Certificateless Generalized Signcryption

Huifang Ji , Wenbao Han, Long Zhao
ZhengZhou Information Science and Technology Institute,
Zhengzhou, 450002, China
huifangji@126.com, *wb.han@263.net*, *zhaolong_email@sohu.com*

Abstract

Generalized Signcryption is a fresh cryptographic primitive that not only can obtain encryption and signature in a single operation, but also provives encryption or signature alone when needed. This paper gives a formal definition of certificateless generalized signcryption and its security model is present. A concrete certificateless generalized signcryption scheme is also proposed in this paper.

Keywords: bilinear pairing; signcryption; generalized signcryption; certificateless cryptography;

1. Introduction

Signcryption is a cryptographic primitive proposed by Zheng in 1997 that could obtain encryption and signature in a single operation, which is more efficient than the traditional signature-then-encryption[1]. Since then, many schemes of signcryption are proposed. Malone-Lee gave the first identity based signcryption scheme in 2002[2], following which several other identity based signcryption were proposed.

Since the key escrow property of identity based cryptosystems is inherent, to avoid this problem, Al-Riyami and Paterson proposed a new cryptographic primitive as certificateless public key system[3]. The users' private key is computed by a KGC and the users themselves, which eliminates the key escrow problem in identity based system and cumbersome certificate management problem in traditional public key system. Many certificateless signature and encryption schemes are provided. The first certificateless signcryption was given by M.Barbosa and P.Farshim in 2003[4], several other certificateless signcryption were proposed since then[5][6].

The notion of generalized signcryption was termed by Han Yiliang and Yang Xiaoyuan in 2006[7]. The idea is that using a special "signcryption", one not only can simultaneously get confidentiality and authentication, but also obtain confidentiality or authentication alone. This special “signcryption” is called *generalized signcryption*. Wang Xu-an el al gave the first security model for generalized signcryption and an improved generalized signcryption[8]. The first ID-based vision is proposed by S. Lal and P. Kushwah[9].

In this paper, we propose a certificateless generalized signcryption(CLGSC). First we define the formal definition of CLGSC; second, we give the security notions for this new primitive; third, a new CLGSC scheme is proposed.

The paper is organized as following: in the next section, we give the formal model and the security model of CLGSC. The definition of bilinear pairings and related computational hard problem are given in section 3. In section 4 we give the concrete CLGSC. The correctness and efficiency is analyzed in section 5. Section 6 concludes the paper.

2. Formal Model of CLGSC

2.1 Certificateless Generalized Signcryption

A certificateless generalized signcryption scheme is defined by the following five probabilistic polynomial-time algorithms.

1 Setup (1^k): Given a security parameter k, PKG executes this algorithm and generates a master key S and global parameters *params*. PKG publishes *params* and keep S secret.

2 Extract-Partial-Private-Key(*ID*, *S*, *params*). Given a user identity *ID*, PKG runs the algorithm and returns a partial private key *D*.

3 Set-User-Key (*ID*, *D*, *params*). Given a user *ID*, partial private key *D* and *params*, user runs the algorithm and returns a public key of the identity *PK*, and a secret key x, the private key of the user is (x, D).

4 GSC. This algorithm has 3 scenarios: signcryption, signature and encryption.

4.1 Signcryption: if user *A* transmits a message *m* confidentially and authenticately to *B*, the input is (S_A, m, ID_B), and outputs $s = GSC(S_A, m, ID_B)$.

4.2 Signature: if user *A* wants to sign a message *m* without definite receiver, the input is (S_A, m, ID_φ), where ID_φ means the receiver is null, the output is $s = GSC(S_A, m, ID_j)$.

4.3 Encryption: if someone wants to send message *m* to *B* confidentially, the input is (S_φ, m, ID_B), where S_φ denotes the private key corresponding to ID_φ. The output is $s = GSC(S_j, m, ID_B)$.

5 UGSC. Given σ, if it is valid, the receiver *B* unsigncrypts the ciphertext and returns *m* and (or) the signature on *m* by *A*, otherwise return $\perp$means fail.

2.2 Security model for CLGSC

Now we describe the security model for certificateless generalized signcryption under the inside attacker. In confidentiality and unforgeability game we provide access to the following oracles:

1. Extract Partial private key: given an identity *ID*, the oracle returns the partial private key *D* using the Extract-Partial-private-key algorithm.

2. Extract Secret Key: given an identity *ID*, the oracle returns the full secret key *SK*=(*x*, *D*) of *ID* using the Set-user-key algorithm.

3. Request public key: given an identity *ID*, the oracle returns the public key *PK* of *ID* using the Set-user-key algorithm.

4. Replace public key: given an identity *ID* and a valid public key *PK* ', this oracle replace the public key of *ID* with *PK*'. If the identity's public key doesn't exist, then it is obtained through the Set-user-key algorithm and then replaced by *PK*'.

5. GSC oracle: given a massage *m*, a sender identity *A*, a receiver identity *B*, this oracle returns the result of running algorithm GSC. Note that if *A* and *B* are not empty, then use the Signcryption model; if *A* is empty, use the encryption model; if *B* is empty, use the signature model. When the identity *A* isn't empty, and its' private key doesn't exist, first running the Set-user-key algorithm to get *A*'s full secret key and then running algorithm GSC.

6. UGSC oracle: given a ciphertext, sender identity *A*, a receiver identity *B*, the oracle returns the result of running UGSC algorithm.

As in many certificateless cryptosystems, we consider two types adversary, Type-I and Type-II adversary in the security definition of CLGSC. Roughly, the Type-I adversary models a common user without the master secret key, while the Type-II adversary models the honest but curious KGC.

Type-I adversary: since a Type-I adversary is a common user, he is allowed to request the above 6 oracles with the following constraint:

1. Adversary is not allowed to request the master secret key;

2. No extract secret key query is allowed on the challenge identities.

3. Adversary is not allowed to request the extract partial private key of the challenge identities.

Type-II adversary: a Type-II adversary is an honest but curious KGC, so he is given the master secret key, he is allowed to request the above 6 oracles with the following constraint:

1. No extract secret key query is allowed on the challenge identities.

2. No replace public key query is allowed on the challenge identities before the challenge phase.

Confidentiality

Definition 1: A certificateless generalized signcryption is called IND-CLGSC-iCCA2 secure if every of the probabilistic polynomial time Type-I or

Type-II adversary has negligible advantage in winning the following game between the challenger C and the adversary A:

Setup: Challenger C runs the setup algorithm to generate master key *Msk* and system parameters *Params*. C gives A *Params* while keeping *Msk* secret(C gives the *Msk* to A when A is a Type-II adversary). After receiving *Params*, A outputs a target identity ID^*. C interacts with A in following phases:

Phase 1: A is given access to the above all the six oracles. A adaptively queries the oracles consistent with the constraints described above.

Challenge: A outputs two message m_0, m_1, and a sender's identity ID_S, C randomly chooses a bit $b\epsilon_R$ {0,1}and computes a generalized signcryption $\sigma^* =$ GSC(m_b, ID_S, SK_S, PK_S, ID^*, PK^*) and sends σ^* to A.

Phase 2: A makes the same queries as in phase, besides it cannot query UGSC oracle on σ^* to for ID^*.

Guess: A output its guess b' on b at the end of the game. If $b' = b$, A wins the game. The advantage of A is defined as $Adv_A^{IND\text{-CLGSC-iCCA2}}$=|2Pr[$b$ = b']-1|.

Authenticity

Definition 2: A certificateless generalized signcryption is called strong existential unforgeability(sEUF-CLGSC-iCMA) if every of the probabilistic polynomial time Type-I or Type-II adversary has negligible advantage in winning the following game between the challenger C and the adversary F:

Setup: Challenger C runs the setup algorithm to generate master key *Msk* and system parameters *Params*. C gives F *Params* while keeping *Msk* secret. After receiving *Params*, F outputs target identity ID^*. C interacts with F in following phases:

Phase 1: F is given access to the above all the six oracles. F makes the same queries as in the game above.

Forgery: F output a signature σ^* and a receiver ID_R, we assume that $ID_R \neq ID^*$. If UGSC(σ^*, S_R, ID_R), returns m and σ^* was not the output of any GSC query GSC(m, ID^*, ID_R), then F wins the game. The probability that F wins the game is defined as $Adv_A^{sEUF\text{-CLGSC-iCMA}}$.

3. Preliminaries

We briefly review the basic definition of bilinear pairings and some related complexity assumptions.

3.1 Bilinear Pairings

Let G_1 and G_2 be two cyclic groups of prime order p and P be a random generator of G_1. The map e: $G_1 \times G_1 \rightarrow G_2$ is called an admissible bilinear pairing if the following conditions hold true.

1) e is bilinear, i.e. $e(aP, bP) = e(P, P)^{ab}$ for all $a, b \in Z_p$;

2) e is non-degenerate, i.e. $e(P, P) \neq 1_{G_2}$;

3) e is efficiently computable.

3.2 Complexity Assumptions

In this subsection, we show several computational assumptions related to bilinear pairing that are relevant to the security of our scheme.

1 **Strong Diffie-Hellman problem (SDHP)** Strong DHP is a stronger version of DHI (Diffie-Hellman Inversion problem). Given $(P, aP) \in G_1^2$ for any $a \in Z_p^*$, the SDH problem in G_1 is to compute $(h,(a+h)^{-1}P)$, $h \in Z_p^*$.

2 **Gap Bilinear Diffie-Hellman problem (GBDHP)** Given $(P, aP, bP, cP) \in G_1^4$ for any random $a,b,c \in Z_p^*$, the GBDH problem in (G_1, G_2, e) is to compute $e(P, P)^{abc}$ given access to DBDH oracle O which on input $(P, aP, bP, cP, T) \in G_1^4 \times G_2$.outputs 1 if $T = e(P, P)^{abc}$ and 0 otherwise. The advantage of any probability polynomial time algorithm A in solving the GBDH in (G_1, G_2, e) is defined as:

$Adv_A^{GBDH}(O,q_{DBDH})=Pr[A^O(P, aP, bP, cP) = e(P, P)^{abc} \mid a,b,c \in Z_p^*]$, where q_{DBDH} is the number of queries to the decisional oracle. We say that GBDHP is (t, ϵ, q_{DBDH}) hard if for any t time probabilistic algorithm A asking q_{DBDH} oracles queries, the advatange $Adv_A^{GBDH} < \epsilon$.

3 Collusion Attack Algorithm with *k*-traitors (*k*-CAA) Given $(P, aP, (h_1+a)^{-1}P, \ldots, (h_k+1)^{-1}P) \in G_1^{k+2}$ for any random $a \in Z_p^*$ and known values $h_1, \ldots, h_k \in Z_p^*$, the k-CAA problem in G_1 is to compute $(a+h)^{-1}P$ for some $h \notin \{h_1,\ldots,h_k\}$. The advantage of any probability polynomial time algorithm A in solving the k-CAA problem in G_1 is defined as:

$Adv_A^{k\text{-}CAA}=Pr[A(P, aP, (h_1+a)^{-1}P,\ldots, (h_k+1)^{-1}P, h_1,\ldots,h_k) = (a+h)^{-1}P \mid a,h \in Z_p^*, h \notin \{h_1,\ldots,h_k\}]$,

We say that k-CAA is (t, ϵ) hard if for any t time probabilistic algorithm A, the advatange $Adv_A^{k\text{-}CAA} < \epsilon$.

4 Modified BDHI for *k*-values (*k*-mBDHIP) Given $(P, aP, (h_1+a)^{-1}P,\ldots, (h_k+1)^{-1}P) \in G_1^{k+2}$ for any random $a \in Z_p^*$ and known values $h_1,\ldots, h_k \in Z_p^*$, the k-mBDHIP problem is to compute $e(P, P)^{(s+h)^{-1}}$ for some $h \notin \{h_1,\ldots,h_k\}$. The advantage of any probability polynomial time algorithm A in solving the k-mBDHIP problem in (G_1, G_2, e) is defined as:

$Adv_A^{k\text{-}mBDHIP}=Pr[A(P, aP, (h_1+a)^{-1}P,\ldots, (h_k+1)^{-1}P, h_1,\ldots,h_k) = e(P, P)^{(s+h)^{-1}} \mid a,h \in Z_p^*, h \notin \{h_1,\ldots,h_k\}]$,

We say that k- mBDHIP is (t, ϵ) hard if for any t time probabilistic algorithm A, the advatange $Adv_A^{k\text{-}mBDHIP} < \epsilon$.

4. A concrete CLGSC

We proposed a certificateless generalized signcryption as following.

Setup(1^k): given a security parameter 1^k, the KGC chooses two groups G_1 and G_2 of prime order p, two random generator P, Q of G_1 such that $P \neq Q$, and a bilinear map e: $G_1 \times G_1 \rightarrow G_2$. Compute $g = e(P, Q) \in G_2$, define 5 hash functions as H_1:$\{0,1\}^* \rightarrow Z_p^*$, H_2: $G_2 \times \{0,1\}^* \rightarrow Z_p^*$, H_3: $\{0,1\}^m \times G_2 \times \{0,1\}^* \times G_2 \times G_1 \times \{0,1\}^* \rightarrow Z_p^*$, H_4: $Z_p^* \times \{0,1\}^* \rightarrow Z_p^*$, H_5: $G_2 \times G_2 \times G_2 \times \{0,1\}^* \rightarrow \{0,1\}^{k_1+k_2}$, where k_1,k_2 denote the number of bits to represent G_1 and Z_p^*elements respectively. KGC chooses random $s \in Z_p^*$ as master secret key and set $P_{\mathrm{pub}} = sP$. KGC publishes the system parameters as $< G_1 , G_2 , P, Q, P_{\mathrm{pub}}, e$: $G_1 \times G_1 \rightarrow G_2, g, H_1 , H_2 , H_3 , H_4 , H_5 >$.

Extract-Partial-Private-Key: given ID_i, the partial private key of the user with identity ID_i is computed by KGC as $D_i = (q_i + s)^{-1}Q$, where $q_i = H_1 (ID_i)$.

Set-user-key: given D_i, the user with identity ID_i chooses random $x_i \in Z_p^*$ and sets his private key $SK_i =< x_i , D_i >$ and public key $PK_i =< PK_{i1}, PK_{i2}>=$ $< g^{x_i} , x_i T_i >$, where $T_i = (q_i + s) P$.

GSC: This algorithm has 3 scenarios: signcryption, signature and encryption.

Signcryption: given message m, sender' identity A, receiver's identity B, A operates the following steps:

1 A chooses random r, $r' \in Z_p^*$, computes $\alpha = g^r$;

2 computes $h = H_2(\alpha, m, ID_A, ID_B)$ and $h_3 = H_3(m, \alpha, h, ID_A, PK_{A1}, PK_{A2}, ID_B)$;

3 computes $Z= \frac{r}{(x_A+h_3)} D_A$;

4 computes $c = H_4(h, ID_A, ID_B) \oplus m \| \alpha$;

5 computes $h_5 = H_5(g^{r'}, (PK_{B1})^{r'}, PK_{B1}, ID_B)$

6 compute $d_1 = r' (q_B + s) P$ and $d_2 = h_5 \oplus h \| Z$.

7 return ciphertext $\sigma = (c, d_1, d_2, ID_B)$.

Signature: given message m, sender' identity A, A operates the following steps:

Step 1 is the same as in signcryption;

2 computes $h = H_2(\alpha, m, ID_A, 0)$ and $h_3 = H_3(m, \alpha, h, ID_A, PK_{A1}, PK_{A2}, 0)$;

3 computes $Z= \frac{r}{(x_A+h_3)} D_A$;

4 sets $c = m \| \alpha$;

5 computes $h_5 = 0$;

6 computes $d_1 = 0$ and $d_2 = h_5 \oplus h \| Z = h \| Z$;

7 returns ciphertext $\sigma = (c, d_1, d_2, 0)$.

Encryption: given message m, receiver's identity B, someone operates the following steps:

1 chooses random r, $r' \in Z_p^*$, computes $\alpha = g^r$;

2 computes $h = H_2(\alpha, m, 0, ID_B)$ and $h_3 = H_3(m, \alpha, h, 0, 0, 0, ID_B)$;

3 computes $c = H_4(h, 0, ID_B) \oplus m \| \alpha$;

4 computes $h_5 = H_5(g^{r'}, (PK_{B1})^{r'}, PK_{B1}, ID_B)$

5 compute $d_1 = r' (q_B + s) P$ and $d_2 = h_5 \oplus h \| 0$.

6 return ciphertext $\sigma = (c, d_1, d_2, ID_B)$.

UCLGSC: given σ, a receiver's identity B, operates the following steps:

1 computes $w' = e(d_1,D_B)$ and $(w')^{x_i}$ (if there is no receiver's identity, then $D_B = 0$, and $w' =1$);

2 sets $h_5'= H_5(w', (w')^{x_i}, PK_{B1}, ID_B)$;(if $D_B = 0$, then $h_5'=0$);

3 computes $h'\| Z' = d_2 \oplus h_5'$;

4 if $ID_B \neq 0$, computes $m'\|\alpha' = c \oplus H_4(h', ID_A, ID_B)$ (if $Z' = 0$, then $ID_A = 0$);

5 if $ID_B \neq 0$, computes $h_3' = H_3(m', \alpha', h', ID_A, PK_{A1}, PK_{A2}, ID_B)$(if $Z' = 0$, then $ID_A = 0$ and $PK_{A1}= PK_{A2}=0$);

6 if $Z' \neq 0$, then B accepts m' if and only if $h'= H_2(\alpha', m', ID_A, ID_B)$ and $e(Z', PK_{A2}+h_3'(q_A + s) P) =\alpha'$ holds. otherwise accepts m' if and only if $h'= H_2(\alpha', m', 0, ID_B)$.

5. Correctness and efficiency

5.2 Correctness

The correctness of our CLGSC scheme is below:

If $Z \neq 0$, then $e(Z, PK_{A2}+h_3(q_A + s)P)= e(r(x_A+h_3)^{-1} D_A,\ x_A(q_A + s) P+h_3(q_A + s) P) = e(r(x_A+h_3)^{-1}(q_A + s)^{-1} Q, (x_A+h_3)(q_A + s) P) = e(P, Q)^r =\alpha$.

5.3 Efficiency

Our scheme needs at most 2 multiplications in G_1 and 3 exponentiations in G_2 in the GSC algorithm, in UGSC algorithm, 2 pairing computation and 1 exponentiation in G_2 is needed.

6. Conclusion

In this paper, we first give the formal definition and security model of certificateless generalized signcryption. A concrete certificateless generalized signcryption scheme is also present based on bilinear pairing. Next we will give the security proof of our scheme.

7. References

[1] Y.Zheng, Digital signcryption or how to achieve cost (signature&encryption) <<cost (signature) + cost (encryption), Advances in CRYPTO' 97, LNCS 1294, Springer-Verlag, Berlin, 1997, pp. 165–179.

[2] Malone-Lee J., Identity based signcryption, in: Cryptology ePrint Archive. Report 2002/098.

[3] S.S.Al-Riyami, K.G. Paterson, Certificateless Public-key Cryptography, Advance in Cryptology ASIACRYPT 2003, LNCS2894, Springer-Verlag, 2003, pp.452-473.

[4] M.Barbosa, P.Farshim, Certifilateless Signcryption, Proceeding of the 2008 ACM Symposium on information, computer and communications security.

[5] S.Sharmila Deva, S.Sree Vivek, Deepanshu Shukla et al, Efficient and provably secure certificateless multi-receiver signcryption, Provable security, LNCS 5324, Springer, 2008, pp.52-67.

[6] S.Sharmila Deva, S.Sree Vivek, C.Pandu Rangan, A note on the certificateless multi-receiver signcryption scheme.

[7] Han Yiliang, Yang Xiaoyuan, New ECDSA– Verifable generalized signcryption, Chinese Journal of Computer, 2006(11), pp.2003-2012.

[8] X.Wang, X.Yang,Y.Han, Provable secure generalized signcryption, in: Cryptology ePrint Archive. Report 2007/173.

[9] Lal S, Kushwah P, ID-based generalized signcryption, in: Cryptology ePrint Archive. Report 2008/084.