Provable Secure Generalized Signcryption

Xu-an Wang Xiaoyuan Yang Yiliang Han

Key Laboratory of Information and Network Security
Department of Electronic Technology, Engineering College of Armied Police Force
Wujing Road, Xi'an 710086
Wangxahq@yahoo.com.cn

Abstract: Generalized Signcryption is a new cryptographic primitive which can work as an encryption scheme, a signature scheme or a signcryption scheme. We give security notions of Generalized Signcryption and improve a Generalized Signcryption scheme proposed by Han et al.We give the formal attacking model of this new cryptographic primitive in the framework of theory of provable security. At last, we give formal proofs for this new improved Generalized Signcryption in our attacking model.

Keywords: Generalized Signcryption, .signcryption, provable security

1. Introduction

Along with developments of information society, security requirements for applications are usually both confidentiality and authentication. And these requirements have given birth of new research fields in cryptography, that is, how to combine confidentiality and authentication properly. A lot of work has been done in this field, such as how to encrypt message by block cipher properly to achieve authentication or how to combine ciphertext with signature properly to achieve authentication [1] [7]. Totally we can divide the work into three types: Encryption then Sign, Sign then Encryption, Encryption and Sign. In 1997, Zheng proposed a new cryptographic primitive: Signcryption[2]. The idea is compressing two independent operations (encryption and signature) in one operation (signcryption). There are three advantages from this transformation: reducing the steps needed by encryption and signature(less computation complexity); reducing length of ciphertext produced by encryption and signature(less communication complexity); reducing two modules of encryption and signature to one module of signcryption(less implementation complexity). Since then, a lot of research results have come out. We can see SCS-DSA, SCS-KCDSA signcryption scheme based on Discrete Logarithm problem, RSA-TBOS signcryption scheme based on Integer Factoring [5], ECSCS signcryption scheme based on elliptic curve [6], identity based signcryption scheme based on pairings. In 2006, Han et al proposed a new primitive Generalized Signcryption [3]. The idea of this new primitive is still reducing, but this time, what's reducing is not the computation complexity or communication complexity, but the implementation complexity. Imagine this scenario, two users want to communicate safely. Sometimes they need both confidentiality and authentication, sometimes they just need confidentiality, and sometimes they just need authentication. If we adopt signcryption in this scenario, we must preserve module of encryption and module of signature for

solely needing confidentiality or authentication. If we do not care very much about speed, we gain no remarkable advantage for adopting signcryption. Furthermore, adding something new to an established system seems no easy. But if we can embed encryption and signature in the signcryption module, we can easily encrypt or sign or signcrypt by only one module. Generalized Signcryption is the one which fits this goal. Generalized Signcryption is a new primitive which can work as an encryption scheme, a signature scheme, or a signcryption scheme. Maybe this can broaden the application range of signcryption.We must point out here that Generalized Signcryption can not substitute of encryption or signature. But it fit some particular application perfectly.

On the one hand, Generalized Signcryption provides more function, but on the other hand it also faces more danger. In the first two sections, we try to give formal model of this new primitive and the attacking model in the framework of provable security [13] [14] [15] [16] [17]. In the last two sections, we improve the origin scheme and give proofs for this new Generalized Signcryption scheme.

2. Security Notions for Generalized Signcryption

Because Generalized Signcryption can work as encryption, signature or signcryption schemes, the adversary can get more oracles' service. For example, when considering confidentiality of Generalized Signcryption in encryption-mode, we must note adversary can get both Decryption Oracle service and Unsigncryption Oracle service. Note that Unsigncryption Oracle can maybe help the adversary decrypt challenge ciphertext. Analogously, when considering unforgeability of Generalized Signcryption in signature-mode, we must note adversary can get Signature Oracle service and Signcryption Oracle service. When considering confidentiality of Generalized Signcryption in signcryption-mode, we must note that the adversary can get Unsigncryption Oracle service and Decryption Oracle service. When considering unforgeability of Generalized Signcryption in signcryption-mode, we must note adversary can get Signature Oracle service and Signcryption Oracle service.

When talking about attacking against encryption schemes, we always emphasis on Decryption Oracle, but in fact, there is also an Encryption Oracle. But because public key is known to all, every one can get this Oracle's service, and it does not give the adversary any more attacking power than usual user. So we often omit this Oracle. The same thing happens in signature and signcryption schemes. Actually for Generalized Signcryption scheme, the adversary can get six types of Oracle's services: Encryption Oracle, Decryption Oracle, Signature Oracle, Verifying Oracle, Signcryption Oracle and Usigncryption Oracle. The reason we list just two types of services in the aboving paragraph is that these two types have closer relationship with security notions.

Definition 1(Confidentiality of Generalized Signcryption in Encryption-mode). Given security parameter $k=|p|$, let

$$Adv^{IND-CCA2}_{GSC^{ENC},A}(k) = \Pr(Exp^{IND-CCA2-1}_{GSC^{ENC},A}(k)=1) - \Pr(Exp^{IND-CCA2-0}_{GSC^{ENC},A}(k)=1)$$

For $b \in \{0,1\}$, the following is the experiment:

$Experiment\ \ Exp^{ind-cca2-b}_{GSC,A}(k)$

$cp_{sc} \leftarrow COM(k)$

randomly choose $G:\{0,1\}^* \to \{0,1\}^l$,

randomly choose $H:\{0,1\}^* \to Z/qZ$;

$PK_A, SK_A \leftarrow_R K_A(k, cp_{sc})$;

$PK_B, SK_B \leftarrow_R K_B(k, cp_{sc})$;

$(x_0, x_1, s) \leftarrow A_1^{ENC^{G,H}_{PK_B}(.),DEC^{G,H}_{SK_B}(.),SIGN^{G,H}_{SK_A}(.),VER^{G,H}_{PK_A}(.),GSC^{G,H}_{SK_a,PK_B}(.),UGSC^{G,H}_{SK_b,PK_A}(.)}(find)$;

$y = GSC^{ENC}_{PK_B}(x_b)$;

$d \leftarrow A_2^{ENC^{G,H}_{PK_B}(.),DEC^{G,H}_{SK_B}(.),SIGN^{G,H}_{SK_A}(.),VER^{G,H}_{PK_A}(.),GSC^{G,H}_{SK_a,PK_B}(.),UGSC^{G,H}_{SK_b,PK_A}(.)}(x_0, x_1, s, guess)$;

$\mathrm{Re}turn\ \ d$

In the above attacking, A can get six services, the only restriction is that y cannot be queried to the Decryption Oracle $DEC_{SK_B}(.)$. If $Adv^{IND-CCA2}_{GSC^{ENC},A}(k)$ is negligible, this Generalized Signcryption scheme is confidential when it work in encryption-mode.

Definition 2(Unforgeability of Generalized Signcryption in Signature-mode). Given security parameter $k=|p|$, the following is the experiment:

$Experiment\ \ ForgeExp^{cma}_{GSC^{SIGN},F}(k)$

$cp_{sc} \leftarrow COM(k)$

randomly choose $G:\{0,1\}^* \to \{0,1\}^l$,

randomly choose $H:\{0,1\}^* \to Z/qZ$

$PK_A, SK_A \leftarrow_R K_A(k, cp_{sc})$

$PK_B, SK_B \leftarrow_R K_B(k, cp_{sc})$

if $F^{ENC^{G,H}_{PK_B}(.),DEC^{G,H}_{SK_B}(.),SIGN^{G,H}_{SK_A}(.),VER^{G,H}_{PK_A}(.),GSC^{G,H}_{SK_a,PK_B}(.),UGSC^{G,H}_{SK_b,PK_A}(.)}(.)$ output (m,s) which satisfy

1. $VER^{G,H}_{GSC^{SIGN},PK_a}(s) = m$,

2. m has never been queried to $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$ or m is allowed to query to $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$ but s was never returned by $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$,

then return 1，else return 0

In the above attacking, A can get six services, the only restriction is m has never been queried to $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$ or m is allowed to query to $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$ but s was never returned by $SIGN^{G,H}_{GSC^{SIGN},SK_A}(.)$. Let $Succ^{CMA}_{GSC^{SIGN},F_C}(K) = \Pr[Exp^{cma}_{GSC^{SIGN},F}(k) = 1]$. If this value is negligible, this Generalized Signcryption scheme is unforgeable when it works in signature-mode.

Definition 3(Confidentially of Generalized Signcryption in Signcryption-mode for Outsider Attacker). Given security parameter $k=|p|$, the following is the experiment:

$Experiment\ \ Exp_{SC,A}(k)$

$cp_{sc} \leftarrow COM(k)$

randomly choose $G:\{0,1\}^* \to \{0,1\}^l$,

randomly choose $H:\{0,1\}^* \to Z/qZ$

$PK_A, SK_A \leftarrow_R K_A(k, cp_{sc})$

$PK_B, SK_B \leftarrow_R K_B(k, cp_{sc})$

$(x_0, x_1, s) \leftarrow A_1^{ENC^{G,H}_{PK_B}(.), DEC^{G,H}_{SK_B}(.), SIGN^{G,H}_{SK_A}(.), VER^{G,H}_{PK_A}(.), GSC^{G,H}_{SK_a, PK_B}(.), UGSC^{G,H}_{SK_b, PK_A}(.)}(find);$

$b \leftarrow_R \{0,1\},$

$c \leftarrow GSC^{G,H,SK_A,PK_B}_{SC}(m_b)$;

$b' \leftarrow A_2^{ENC^{G,H}_{PK_B}(.), DEC^{G,H}_{SK_B}(.), SIGN^{G,H}_{SK_A}(.), VER^{G,H}_{PK_A}(.), GSC^{G,H}_{SK_a, PK_B}(.), UGSC^{G,H}_{SK_b, PK_A}(.)}(x_0, x_1, s, guess)$

if $b = b'$ and c was never queried to $UGSC^{G,H,SK_A,PK_B}_{SC}(.)$

return 1, else return 0.

In the above attacking, A can get six services, the only restriction is that c was never queried $toUGSC^{G,H,SK_A,PK_B}_{SC}(.)$ .Let $Succ^{CMA}_{GSC^{SIGN},F_C}(K) = 2\Pr[Exp^{G,H}_{SC,F}(k) = 1] - 1$ .If this value is negligible, this Generalized Signcryption scheme is confidential when it works in signcryption mode.

Definition 4 (Unforgeablity of Generalized Signcryption in signcryption-mode for Outsider Attacker). Given security parameter $k=|p|$, the following is the experiment:

Experiment $ForgeExp_{GSC,F}(k)$

$cp_{sc} \leftarrow COM(k)$

randomly choose $G:\{0,1\}^* \to \{0,1\}^l,$

randomly choose $H:\{0,1\}^* \to Z/qZ$

$PK_A, SK_A \leftarrow_R K_A(k, cp_{sc})$

$PK_B, SK_B \leftarrow_R K_B(k, cp_{sc})$

if $F^{ENC^{G,H}_{PK_B}(.), DEC^{G,H}_{SK_B}(.), SIGN^{G,H}_{SK_A}(.), VER^{G,H}_{PK_A}(.), GSC^{G,H}_{SK_a, PK_B}(.), UGSC^{G,H}_{SK_b, PK_A}(.)}$(.) output (m, C) which satisfy

1. $USC^{G,H,PK_A,SK_B}_{GSC^{SC}}(C) = m,$
2. m never queried to $GSC^{G,H}_{SK_{Aa},PK_B}(.)$,

Then return 1, else return 0

In the above attacking, A can get six services, the only restriction is that c was never queried to $GSC^{G,H}_{SK_{Aa},PK_B}(.)$. Let $Succ^{CMA}_{GSC^{SIGN},F_C}(K) = \Pr[Exp^{cma}_{GSC^{SIGN},F}(k) = 1]$. If this value is negligible, the Generalized Signcryption scheme is unforgeable when it works in signcryption-mode.

Definition 5(Confidentiality of Signcryption against Insider Attacker) [11] [12]. Because insider attacker can get the sender's private key, he can signcrypt as the sender. Also the attacker can do anything as the sender does with his private key.

Definition 6(Unforgeability of Signcryption against Insider Attacker) [11] [12]. Because insider attacker can get the receiver's private key, the attacker can unsigncrypt as the receiver. Also the attacker can do anything as the receiver does with his private key.

3. A Generalized Signcryption Based on ECDSA

Han et al proposed a Generalized Signcryption based on ECDSA [4]. We point out that this scheme is not secure as its author claims and not very natural and the security analysis is not very formal. We try to solve these problems in this paper.

3.1 Description of the Origin Scheme

Parameters:

Parameters of the elliptic curve: the parameters follow the SEC1 standard, which can be described as a sixtuple $T= (p, a, b, G, n, h)$.G is a base point, ord $(G) =n$. O is the infinite element of group $<G>$

Syntax:

$Q=[x]\ G$ denotes the scalar multiplex on the elliptic curve.|| denotes connecting two messages. $\in_R$ denotes randomly choosing an element in one set. *Bind* denotes Alice and Bob's identity. $\{0,1\}^l$ denotes binary sequence of length l . $K_{enc}, K_{mac}, K_{sig}$ is a binary sequence. H：$\{0,1\}^* \to Z_p^*$ and K：$Z_p^* \to \{0,1\}^{Z+*}$ denote two hash functions. LH (.)：$\{0,1\}^* \to \{0,1\}^{l+z}$ denotes hash function output long digest, we can choose SHA-256、SHA-384 or SHA-512. MAC_k:$\{0,1\}^l \times \{0,1\}^t \to \{0,1\}^z$ denote message authenticate function which has key k. |k|=t, $|m|= l$，$l+|MAC(.)|=|LH(x_2)|$. These hash functions have property :$H(0)\to 0$，$K(0)\to 0$，$LH(0)\to 0$，$MAC_0 \to 0$.

Algorithm description:

Table 1 Han et al's original Generalized Signcryption

Key generation（n,T）
Generate Alice's private and public key: Gen(Alice, T) $d_A \in_R \{1,\ldots,n-1\}$； $Q_A=[d_A]G$； return (d_A, Q_A). Generate Bob's private and public key:Gen(Bob, T) $d_B \in_R \{1,\ldots,n-1\}$； $Q_B=[d_B]G$； return (d_B, Q_A). Generate null user's private and public key $(0, O) \leftarrow$Gen(U, T)， $U \in \Phi$。
Generalized Signcryption scheme's signcryption: SC(m, d_A, Q_B)
1. $k \in_R \{1,\ldots,n-1\}$； 2. $R \leftarrow [k]G=(x_1, y_1)$； $r \leftarrow x_1 \bmod p$； 3. $[k]P_B=(x_2, y_2)$； 4. $K_{enc} \leftarrow LH(x_2)$； $(K_{mac}, K_{sig}) \leftarrow K(y_2)$； 5. If $d_A=0$, $s \leftarrow \varphi$； Else $s \leftarrow k^{-1}(H(m\|\|Bind\|\|K_{sig})+rd_A) \bmod n$； 6. $e \leftarrow MACK_{mac}(m)$； 7. $c \leftarrow (m\|\|e) \oplus K_{enc}$； Return $\omega=(c, R, s)$.
Generalized Signcryption scheme's unsigncryption: DSC(ω, d_B, Q_A)

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

3.2 An attack on this Scheme and Some Remarks

Attack. In the above scheme the adversary intercept the ciphertext $\omega=(c, R, s)$, set $s=\varphi$, query the new ciphertext $\omega=(c, R, \varphi)$ to Decryption Oracle, the Decryption Oracle will return m, which break the confidentiality of Generalized Signcryption in signcryption-mode. Note here, the adversary does not query $\omega=(c, R, s)$ to Unsigncryption Oracle, which is the only restriction for the adversary. The attack can be successful just because we use Decryption Oracle to decrypt the modified challenge signcryption ciphertext.

Remarks on hash function. The origin scheme depend on hash function with additional property, that is, $H(0)\rightarrow 0$, $K(0)\rightarrow 0$, $LH(0)\rightarrow 0$, $MAC(0)\rightarrow 0$.But we know, if there exists non-change point in hash function, this would bring bad effects to the hash function. Especially, for hash function working in CBC mode, this can be damage. Another reason is that hash function with addition property can not be easily devised. It does not follow principal of modern hash family. So we suggest deleting this additional property.

Remarks on if-clause used in the algorithm. The original scheme uses if/else clause, and the conditional variant is s ,and s is just a local variant, programs with normal access rights can modify it. For example, some adversary can just add some program in the origin scheme's code at proper time, let $s=\varphi$, he would get the plaintext m. So we suggest delete the if-clause in the algorithm.

3.3 An Improved Generalized Signcryption Based on ECDSA

In this section, we give an improved Generalized Signcryption scheme. Improved scheme has the same parameter, syntax with the origin scheme. But we do not need hash function satisfy $H(0)\rightarrow 0$, $K(0)\rightarrow 0$, $LH(0)\rightarrow 0$, $MAC(0)\rightarrow 0$, and we introduce another point Q, which can be any point not belonging to the elliptic curve (or no one would choose this point as his public key).Here we can assume Q= (0, 0). The reason we introduce this point is for encryption-mode and signature-mode. We define a function $f(t)$. if $t=Q$, $f(t)=0$, if $t\neq Q$, then $f(t)=1$. For signcryption-mode, $Bind$= SH （$Q_A\|Q_B$）, for encryption-mode, $Bind$= SH $(Q_A\|Q)$,for signature-mode, $Bind$= SH $(Q\|Q_B)$ .SH represents hash function, its output is 32 bit, and we denote its length by |sh|. We change the length of LH's output to $l+z+|sh|$, we denote |Ksig|=|sig|

Table 2 Improved Generalized Signcryption scheme based on ECDSA

Key generation（n,T）
Alice's private and public key generation :Gen(Alice, T) $d_A \in_R \{1,\ldots n\text{-}1\}$； $Q_A=[d_A]G$； return (d_A, Q_A). Bob's private and public key generation :Gen(Bob, T) $d_B \in_R \{1,\ldots,n\text{-}1\}$； $Q_B=[d_B]G$； return (d_B, Q_A).
Generalized Signcryption Encrypt/Signcryption/Sign SC(m, d_A, Q_A ,Q_B)
1. Compute $f(Q_A)$,$f(Q_B)$ 2. $k \in_R \{1,\ldots,n\text{-}1\}$； 3. $R \leftarrow [k]G=(x_1, y_1)$； $r \leftarrow x_1 \bmod p$； 4. $[k]Q_B=(x_2, y_2)$； 5. $K_{enc} \leftarrow f(Q_B)*LH(x_2)$； ($K_{mac}$, K_{sig}) $\leftarrow f(Q_B)*K(y_2)$； 6. $s \leftarrow k^{-1}(f(Q_A)*H(m\|\|Bind\|\|K_{sig})+ f(Q_A)*rd_A) \bmod n$； 7 $e \leftarrow f(Q_B)*MACK_{mac}(m \|\|Bind \|\|s)$； 8. $c \leftarrow (m\|\|Bind \|\|e) \oplus K_{enc}$； Return $\omega=(c, R, s)$.
Generalized Signcryption Decryption/Unsigncryption/Verify DSC(ω, d_B, Q_A,Q_B)
1.Compute $f(Q_A)$,$f(Q_B)$ 2. $r \leftarrow x(R)$(R's x-coordinate)； 3. $(x_2,y_2)=[d_B]R$； . 4. $K_{enc} \leftarrow f(Q_B)*LH(x_2)$； ($K_{mac}$, K_{sig}) $\leftarrow f(Q_B)*K(y_2)$； 5. $(m\|\|Bind \|\|e) \leftarrow c \oplus K_{enc}$； 6.$e' \leftarrow f(Q_B)*MAC_{Kmac}(m \| Bind \| s)$； If $e \neq e'$, return $\perp$； 7. $u_1 \leftarrow s^{-1}*f(Q_A)*H(m\|\|Bind\|\|K_{sig})$； $u_2 \leftarrow s^{-1}*f(Q_A)*r$； 8. $R' \leftarrow [u_1]G+[u_2]Q_A$； If $R' \neq [f(Q_A)]R$, return $\perp$； else return m

3.4 Security Proofs for Improved Generalized Signcryption Based on ECDSA

The idea of the origin scheme's author about security proofs is the following. When the Generalized Signcryption work as in signcryption-mode, the author can reduce confidentiality of signcryption to a scheme proposed by Krawczyk in Crypto 2001[4], and this scheme is proved to be ciphetext unforgeable under chosen plaintext attacks. We denote this encryption scheme ATEOTP and the analog Elliptic Curve's variant ECATEOTP. But the author just discussed the Signcryption Oracle service, no caring about other Oracle service, this is not sufficient. The author can also reduce SUF-CMA of signcryption to SUF-CMA of ECDSA, but the analysis is not very formal. This paper tries to give formal analysis.

3.4.1 Prove SUF-CMA of the Generalized Signcryption in Signcryption-mode

We will apply a standard technique of provable security theory game hopping in our proofs. We define a sequence of games: $G1$, $G2$ …. they are reduced from the real attacking game $G0$. In every game, the private and public key, the adversary and the Random Oracle's coin flipping space are not changed. The difference comes from the view defined by rules. We will reduce the attack to SUF-CMA of ECGSC to

SUF-CMA of ECDSA. Assume the success probability of attacking SUF-CMA is τ, its running time is T. We denote character with $*$ as the forged ciphetext and its related variables

GAME G0: In *GAME G0*, we just use the standard technique of simulating hash function. We can know this environment and the really environment is indistinguishable in the random oracle model. Let S_0 denote attacking successfully, assume $\Pr[S_0] = \varepsilon$.

Table 3 Simulation in *GAME G0*
for SUF-CMA Proof of Generalized Signcryption in Signcryption-mode

Simulate Random Oracle LH,	Query $LH(x)$: if the record (x, lh) is found in LH-list, then Oracle return lh, else randomly choose $lh \in \{0,1\}^{l+z+|sh|}$, add (x, lh) to the H-list.
Simulate Random Oracle K	Query $K(y)$: if the record (y, k) is found in K-list, then Oracle return k, else randomly choose $k \in \{0,1\}^{z+|sig|}$, add (y, k) to K-list
Simulate Random Oracle H	Query $H(m \| SH(Q_A \| Q_B) \| K_{sig})$: if the record $(m \| SH(Q_A \| Q_B) \| K_{sig}, h)$ is found in H-list, then Oracle return h, else, randomly choose $h \in \{0,1\}^{|p|}$, add record $(m \| SH(Q_A \| Q_B) \| K_{sig}, h)$ to H-list.
Simulate Random Oracle MAC	Query $MAC(K_{mac}, m \| SH(Q_A \| Q_B) \| s)$: If the record $(K_{mac}, m \| SH(Q_A \| Q_B) \| s, mac)$ is found in MAC-list, then Oracle return mac, else randomly choose $mac \in \{0,1\}^z$, add the record $(K_{mac}, m \| SH(Q_A \| Q_B) \| s, mac)$ into the MAC-list
Simulate Signcryption Oracle GSC	Real Signcryption in real environment. In *GAME G0* assume adversary can get this service.
Simulate Unsigncryption Oracle UGSC	Think about insider adversary. Because the adversary know the receiver's private key, he can get this integrated service (The simulator just gives the receiver's private key to the adversary.)
How to forge valid signcryption ciphertext	Assume the forged ciphetext is $\omega^*=(c^*, R^*, s^*)$, the only restriction is that ω^* was not queried to SC .Totally there are two methods of forging ciphetext: One is by attacking signcryption directly, the other is utilizing Sign Oracle. Note the adversary can forge new valid signcryption ciphetext by utilizing Sign Oracle.
Simulate Encryption Oracle ENC	Because the adversary can get the Encryption Oracle service by only needing to know the receiver's public key, but this is public to all. So the adversary can get the integrated service. (The simulator just gives the receiver's public key to the adversary.)
Simulate Decryption Oracle DEC	Think about insider adversary. Because the insider adversary know the receiver's private key, he can get the integrated service. (The simulator just gives the receiver's private key to the adversary.)
Simulate Sign And Verify Oracle Sign/Ver	In *GAME G0* assume the adversary can get the integrated service of Sign Oracle. Because implementing Verify Oracle just needs the signer's public key, and the public key is know to all. So the adversary can get this integrated service.

GAME G1: In this game, we will remove the restriction of linkage of encryption and signature in simulating GSC Signcryption Oracle. We remove the layer of encryption and reduce signcryption scheme to ECDSA signature scheme. We will substitute Sign Oracle by ECDSA algorithm. Other oracles are simulated as in *GAME G0*.

Table 4 Simulation in *GAME G1* for SUF-CMA Proof of Signcryption

Simulate Signcryption Oracle GSC and Unsigncryption Oracle UGSC in Random Oracle model	1. Add new elements of $(\square,(K_{mac},K_{sig}))$ in K-list. Note we must set the first item of new element vacant; we give it some value later. Add new elements of $(\square,K_{enc})$ in H-list. We also set the first item of new element vacant, we will give it some value later. 2. Call algorithm of ECDSA（ m|| SH（Q_A||Q_B）||K_{sig}, d_A）in Random Oracle, let（m|| SH（Q_A||Q_B）||K_{sig}, R, s）be the output result. In this process there will be a H-list. 3.Find element of （ $K_{mac},m\|\|SH(Q_A\|\|Q_B)\|\|s$ ） in MAC-list. If $(K_{mac},m\|\|SH(Q_A\|\|Q_B)\|\|s,mac)$ is found in the MAC-list, then we return mac.Else, choosing randomly $mac\in\{0,1\}^z$，return mac, add record of $(K_{mac},m\|\|SH(Q_A\|\|Q_B)\|\|s,mac)$ in *MAC*-list 4.Compute $c\leftarrow(m\|\|SH(Q_A\|\|Q_B)\|\|mac)\oplus K_{enc}$ 5.Let (c,R,s) be the output of Signcryption Oracle GSC when the input is （m, d_A, Q_A ,Q_B） Now we think about how to map vacant of elements in K-list and H-list to x_2,y_2.Because the simulator know the private key, so it can decryption the ciphertext. First we show how to simulate the Unsigncryption Oracle, in this process, we can give this map 1. Query (c,R,s) to Unsigncryption Oracle UGSC 2. The simulator compute $(x_2,y_2)=d_BR$ 3. First we find s in the second item of $(K_{mac},m\|\|SH(Q_A\|\|Q_B)\|\|s,mac)$ *MAC*-list. If s is found in $(K_{mac},m\|\|SH(Q_A\|\|Q_B)\|\|s,mac)$, return K_{mac}，$m\|\|SH(Q_A\|\|Q_B)$，mac，else return “Invalid Ciphertext”. 4. Next find K_{mac} in the second item of elements in K-list. If K_{mac} is found in $(\square,(K_{mac},K_{sig}))$-list, let the first item of this element be y_2, else return “Invalid Ciphertext” 5. Compute $t=c\oplus m\|\|SH(Q_A\|\|Q_B)\|\|mac$ and find t in the LH-list. If t is found equal to some element of $(\square,K_{enc})$, then let the first item of this element be x_2, else return “Invalid Ciphertext”.
Simulate Sign Oracle SIGN	Using algorithm of ECDSA（ m|| SH（Q_A||Q）, d_A）,let its output be Sign Oracle’s output.

GAME G1 and *GAME G0* are indistinguishable, except some queries have been given to K-list, LH-list before simulation or some ciphertexts have been guessed correctly by adversary. Assume the adversary has queried K-Random Oracle, H-Random Oracle, LH-Random Oracle, MAC-Random

Oracle $q_K, q_H, q_{LH}, q_{MAC}$ times, denote S_1 as the adversary forges successfully in GAME G1, then

$$|\Pr[S_0] - \Pr[S_1]| \leq \frac{q_H}{2^{|p|}} + \frac{q_{LH}}{2^{l+z+|SH|}} - \frac{q_H}{2^{|p|}} * \frac{q_{LH}}{2^{l+z+|SH|}} * \frac{q_{MAC}}{2^z} * \frac{q_K}{2^{z+|sig|}}$$

Table 5 Simulation in *GAME G2* for SUF-CMA Proof of Signcryption

How to forge the ciphertext	Assume the forged ciphetext is $\omega^*=(c^*, R^*, s^*)$, the only restriction is that ω^* was not queried to SC .In *GAME G2* there is only one method of forging ciphetext— utilizing attacking on signcryption. If the adversary can get (m, R, s) from Sign Oracle, s is the signature of message of format $m\|\| SH(Q_A\|\|Q)$.He cannot forge a signature of message of format $m \| SH(Q_A \| Q_B) \| K_{sig}$, the probability of forging signcryption ciphertext is the probability of forging signature of ECDSA.

GAME G2:The difference between *GAME G2* and *GAME G1* lies in how to forge signcryption ciphertext。Denote S_2 as the adversary forging successfully in *GAME G2*, and then we have

$$\Pr[S_2] = \Pr[S_1] - \tau .$$

Theorem 1 If the adversary *A* can forge signature of ECDSA with probability τ ,and the running time is T .Assume *A* queries *K* -Random Oracle, *H* -Random Oracle, *LH* -Random Oracle, *MAC* -Random Oracle $q_K, q_H, q_{LH}, q_{MAC}$ times, queries Signcryption Oracle, Sign Oracle, Encryption Oracle, Unsigncryption Oracle, Verify Oracle, Decryption Oracle $q_{GSC}, q_{SIGN}, q_{ENC}$ $q_{UGSC}, q_{VER}, q_{DEC}$ times. Then he forges valid signcryption ciphertext of Generalized Signcryption in signcryption-mode successfully with probability

$$\varepsilon \geq 2\tau - (\frac{q_H}{2^{|p|}} + \frac{q_{LH}}{2^{l+z+|SH|}}) + \frac{q_H}{2^{|p|}} * \frac{q_{LH}}{2^{l+z+|SH|}} * \frac{q_{MAC}}{2^z} * \frac{q_K}{2^{z+|sig|}}$$

The running time

$$T' \leq T + (q_{LH} + q_K) f + (q_{GSC} + q_{SIGN}) g$$

f denote the running time of compute $d_B R$ one time, g denote the running time of compute kG one time

3.4.2Prove Confidentiality of the Generalized Signcryption in Signcryption-mode

We reduce confidentiality of the Generalized Signcryption in signcryption-mode to confidentiality of ECATEOTP. We give an encryption scheme as following. Assume the success probability of forging Valid Ciphertext of ECATEOTP is η , and running time is T

Table 6 An Encryption Scheme ECATEOTP

Encryption ECATEOTP ENC(m, Q_A ,Q_B)
1. $k \in_R \{1,\ldots,n-1\}$ 2. $(x_1, y_1)=R \leftarrow [k]G$ 3. $(x_2, y_2)= [k]Q$ 4.$Kenc \leftarrow LH(x_2)$, $(Kmac, Ksig) \leftarrow K(y_2)$

5. e←$MACK_{mac}$ (m || SH（Q_A||Q_B）) 6. c←(m ||SH（Q_A||Q_B）||e)⊕ $Kenc$ Return ω=(c, R).
Decryption ECATEOTP DEC(ω , d_B ,Q_A ,Q_B)
1. [d_B]R=(x_2, y_2) 2.$Kenc$←LH(x_2)，($Kmac$, $Ksig$) ←K (y_2) 3. (m|| SH（Q_A||Q_B）||e) ←c⊕$Kenc$ 4. e'← $MACK_{mac}$ (m || SH（Q_A||Q_B）) If e≠e'，return ⊥；else，return m

GAME G0: In *GAME G0*, we just use the standard technique of simulating hash function. Then this environment and the really environment is indistinguishable for the attacker in random oracle model. Let S_0 denote attacking successfully, we define

$$\Pr[S_0] = \gamma$$

Table 7 Simulation in *GAME G0*
for Confidentiality Proof of Generalized Signcryption in Signcryption-mode

Simulate Random Oracle LH,K, H,MAC	The same as simulating Random Oracle LH,K, H,MAC in Table 3
Simulate Signcryption Oracle GSC	Think about insider adversary. Because the adversary know the sender's private key, he can get this integrated service.
Simulate Unsigncryption Oracle UGSC	Real Unsigncryption under real environment. Assume adversary can get this service
How to decrypt challenge ciphertext	Denote the challenge ciphertext (c^*, R^*, s^*). There are two ways to decrypt the challenge ciphertext: One is to utilize attacking on the signcryption scheme. The other is to use Decryption Oracle.
Simulate Encryption Oracle ENC	The adversary can get the Encryption Oracle service by only needing to know the receiver's public key. And this is public to all, so the adversary can get this integrated service.
Simulate Decryption Oracle DEC	Assume the adversary can get this integrated service
Simulate Sign Oracle SIGN	Think about insider adversary. Because insider adversary know the receiver's private key, he can get this integrated service.
Simulate Verify Oracle VER	The adversary can get the Verify Oracle service by only needing to know the sender's public key, but this is public to all. So the adversary can get this integrated service.

GAME G1: In this game, we try to reduce Unsigncryption Oracle to Decryption Oracle of ECATEOTP and substitute Decryption Oracle of Generalized Signcryption by Decryption Oracle of ECATEOTP.

Table 8 Simulation in *GAME G1* for Confidentiality Proof of Signcryption

Simulate Signcryption Oracle GSC	Everything is done honestly just as in the real Signcryption Algorithm. But when some queries to the Random Oracle LH, K, H, and MAC, we return something following the standard technique of simulating Hash Function.
Simulate Unsigncry-ption Oracle UGSC	1. There have been LH, K, H, MAC-list in simulate Signcryption Oracle GSC 2. Using Decryption Oracle of ECATEOTP: DEC(ω, d_B, Q_A, Q_B) in Random Oracle 3 .Algorithm DEC will compute $(x_2, y_2) = [d_B]R$,it must get value of $LH(x_2)$, K (y_2) according to LH-list, K-list. It finds (x_2, K_{enc}) and $(y_2, (K_{mac}, K_{sig}))$ in K-list and LH-list. If the element is found, then return the second item of element; else return "Invalid Ciphertext" 4. Compute $(m \| Bind \| e) \leftarrow c \oplus K_{enc}$; 5.Find (Kmac，$m$ || SH（$Q_A \| Q_B$）|| s) in MAC-List. If element of （(Kmac，m || SH（$Q_A \| Q_B$）|| s)，mac）is found, Simulator return Mac. Else return "Invalid Ciphertext" 6.Let e′← mac；If e≠e′, return ⊥; 7.Find m|| SH（$Q_A \| Q_B$）||K_{sig} in the first item of elements in H-List. If （m || SH（$Q_A \| Q_B$）|| K_{sig} ，h）is found, Simulator return .Else return "Invalid Ciphertext"; 8.Compute $u_1 \leftarrow s^{-1} * h$； $u_2 \leftarrow s^{-1} * r$; 9.Compute $R' \leftarrow [u_1]G + [u_2]Q_A$; If $R' \neq R$, return ⊥； else return m.
Simulate Decryption Oracle DEC	Using algorithm of DEC (ω, d_B, Q, Q_B), let its output be Decryption Oracle's output

GAME G1 and *GAME G0* are indistinguishable, except some ciphertexts have been guessed validly by adversary. Assume the adversary has queried K -Random Oracle, H -Random Oracle, LH -Random Oracle, MAC -Random Oracle $q_K, q_H, q_{LH}, q_{MAC}$ times, denote S_1 as the adversary forges successfully in *GAME G1*, then

$$|\Pr[S_0] - \Pr[S_1]| \leq \frac{q_H}{2^{|p|}} * \frac{q_{LH}}{2^{l+z+|SH|}} * \frac{q_{MAC}}{2^z} * \frac{q_K}{2^{z+|sig|}}$$

Table 9 Simulation in Game G2 for Confidentiality Proof of Signcryption

How to decrypt challenge ciphertext	In *GAME G2* there is only one method of decrypt challenge ciphetext—utilizing attacking the signcryption algorithm. In *GAME G2*, we remove the probability of decrypt challenge ciphertext by Decryption Oracle. If we utilize Decryption Oracle to decrypt, we must first transfer the challenge ciphertext to valid encryption ciphertext, that is, $f(c, R, s) = (c', R')$ .But we note c is related to s through authentication technique. So this probability is equal to probability of forging valid ciphertext

GAME G2:The difference between *GAME G2* and *GAME G2* comes from how to decrypt challenge ciphertext。Denote S_2 as the adversary decrypt successfully in *GAME G2*, and then we have

$$\Pr[S_2] = \Pr[S_1] - \eta = \eta \quad .$$

We can get

$$\Pr[S_0] \geq 2\eta + \frac{q_H}{2^{|p|}} * \frac{q_{LH}}{2^{l+z+|SH|}} * \frac{q_{MAC}}{2^z} * \frac{q_K}{2^{z+|sig|}}$$

Theorem 2 If the adversary A can forge valid ciphertext of ECATEOTP with probability η, the running time is T .Assume A queries K -Random Oracle, H -Random Oracle, LH -Random Oracle, MAC -Random Oracle $q_K, q_H, q_{LH}, q_{MAC}$ times, queries Signcryption Oracle, Sign Oracle, Encryption Oracle, Unsigncryption Oracle, Verify Oracle, Decryption Oracle $q_{GSC}, q_{SIGN}, q_{ENC}$, $q_{UGSC}, q_{VER}, q_{DEC}$ times. Then he can attack confidentiality of Generalized Signcryption in signcryption-mode successfully with probability

$$\gamma \geq 2\eta + \frac{q_H}{2^{|p|}} * \frac{q_{LH}}{2^{l+z+|SH|}} * \frac{q_{MAC}}{2^z} * \frac{q_K}{2^{z+|sig|}}$$

The running time

$$T' \leq T + (q_{LH} + q_K) f + (q_{GSC} + q_{SIGN} + q_{ENC} + q_{UGSC} + q_{VER} + q_{DEC}) g$$

f denote the running time of compute $d_B R$ one time, g denote the running time of compute kG one time.

3.4.3 Prove SUF-CMA of the Generalized Signcryption in Sgnature-mode

When Generalized Signcryption Oracle work as a signature scheme, Generalized Signcryption is actually ECDSA. So we omit the proof and give the following theorem

Theorem 3 If the adversary A can forge valid signature of ECDSA with probability η, the running time is T . Then he can attack SUF-CMA of Generalized Signcryption in signature-mode successfully with probability

$$\nu \geq 2\tau$$

The running time $T' \leq 2T$.

3.4.4 Prove Confidentiality of the Generalized Signcryption in Encryption-mode

When Generalized Signcryption Oracle work as an encryption scheme, Generalized Signcryption is actually ECATEOTP. So we omit the proof and give the following theorem

Theorem 4 If the adversary A can forge valid ciphertext of ECATEOTP with probability η, and the running time is T . Then he can attack confidentiality of Generalized Signcryption in encryption-mode successfully with probability

$$\mu \geq 2\eta$$

The running time $T' \leq 2T$.

4. Conclusion

Based on Han et al's paper [3] [4], our paper tentatively analysis the formal model of Generalized Signcryption.We compare it with the usual signcryption and claim its advantage. We give an improved Generalized Signcryption scheme based on ECDSA and give its security proof by using theory of provable security.

We note that Dodis et al's paper [9] [10] also give a Generalized Signcryption scheme. The technique in their paper is padding message before processing. In the two extremities, the scheme turns to be OAEP-padding and PSS-padding. In the non-extremity, the scheme turns to be signcryption. So we can see merge encryption, signature and signcryption into one primitive is not as difficult as it seems.

As we can see, Generalized Signcryption is a new cryptographic primitive with good property. It can bring many benefits to a lot of applications. We remark that this paper just gives a Generalized Signcryption scheme based on ECC, Generalized Signcryption schemes based on DL or IF problems have not been proposed, and we note that RSA-TBOS is not a good Generalized Signcryption scheme, so they are open problems.

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 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 1294. Berlin: Springer-Verlag, 1997, 165-179
3. HanYiliang, Yang Xiaoyuan. ECGSC: Elliptic Curve based Generalized Signcryption Scheme, Cryptology Eprint Archive, 2006/126.
4. HanYiliang, Yang Xiaoyuan. New ECDSA-Verifiable Generalized Signcryption, Chinese Journal of Computer, 2006, 11, 2003-2012.
5. Malone-Lee J., 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 2612, Berlin: Springer-Verlag, 2003, 210-224
6. Y.Zheng, H. Imai. How to construct efficient signcryption schemes on elliptic curves. Information Processing Letters, 1998, 68(5): 227-233
7. Bellare M., 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 1976, Berlin: Springer-Verlag, 2000, 531-545
8. 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 2332. Berlin: Springer-Verlag, 2002, 83-107
9. Dodis Y., Rreedman M., Jarecki S. and Walfish S., Optimal signcryption from any trapdoor permutation. Cryptology ePrint Archive, Report: 2004/020, 2004
10. 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
11. Dent Alexander W. Hybrid Signcryption Schemes With Outsider Security. In: Proceedings of The 8th Information Security Conference(ISC 2005), Singapore, 2005, 203-217
12. Dent Alexander W. Hybrid Signcryption Schemes With Insider Security. In: Proceedings of Information Security and Privacy - ACISP 2005, Brisbane, Australia, 2005, 253-266
13. Bellare M., 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
14. 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 2274, Berlin: Springer-Verlag, 2002, 80-98
15. 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 2442, Berlin: Springer-Verlag, 2002, 93–110
16. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption - How to Encrypt with RSA. In Eurocrypt'94, LNCS 950, pages 92~111. Springer-Verlag, Berlin, 1995.
17. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures -How to Sign with RSA and Rabin. In Eurocrypt '96, LNCS 1070, pages 399~416. Springer-Verlag, Berlin, 1996.