Formal Security Proof for Generalized Signcryption *

Jindan Zhang[1], Xu an Wang[2]

Department of Electronic Information
Xianyang Vocational Technical College, 712000, P. R. China
Key Laboratory of Information and Network Security
Engineering College of Chinese Armed Police Force, 710086, P. R. China
E-mail:wangxahq@yahoo.com.cn

Abstract— Generalized signcryption which proposed by Han is a new cryptographic primitive which can work as an encryption scheme, a signature scheme or a signcryption scheme [10]. However, the security proof in their paper is uncorrect. In this paper, we give security notions for this new primitive. And we give an attack to [9] and propose an improved generalized signcryption scheme. Furthermore, we give formal security proofs for this new scheme.

I. INTRODUCTION

In 1997, Zheng proposed a new cryptographic primitive:Signcryption [14]. The idea is compressing two indepen-dent operations (encryption and signature) in one operation(signcryption). In 2006, Han proposed a new primitive gen-eralized signcryption [8]. 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.

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 sign- cryption.We must point out here that Generalized Signcryption can not substitute of encryption or signature. But it fit some particular application perfectly.

A. Our Contribution

However, [10] do not give the formal model for this new primitive and unfortunately the security proof for their scheme is uncorrect.Actually, all the papers [3], [6], [7]mentioned above do not consider formal security model for this multi-functionality cryptographic primitive. In this paper, we recon-sider this new primitve thoroughly. our contribution are as following:First we give security notions for this new primi-tive.Secnond,we give an attack to [9]which is the first vision of [10] and propose an improved generalized signcryption scheme. Third, we give correct proofs for this new scheme.

II. GENERALIZED SIGNCRYPTION

A. Definition of Generalized Signcryption and a Concrete Scheme ECGSC

Definition 1: Given a normal secure signature scheme SIG = (Gen, Sig, Ver) where Gen is a key gen-
eration algorithm, $\tau \leftarrow Sig(m, SDKS)$, $(T, \perp) \leftarrow Ver(\tau, V EKS)$, a normal secure encryption scheme ENC =
(Gen, Enc, Dec) where Gen is the same algorithm as SIG's Gen, $\varepsilon \leftarrow Enc(m, V EKR)$, $m \cup \{\perp\} \leftarrow Dec(\varepsilon, SDKR)$ and a normal secure signcryption scheme SC = (Gen, Sc, U sc) where Gen is the same algorithm as SIG's Gen, $w \leftarrow Sc(m, SDKS, V EKR)$, $(m \cup \{\perp\}) \cup (T, \perp) \leftarrow U sc(w, SDKR, V EKS)$.A generalized signcryp-tion scheme GSC = (Gen, Gsc, U gsc) should be constructed satisfying the following:

1) KeyGen: Must be the same algorithm as Gen.

2) Generalized Signcryption:

For $m \in M$, $w \leftarrow Gsc(m, SDKS, V EKR)$. When S is a special value, $Gsc(m, SDKS, V EKR) = Enc(m, V EKR)$; When R is a special value, $Gsc(m, SDKS, V EKR) = Sig(m, SDKS)$; When S and R are both not special values, $Gsc(m, SDKS, V EKR) = Sc(m, SDKS, V EKR)$.

※该论文是2009IEEE电子商务和信息系统安全国际会议论文（EI：20094212377064）。
作者简介：张金丹（1983—），女，硕士，助讲，主要从事计算机专业教学及研究工作。

3) Generalized Unsigncryption: For w∈C, (m ∪ {⊥}) ∪ (T, ⊥) ←U gsc(w, SDKR, V EKS). When S is a special value, U gsc(w, SDKR, V EKS) = Dec(ε , SDKR); When R is a special value,U gsc(m, SDKS , V EKR) = V er(τ , V EKS); When S and R are both not special values, U gsc(w, SDKR, V EKS) = U sc(w, SDKR, V EKS).

In 2006, Han proposed a Generalized Signcryption ECGSC based on ECDSA. For page limitation, we do not describe the scheme here, readers please refer to [8] - [10].

III. AN IMPROVED GENERALIZED SIGNCRYPTION BASEDON ECDSA

A. An attack on this Scheme and Some Remarks In the ECGSC scheme the adversary intercept the ciphertext w = (c, R, s)set s = Φ, query the new ciphertext w = (c, R, Φ) 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 w = (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.

Remark 1 The origin scheme depend on hash function with additional property, that is, H (0) → 0,K(0) → 0,LH (0) →0,M AC(0) → 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.

Remark 2 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 = Φ , he would get the plaintext m. So we suggest delete the if-clause in the algorithm.

B. An Improved Generalized Signcryption Based on ECDSA

In this section, we give an improved Generalized Signcryp- tion scheme. Improved scheme has the same parameter, syntax with the origin scheme. But we do not need hash function satisfy H (0) → 0,K(0) → 0,LH (0) → 0, M AC(0) → 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 benefitting encryption- mode and signature-mode. We define a function f (t). if t = Q, f (t) = 0,if t ≠ 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 $|K_{sig}|$ = |sig|.

1) Parameters: Same as the original scheme.

2) Syntax:Almost same as the original scheme except we do not need hash functions with additional prop-erty, introduce a new point and modify some syntex' s meaning.

- we do not need hash function satisfy H (0) →0,K(0) → 0,LH (0) → 0, M AC(0) → 0;
- 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 benefitting encryption-mode and signature-mode. We define a function f (t). if t = Q,f (t) = 0,if t ≠ Q, then f (t) = 1;
- 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$|K_{sig}|$ = |sig|;
- 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).

3) Key generation(n, T):Same as the original scheme.

4) Generalized Signcryption SC(m, d_A, Q_A, Q_B): it con-sists of seven algorithms

- Compute f (Q_A),f (Q_B),
- k ∈R 1, • • • , n − 1;
- R = [k]G = (x_1, y_1),r = x_1 mod p;

• $[k]P_B = (x_2, y_2)$;
• $K_{enc} = f(Q_B)LH(x_2), (K_{mac}, K_{sig}) = f(Q_B)K(y_2)$;
If $d_A = 0$, $s = \Phi$, Else $s = k^{-1}(f(Q_A)H(m \text{ Bind } K_{sig}) + f(Q_A) \cdot rd_A) \bmod n$;
• $e = f(Q_B)M\,ACK_{mac}(m)$;
• $c = (m\ e) \oplus K_{enc}$, Return $w = (c, R, s)$.

5) Generalized Unsigncryption DSC(w, d_B, Q_A, Q_B) :it also consists of seven algorithms
• Compute $f(Q_A)$, $f(Q_B)$,
• $r = x(R)$(R' s x axiom);
• $(x_2, y_2) = [d_B]R$;
• $K_{enc} = f(Q_B)LLH(x_2)$, $(K_{mac}, K_{sig}) = f(Q_B)LK(y_2)$;
• $(m||e) = c \oplus K_{enc}$;
• $e' = f(Q_B)LM\,ACK_{mac}(m)$, If $e \neq e'$, return $\perp$ else if $s = \Phi$, return m;
$u1 = s^{-1} f(Q_A) \cdot H(m||Bind||K_{sig})$, $u2 = s^{-1} \cdot f(Q_A) \cdot r$;
• $R' = [u_1]G + [u_2]Q_A$; If $R' \neq R$, return $\perp$, else return m.

IV. SECURITY PROOFS FOR OUR IMPROVED GENERALIZED SIGNCRYPTION

The idea of the origin scheme' s 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 [12], and this scheme is proved to be ciphetext unforge-able 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. [10] can also reduce SUF-CMA of signcryption to SUF-CMA of ECDSA, but the reduction is uncorrect. Also[10] do not give secrurity proof for generalized signcryption working in encryption-mode and signature- mode. This paper tries to solve these problems.

A. 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:G0, G1. they are reduced from the real attacking game. 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 GAM EG0, 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 $Pr[S_0] = \varepsilon$.

1) Simulate Random Oracle LH (x)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;

2) Simulate Random Oracle K(y):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) toK-list.

3) 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 $h \in \{0, 1\}^{lp|}$ add record $(m \| SH(Q_A \| Q_B) \| K_{sig}, h)$ to H -list.

4) Simulate Random Oracle M AC:Query M AC(Kmac, $m \| SH(Q_A \| Q_B) \| s$), If the record (Kmac, $m \| SH(Q_A \| Q_B) \| s$, mac) is found in M AC-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.

5) Simulate Signcryption Oracle Sc:Real Signcryption in real environment. In assume adversary can get this service.

6) Simulate Unsigncryption Oracle Usc: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).

7) 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).

8) Simulate Decryption Oracle Dec:Think about insider adversary. Because the insider adversary know the re-ceiver’s private key, he can get the integrated service.(The simulator just gives the receiver’s private key to the adversary).

9) Simulate Sign And Verify Oracle Sig/Ver:In this game, 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 known to all. So the adversary can get this integrated service.

10) How to forge valid signcryption ciphertext:Assume the forged ciphetext is $w^{\square} = (c^{\square}, R^{\square}, s^{\square})$ the only restriction is that $w^{\square}$ was not queried to Sc Oracle.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.

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 GAM EGO.

1) Simulate Signcryption Oracle Gsc

• Add new elements of ($\diamond$, (K_{mac}, K_{sig})) in K-list.Note we must set the first item of new element va-cant; we give it some value later. Add new elements of ($\diamond$, K_{enc}) in H -list. We also set the first item of new element vacant, we will give it some value later.

• Call algorithm of ECDSA($m \| SH(Q_A \| Q_B) \| K_{sig}$, d_A)in Random Oracle, let($m \| SH(Q_A \| Q_B) \| Ksig$, R, s)be the output result. In this process there will be a H -list;

• Find element of (K_{mac}, $m \| SH(Q_A \| Q_B) \| s$)in M AC-list. If (K_{mac}, $m \| SH(Q_A \| Q_B) \| s$, K_{mac}) is found in the M AC-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 M AC-list;

• Compute $c = (m \| SH(Q_A \| Q_B) \| mac) \oplus K_{enc}$;

• Let (c, R, s) be the output of Signcryption Oracle Gsc when the input is (m, d_A, Q_A, Q_B);

2) 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.

3) Simulate Unsigncryption Oracle Ugsc

• Query (c, R, s) to Unsigncryption Oracle Ugsc;

• The simulator compute (x_2, y_2) = dB R;

• First we find s in the second item of (Kmac, $m \| SH(Q_A \| Q_B) \| s$, mac) M AC-list. If s is found in (K_{mac}, $m \| SH(Q_A \| Q_B) \| s$, mac), return $K_{mac}m \| SH(Q_A \| Q_B) \| s$, mac else return ”Invalid Ciphertext”;

• Next find K_{mac} in the second item of elements in K-list. If K_{mac} is found in ($\diamond$, (K_{mac}, K_{sig}))-list, let the first item of this element be y_2, else return ”Invalid Ciphertext”;

• 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 ($\diamond$, K_{enc}), then let the first item of this element be x2, else return ”Invalid Ciphertext”.

4) Simulate Sign Oracle Sig:Using algorithm of ECDSA($m \| SH(QA \| Q)$), dA), let its output be Sign Oracle’s output.

Remark 3:In the above simulation,we use a technique different from usual. Here we use the condition that attacker can know the receiver’s private key and can compute [dB]R and x2, y2.So we can find the relationship between x2, y2 and (Kmac, Ksig),Kenc.

GameG1 and GameGO 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,M AC-Random Oracle qK , qH , qLH , qM AC times, denote S1 as the adversary forges successfully in GAME G1, then

$|\Pr[S_0]-\Pr[S_1]| \le \frac{q_H}{2^{|p|}}+\frac{q_{LH}}{2^{l+z+|SH|}}-\frac{q_H}{2^{|p|}}\cdot\frac{q_{LH}}{2^{l+z+|SH|}}\cdot\frac{q_{MAC}}{2^{Z}}\cdot\frac{q_K}{2^{Z+|Sig|}}$

Theorem 1: If the adversary A can forge valid signcryption ciphertext of Generalized Signcryption in signcryption-mode successfully with probability τ and the running time is T. Assume A queries K-Random Oracle, H-Random Oracle, LH-Random Oracle, MAC-Random Oracle qK, qH, qLH, qMAC times, queries Signcryption Oracle, Sign Oracle, Encryption Oracle, Unsigncryption Oracle, Verify Oracle, Decryption Oracle qGsc, qUgsc, qSig, qVer, qEnc, qDec times. Then he forges signature of ECDSA with probability ε,

$\varepsilon \ge \tau-\frac{q_H}{2^{|p|}}+\frac{q_{LH}}{2^{l+z+|SH|}}-\frac{q_H}{2^{|p|}}\cdot\frac{q_{LH}}{2^{l+z+|SH|}}\cdot\frac{q_{MAC}}{2^{Z}}\cdot\frac{q_K}{2^{Z+|Sig|}}$

The running time
T' ≥ T + (qLH + qK)f + (qGsc + qSig)g
f denote the running time of computedbR one time, g denote the running time of compute kG one time.

B. Prove Confidentiality of the Generalized Signcryption in Signcryption-mode We reduce confidentiality of the Generalized Signcryption in signcryption-mode to confidentiality of ECATEOTP which as following.

Definition 2: ECATEOTP is an encryption scheme, and we know it's IND-CCA2 secure [12].

1) Encryption Enc(m, QA, QB)
- k ∈R {1, • • • , n − 1};
- (x1, y1) = R = [k]G
- (x2, y2) = [k]Q;
- Kenc = LH (x2)‖ (Kmac, ‖Ksig) = K(y2);
- e = MAC Kmac (m ‖SH (QA ‖QB));
- c = (m ‖ SH (QA ‖ QB ‖ e) ⊕ Kenc;
- Return w = (c, R).

2) Decryption Dec(w, dB, QA, QB)
- [dB]R = (x2, y2);
- Kenc = LH (x2), (Kmac, ‖ Ksig) = K(y2);
- (m ‖ SH (QA ‖ QB) ‖ e) = c ⊕ Kenc;
- e = MAC Kmac (m ‖ SH (QA ‖ QB));
- if e = e' ,return ⊥;else return m.

Assume the success probability of forging Valid Ciphertext of ECATEOTP is η, and running time is T.

GAME G0: In GAMEG0, 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 S0 denote attacking successfully, assume Pr[S0] = γ.

1) Simulate Random Oracle LH (x), K(y), H, MAC: Same as common name oracles in section 4.1;
2) Simulate Signcryption Oracle Sc: Think about insider adversary. Because the adversary know the sender's sprivate key, he can get this integrated service;
3) Simulate Unsigncryption Oracle Usc: Real Unsigncryp-tion under real environment. Assume adversary can get this service;
4) 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;
5) Simulate Decryption Oracle Dec: Assume the adversary can get this integrated service;
6) Simulate Sign And Verify Oracle Sig/Ver: Think about insider adversary. Because insider adversary know the receiver's private key, he can get this integrated ser-vice. 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.
7) How to decrypt challenge ciphertext: Denote the chal-lenge ciphertext(c□, R□, s□). There are two ways todecrypt the challenge ciphertext: One is to utilize at-tacking on the signcryption scheme. The other is to use Decryption Oracle.

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..

1) 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 M AC, we return something following the standard technique of simulating Hash Function.

2) Simulate Unsigncryption Oracle Ugsc

• There have been LH , K, H , M AC-list in simulate Signcryption Oracle Gsc;

• Using Decryption Oracle of ECATEOTP:Dec(w, dB , QA, QB)in Random Oracle;

Algorithm Dec will compute (x2, y2) = [dB]R, it must get value of LH (x2)K(y2) according to LH -list, K-list. It finds (x2, Kenc) and (y2, (KM ac, Ksig)) in K-list and LH -list. If the element is found, then return the second item of element; else return ”Invalid Ciphertext”;

• Compute (m ‖ Bind ‖ e) = c ⊕ Kenc;

• Find m ‖ SHQA ‖ QB ‖ Ksig in the first item of elements in H -List. If (m ‖ SH (QA ‖ QB) ‖ Ksig , h) is found, Simulator return h. Else return”Invalid Ciphertext”;

Computer u1=s-1 • hu2 = s−1 • r;

• Compute R’ = [u1]G + [u2]QA If R’ ≠ R, return⊥else return m.

3) Simulate Decryption Oracle Dec:Using algorithm of Dec(w, dB , Q, QB), let its output be Decryption Ora-cle’s output.

GAM EG1 and GAM EG0 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, M AC-Random Oracle qK , qH , qLH , qM AC times, denote S1 as the adversary forges successfully in GAM EG1, then

$$\left|\Pr[S_0]-\Pr[S_1]\right| \le \frac{q_H}{2^{|p|}}\cdot\frac{q_{LH}}{2^{l+z+|SH|}}\cdot\frac{q_{MAC}}{2^{Z}}\cdot\frac{q_K}{2^{Z+|Sig|}}$$

Theorem 2: If the adversary A can attack confidentiality of Generalized Signcryption in signcryption-mode successfully with probability η , the running time is T . Assume A queriesK-Random Oracle, H -Random Oracle, LH -Random Oracle, M AC-Random Oracle times, queries Signcryption Oracle, Sign Oracle, Encryption Oracle, Unsigncryption Oracle, Verify Oracle, Decryption Oracle qGsc, qU gsc, qSig , qV er, qEnc, qDec times. Then he can attack IND-CCA2 property of ECATEOTP with probability

$$\zeta \succ \eta + \frac{q_H}{2^{|p|}}\cdot\frac{q_{LH}}{2^{l+z+|SH|}}\cdot\frac{q_{MAC}}{2^{Z}}\cdot\frac{q_K}{2^{Z+|Sig|}}$$

The running time T’ ≥ T + (qLH + qK)f + (qGsc +qSig + qUgsc, qVer, qEnc, qDec)g

f denote the running time of computedbR one time, g denote the running time of compute kG one time

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

$$\mu \approx \eta$$

The running time T’ =T.

D. Prove Confidentiality of the Generalized Signcryption in Encryption-mode. When Generalized Signcryption Oracle work as an encryp-tion scheme, Generalized Signcryption is actually ECATEOTP. So we omit the proof and give the following theorem.

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

$$\mu \approx \eta$$

The running time T’ ≈T.

V. CONCLUSION AND OPEN PROBLEMS

Based on Han et al’s paper [8] - [10] our paper pay attention to the formal model of Generalized Signcryption. We give an improved Generalized Signcryption

scheme based on ECDSA and give its security proof . We remark that this paper just gives a Generalized Signcryption scheme based on ECC, there are still much work can be done on this new primitive.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under contract no. 60842006.

REFERENCES

[1] J.H. An, Y. Dodis and T. Rabin. On the security of joint signature and encryption.In Advances in Cryptology, Proc. EUROCRYPT 2002, LNCS2332, pages 83 107. Springer Verlag, 2002.

[2] J. Baek, R. Steinfeld and Y. Zheng. Formal Proofs for the Security of Signcryption.In Public Key Cryptography' 02 (PKC 2002), LNCS 2274,pages. 80 98, Springer Verlag, 2002.

[3] X. Boyen. Multipurpose identity-based signcryption:a swiss army knife for identity-based cryptography.In (Crypto03), pages. 382 398, Springer Verlag, 2003.

[4] A. Dent. Hybrid Signcryption Schemes With Outsider Security.In Pro-ceedings of The 8th Information Security Conference (ISC 2005), LNCS4212, pages. 203 217, Springer Verlag, 2005.

[5] A. Dent. Hybrid Signcryption Schemes With Insider Security. In Proceed- ings of Information Security and Privacy 2005) (ACISP 2005), LNCS4307, pages. 253 266, Springer Verlag, 2005.

[6] Y. Dodis, M. Rreedman, S. Jarecki and S.Wal□sh. Optimal signcryption from any trapdoor permutation. Cryptology ePrint Archive, Report:2004/020, 2004.

[7] Y. Dodis, M. Rreedman, S. Jarecki and S.Wal□sh. Versatile padding schemes for joint signature and encryption. In Proceedings of Eleventh ACM Conference on Computer and Communication Security (CCS2004),pages 196 205. IEEE Computer Society, 2004.

[8] Y. Han , X. Yang. ECGSC: Elliptic Curve based Generalized Signcryption Scheme,Cryptology Eprint Archive, 2006/126.

[9] Y. Han, X. Yang. New ECDSA-Veri□able Generalized Signcryption.Chinese Journal of Computer, No. 11., pages. 2003 2012, 2006.

[10] Y. Han. Generalization of Signcryption for Resources-constrained Envi-ronments. Wireless Communication and Mobile Computing, pages. 919 931, 2007.

[11] J.M. Lee, W. Mao. Two birds one stone: Signcryption using RSA.In Topics in Cryptology - CT-RSA 2003, LNCS 2612, pages. 210 224.Springer Verlag, 2003.

[12] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In Advances in Cryptology,Proc. CRYPTO2001, LNCS 2139, pages 310 331. Springer Verlag, 2001.

[13] L. Sunder and K. Prashant. ID based generalized signcryption, Cryptol-ogy Eprint Archive, 2008/084.

[14] Y. Zheng. Digital signcryption or how to achieve cost (signature &encryption) cost (signature) + cost (encryption). In Advances inCryptology, Proc. CRYPTO 1997, LNCS 1294, pages 165 179. Springer Verlag, 1997