ASC*.

There is a deterministic polynomial time algorithm *AENC* $(T, Q_B, R, c, [h]G)$

1. $s\in_R\{1,\ldots,n\}$;
2. computes $Q_A=[r^{-1}s]R-[r^{-1}h]G$;
3. $\tilde{m} \leftarrow ASC(T, Q_A, Q_B, R, c, s)$;
4. Return $\tilde{m}$.

If *ASC* gets any partial information of message m, so does *AENC*.

ECGSC has the same confidentiality as ENC.

By Lemma 2, ECGSC is CUF-CPA and implements secure channel.

4.4 Efficiency of ECGSC

In this section, ECGSC will be compared with other typical schemes which include SCS[2], Bao&Deng[3], KCDSA[4], SC-DSA[5], TBOS[6] and ECSCS[9].

4.4.1 Computational Cost. In public key cryptosystems, computing modular multiplication, modular exponential, modular inverse and multiples of points on elliptic curve consume the most of computational resources, while the cost of addition, hash, encrypt\decrypt (symmetric cryptosystems) are negligible. So we will examine previous ones only.

Table 1. Comparison of computation cost.

Schemes	KG	S	D	AC	VP
SCS	2*E*	1*E*+1*I*	2*E*	/	/
ECSCS	2*k*P	1*k*p+1*I*	2*k*P	/	/
B&D	2*E*	2*E*+1*I*	3*E*	0	2*E*
KCDSA	2*E*	2*E*	3*E*	save *r,s*or 3*E*	2*E*
SC-DSA	2*E*	2*E*+2*I*	3*E*+1*I*	save *r,s* or 2*E*+1I	2*E*+1*I*
TBOS	2*E*+2*I*	2*E*	2*E*	0	*E*
ECGSC	2*k*P	2*k*P+1*I*	3*k*P+1*I*	0	2*k*P+1*I*

Notes in notations: a. KG denotes the keys generation algorithm; S denotes the signcrytpion algorithm; D denotes the designcryption algorithm; AC denotes the additional computation have to accomplish for public verify; VP denotes the public verify by the third one. b. *E* denotes the modular exponential computation; *I* denotes modular inverse computation; *k*P denotes scalar multiplication computation of points on elliptic curve. / denotes no such function.

Remark 1. (Compared with DLP based signcryption schemes). SCS is the fastest scheme in all of the four DLP based schemes (SCS, B&D, KCDSA and SC-DSA). The operation of multiple double additions on elliptic curve can be expected to be about 8 times faster than the operation of modular exponential[22]. By the results, the computation cost of keys generation operation in ECGSC is 1/8 of that in SCS; signcryption operation in ECGSC is 1/4 of that in SCS, and designcrption is 1/5 of that in SCS. ECGSC saves computational costs 78% over SCS in all.

Remark 2. (Compared with RSA based signcryption scheme). TBOS is the only scheme based on RSA. By the result of [22], the computation cost of keys generation operation and signcryption operation in ECGSC are 1/8 of that in TBOS approximately; and designcrption is 1/5 of that in TBOS. ECGSC saves computational costs 82% in all.

*Remark 3***.** (Compared with other ECDLP based schemes). ECSCS is the only known scheme based on ECDLP except for our ECGSC. The computation cost of ECGSC is slightly higher than that of ECSCS which has the flaw of verify publicly. The cost of signcrytion operation in ECGSC is 2 times of ECSCS. The cost of designcrption operation in ECGSC is 1.5 times of ECSCS.

To sum up, ECGSC has the highest speed in all of the verifiable schemes.

4.4.2 Communication Cost

Definition 8 (Data rate). In a scheme *S* on plaintext *m*, Data Rate will defined as DR(*S*) = $|m|/|C_{\mathring{a}}|$. Where, $C_{\mathring{a}}$ denotes all of the cipher text including additional information for decryption and verification, $|m|$ denotes the length of plaintext *m*.

Table 2. Comparison of Data Rate.

Schemes	m	C_{Σ}	DR1	DR2
SCS	$\|D(.)\|$	$\|D(.)\|+\|KH(.)\|+\|q\|$	18%	26%
ECSCS	$\|D(.)\|$	$\|D(.)\|+\|h\|+\|n\|$	18%	26%
B&D	$\|D(.)\|$	$\|D(.)\|+\|h(.)\|+\|q\|$	18%	26%
KCDSA	$\|D(.)\|$	$\|D(.)\|+\|h(.)\|+\|q\|$	18%	26%
SC-DSA	$\|D(.)\|$	$\|D(.)\|+2\|q\|$	17%	25%
TBOS	$\|N\|-\|h(.)\|-\|G(.)\|$	$\|N\|$	50%	67%
ECGSC	l	$\|n\|+\|LH(.)\|+2\|p\|$	32%	35%

Notes in notations: a. For DLP based schemes (SCS, B&D, KCDSA, SC-DSA): $|a|$ denotes the size of finite field, $|q|$ denotes the order of base element. b. For RSA based schemes (TBOS): $|N|$ denotes the size of public module, $|G(.)|$ denotes the length of a hash function used in TBOS. c. For ECDLP based schemes (ECSCS, ECGSC): $|p|$ denotes the size of finite field F*p*, $|n|$ denotes the order of base point. d. $|D(.)|$ denotes the block length of the block cipher. $|h|$ denotes the secure hash function outputs length. $|LH(.)|$ denotes the length of hash function with long message digest. $|KH(.)|$ denotes the length of key hash function used in SCS, the same as $|h|$.

Data rates of mentioned schemes are shown in Table 2.

The minimum security parameters of cryptographic primitive recommended for the current practice as follows: For DLP, $|a|$=1024bits, $|q|$=160bits. For RSA, $|N|$=1024bits. For ECDLP, $|p|$=131bits (79, 109 also may be chosen), $|n|$=160bits. The block length of the block cipher is 64bits (e.g. IDEA). The length of secure hash function is 128bits (e.g. MD5). The length of long hash function is 384bits (e.g. SHA-384). DR1 are the data rates of each schemes in Table2.

The security parameters recommended for long term security as follows: For DLP, $|a|$=2048bits, $|q|$=192bits. For RSA, $|N|$=2048bits. For ECDLP, $|p|$=191bits, $|n|$=192bits. The block length of block cipher is 128bits (e.g. AES). The length of secure hash function is 160bits (e.g. SHA-1). The length of long hash function is 512bits (e.g. SHA-512). DR2 are the data rates of each schemes in Table 2.

Obviously, ECGSC has the highest communication cost in all of ELGamal type schemes except for RSA based TBOS.

5. Conclusion

The target of generalized signcryption is to fulfill multiple functions using a universal primitive. So, it must prove three functions without any additional computation and revising. There are two problems concerned in generalized signcryption designing: (1) distinguishing among three cases: signcryption, signature-only and encryption-only; (2) selecting a special operation which will output specific value under specific inputs. We use null to identify a nonsexist party because the algorithm will mask encryption/signature operation. Namely, the XOR operation will output message itself when using null key. The security must be investigated carefully when symmetric ciphers are used.

ECGSC proposed in this paper has four advantages: (1) based on standard signature ECDSA; (2) is an efficient scheme in computation and communication (storage) cost; (3) is a provable secure scheme (the same unforgeability and non-repudiation as ECDSA, CUF-CPA confidentiality); (4) is a typical generalized signcryption used broadly.

References

1 Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Kilian J. ed.. Advances in Cryptoloty-CRYPTO2001. Lecture Notes in Computer Science vol.2139. Berlin: Springer-Verlag, (2001) 310-331

2 Zheng,Y.: Digital signcryption or how to achieve cost (signature &encryption) << cost (signature) + cost (encryption). In: Kaliski. B.S. ed.. Advances in Cryptoloty-CRYPTO' 97, Lecture Notes in Computer Science vol.1294. Berlin: Springer-Verlag, (1997) 165-179

3 Bao, F.and Deng, R.H.: A signcryption scheme with signature directly verifiable by public key. In: Imai H., Zheng Y. ed.. Public Key Cryptography' 98, Lecture Notes in Computer Science vol.1431, Berlin:Springer-Verlag,(1998) 55-59

4 Yum, D.H. and Lee, P.J.: New Signcryption Schemes based on KCDSA. In: Proceedings of the 4th International Conference on Information Security and Cryptology, Seoul, South Korea, (2002) 305-317

5 Shin, J. B., Lee, K. and Shim, K.: New DSA-Verifiable Signcryption Schemes. In: Proceedings of the 5th International Conference on Information Security and Cryptology, Seoul, South Korea, (2003) 35-47

6 Malone-Lee, J. and Mao, W.: Two birds one stone: Signcryption using RSA. In: Joye M. ed.. Topics in Cryptology – Cryptographers' Track, RSA Conference 2003, Lecture Notes in Computer Science vol.2612, Berlin: Springer-Verlag, (2003) 210-224

7 Boyen, X.: Multipurpose identity-based signcryption: a swiss army knife for identity-based cryptography. In: Advances in Cryptology-Crypto' 03, Lecture Notes in Computer Science vol.2729, Berlin: Springer-Verlag, (2003) 382-398

8 Libert, B. and Quisquater, J.: Efficient signcryption with key privacy form gap Diffie-Hellman group. In: Bao Feng ed.. Public key Cryptography-PKC' 04, Lecture Notes in Computer Science vol.2947, Berlin: Springer-Verlag, (2004) 187-200

9 Zheng, Y. and Imai, H.: How to construct efficient signcryption schemes on elliptic curves. Information Processing Letters, 1998, 68(5): 227-233

10 Bellare, M. and Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Okamoto T. ed.. Advances in Cryptology-ASIACRYPT2000, Lecture Notes in Computer Science vol.1976, Berlin: Springer-Verlag, (2000) 531-545

11 Rogaway, P.: Authenticated-encryption with associated-data. In: Proceedings of Ninth ACM Conference on Computer and Communication Security(CCS2002), Washingtion DC, USA, (2002) 98-107

12 An, J.H., Dodis Y. and Rabin, T.: On the security of joint signature and encryption. In: Knudsen L. ed.. Advances in Cryptology-EUROCRYPT2002, Lecture Notes in Computer Science vol.2332. Berlin: Springer-Verlag, (2002) 83-107

13 Dodis, Y., Rreedman, M., Jarecki, S. and Walfish S.: Optimal signcryption from any trapdoor permutation. Cryptology ePrint Archive, Report: 2004/020, (2004)

14 Dodis, Y., Rreedman, M., Jarecki, S., Jarecki, S. and Walfish S.: Versatile padding schemes for joint signature and encryption. In Pfitzmann B. ed.. Proceedings of Eleventh ACM Conference on Computer and Communication Security (CCS2004) , Washingtion DC, USA, (2004) 196-205

15 Dent, A. W.: Hybrid Signcryption Schemes With Outsider Security. In: Proceedings of The 8th Information Security Conference(ISC 2005), Singapore, (2005) 203-217

16 Dent, A. W.: Hybrid Signcryption Schemes With Insider Security. In: Proceedings of Information Security and Privacy - ACISP 2005, Brisbane, Australia, (2005) 253-266

17 Bellare, M. and Rogaway, P.: Random oracle are practical: a paradigm for designing efficient protocols. In: Proceeding of the First ACM Conference on Computer and Communication Security (CCS1993), Fairfax, Virginia, USA, (1993) 62-73

18 Baek, J., Steinfeld, R. and Zheng, Y.: Formal Proofs for the Security of Signcryption. In: Naccache D. ,

Paillier P. ed.. Public Key Cryptography' 02, Lecture Notes in Computer Science vol.2274, Berlin: Springer-Verlag, (2002) 80-98

19 Johnson, D. and Menezes, A.: The elliptic curve digital signature algorithm (ECDSA). Department of C&O, University of Waterloo, Technical report: CORR 99-34, (1999)

20 Brown, D.: Generic Groups, Collision Resistance, and ECDSA. Design, Codes Cryptography, 2005, 35(1): 119-152

21 Stern, J., Pointcheval, D., Malone-Lee, J. and Smart Nigel P.: Flaws in Applying Proof Methodologies to Signature Schemes. In: Yung Moti ed.. Advances in Cryptology-Crypto' 02, Lecture Notes in Computer Science vol.2442, Berlin: Springer-Verlag, (2002), 93–110

22 Koblitz, N., Menezes, A. and Vanstone S.: The state of elliptic curve cryptography. Designs, Codes and Cryptography, 2000, 30(19): 173-193