An Identity-Based Signcryption Scheme with Short Ciphertext from Pairings*

Huiyan Chen[1,2], Shuwang Lü[1,3], Zhenhua Liu[1], and Qing Chen[3]

[1] State Key Laboratory of Information Security Graduate School of Chinese Academy of Sciences , Beijing 100049, P.R. China
[2] Institute of Electronics, Chinese Academy of Sciences, Beijing 100080
[3] Peking Knowledge Security Engineering Center, Beijing 100083
chenhuiyan@mails.gucas.ac.cn

Abstract. In this paper, we give a new identity-based signcryption scheme based on pairings. It is secure against adaptive chosen ciphertext and identity attack in the random oracle with the Modified Bilinear Diffie-Hellman assumption [14]. It produces shorter ciphertext than any one of schemes [7],[14] for the same plaintext and adapts to the band-constrained scenario very well.

Keywords: Signcryption, pairings, identity-based cryptography.

1 Introduction

The two fundamental services of public key cryptography are encryption and signing. Encryption provides confidentiality. Digital signatures provide authentication and non-repudiation. Often when we use one of these two services, we would like to use also the other. In 1997, Zheng [1] proposed a novel cryptographic primitive which he called as signcryption. The idea behind signcryption is to simultaneously perform signature and encryption in a logically single step in order to obtain confidentiality, integrity, authentication and non-repudiation at lower computational cost than the traditional "signature then encryption"approach. In addition, this latter solution also expends the final ciphertext size. Several efficient signcryption schemes [2],[3],[4],[5],[6] have been proposed since 1997. Malone-Lee afterward extended the signcryption idea to identity-based cryptography and firstly presented an identity-based signcryption scheme [8]. Indeed, the concept of identity-based cryptography was proposed in 1984 by Shamir [16]. The idea behind identity-based cryptography is that the user's public key can be derived from arbitrary string (e-mail address, IP address combined to a user name,...) which identifies him in a non ambiguous way. This greatly reduces the problems with key management. This kind of system needs trusted authority called private key generator(PKG) whose task is to compute user's private key from user's identity information. Several identity-based signcryption schemes have been proposed so far, e.g. [7],[9],[10],[11],[12],[13],[14].

* This work is supported by the National Natural Science Foundation of China (No. 60577039).

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 342–351, 2006.

Unfortunately, most of these schemes only operate on plaintexts of less than or equal to some fixed length.

In some situations, e.g. bandwidth-constrained scenario, it is desirable to shorten the length of ciphertext. In this paper we propose a new identity-based signcryption scheme which can deal with plaintexts of arbitrary length. For the same plaintext, it produces shorter ciphertext than any one of the schemes [7],[14] and adapts to the bandwidth-constrained scenario very well.

The paper will proceed as follow. In section 2, we review some preliminaries used throughout this paper. Our scheme is presented in Section 3. In Section 4, we compare our scheme with others. Section 5 concludes the paper.

2 Preliminaries

2.1 Notations

Throughout this paper, we will use the following notations. $|q|$ denotes the length of q in bits. If $|q| = 0$, q is denoted as ϕ. Z^+ denotes the set of natural numbers and $\{0,1\}^*$ denotes the the space of finite binary strings. Let $[m]^{l_1}$ denote the most significant l_1 bits of m and $[m]_{l_2}$ denote the least significant l_2 bits of m. We denote by $a||b$ the string which is the concatenation of strings a and b. We also denote $[x]= y$ if $y \le x < y+1$ and $y \in Z^+$. $a \bigoplus b$ denotes the bitwise XOR of bit strings a and b. If G is a set, $x \in_R G$ denotes that x is an element randomly selected from G. $Z_q = \{0, 1, \ldots, q-1\}$

2.2 Bilinear Map and Some Problems

Let G_1 be a cyclic additive group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group with the same order q. The bilinear map is given as $e : G_1 \times G_1 \rightarrow G_2$, which satisfies the following properties:

(1) Bilinearity: $\widehat{e}(aP, bQ) = \widehat{e}(P,Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in Z_q$
(2) Non-degeneracy: There exists $P, Q \in G_1$ such that $\widehat{e}(P,Q) \neq 1$, in other words, the map does not send all pairs in $G_1 \times G_1$ to the identity in G_2;
(3) Computability: There is an efficient algorithm to compute $\widehat{e}(P,Q)$ for all $P, Q \in G_1$.

We note that the weil and Tate pairings associated with supersingular elliptic curves can be modified to create such bilinear maps.

Definition 1. *Let l be a security parameter. Given two groups G_1 and G_2 of the same prime order q ($|q|$=l), a bilinear map $\widehat{e} : G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1, the Decisional Bilinear Diffie-Hellman Problem (DBDHP) in $(G_1, G_2, \widehat{e})$ is, given (P, aP, bP, cP, h) for unknown $a, b, c \in Z_q$, to decide whether $h = \widehat{e}(P,P)^{abc}$. The Modified Decisional Bilinear Diffie-Hellman Problem (MDB-DHP) in $(G_1, G_2, \widehat{e})$ is, given $(P, aP, bP, cP, c^{-1}P, h)$ for unknon $a, b, c \in Z_q$, to decide whether $h = \widehat{e}(P,P)^{abc^{-1}}$.*

We define the advantage of a distinguisher $\mathcal{D}$ against MDBDHP like this: $Adv_{\mathcal{D}}^{MDBDHP(G_1,G_2,P)}(l)$=$|Pr_{a,b,c\in_R Z_q}[1\leftarrow \mathcal{D}(aP, bP, cP, c^{-1}P, \widehat{e}(P, P)^{abc^{-1}})]$ $-Pr_{a,b,c\in_R Z_q, h\in_R G_2}[1\leftarrow\mathcal{D}(aP, bP, cP, c^{-1}P, h)]|$.

Obviously DBDHP is harder than MDBDHP. However, no known existing efficient algorithm can solve MDBDHP, to the best of our knowledge.

2.3 Framework of Identity-Based Signcryption Scheme

Signcryption schemes are made of five algorithms: *Setup*, *Keygen*, *Signcrypt*, *Unsigncryp* and *TPVerify*(if public verifiability is satisfied).
–*Setup:* Given a security parameter l, the private key generator(PKG)generates the system's public parameters *params*.
–*Keygen:* Given an identity ID, the PKG computes the corresponding private keys s_{ID}, d_{ID} and transmits them to their owner in a secure way.
–*Signcrypt:* To send a message m to Bob, Alice computes *Signcrypt*(m, s_{ID_A}, ID_B) to obtain the ciphertext σ.
–*Unsigncrypt:* When Bob receives σ, he computes *Unsigncrypt*(σ, ID_A, d_{ID_B}) and outputs the clear text m and ephemeral data *temp* for public verifiability, or the symbol $\perp$ if σ was an invalid ciphertext between identities ID_A and ID_B.
–*TPVerify:* On input $(\sigma, ID_A, m, temp)$, it outputs $\top$ for true or $\perp$ for false, depending on whether σ is a valid ciphertext of message m signcrypted by ID_A or not .

For obvious consistency purposes, we of course require that if σ=*Signcrypt*(m, s_{ID_A}, ID_B), then we have the relation $(m, temp)$ = *Unsigncrypt*(σ, ID_A, d_{ID_B}) and $\top$=*TPverify*$(\sigma, ID_A, m, temp)$.

2.4 Security Notions

Malone-Lee [8] extended notions of sematic security for public key encryption to identity-based signcryption schemes(IBSC). Sherman et al. slightly modified the definitions of these notions. these modified notions are indistinguishability against adaptive chosen ciphertext and identity attacks(IND-IBSC-CCIA) and existential unforgery of identity based signcryption under adaptive chosen message and identity attacks (EUF-IBSC-ACMIA). Now we recall the following definitions.

Definition 2. *An identity-based signcryption scheme has the IND-IBSC-CCIA property if no adversary has a non-negligible advantage in the following game.*

(1) *The challenger runs the* Setup *algorithm and sends the system parameters to the adversary*

(2) *The adversary $\mathcal{A}$ performs a polynomially bounded number of queries:*
– Signcrypt *query: $\mathcal{A}$ produces two identities ID_A, ID_B and a plaintext m. The challenger computes (s_{ID_A}, d_{ID_A}) =* Keygen(ID_A) *and then* Signcrypt(m, s_{ID_A}, ID_B) *and sends the result to $\mathcal{A}$.*
– Unsigncrypt *query: $\mathcal{A}$ produces two identities ID_A and ID_B , a ciphertext σ. The challenger generates the private key (s_{ID_B}, d_{ID_B}) =* Keygen(ID_B)

and sends the result of Unsigncrypt(σ, d_{ID_B}, ID_A) *to* $\mathcal{A}$ (*this result can be the* $\perp$ *symbol if* σ *is an invalid ciphertext*).

– Keygen *query:* $\mathcal{A}$ *produces an identity* ID *and receives the extracted private key* (s_{ID}, d_{ID}) = Keygen(ID).

$\mathcal{A}$ *can present its queries adaptively: every query may depend on the answer to the previous ones.*

(3) $\mathcal{A}$ *chooses two plaintexts* $m_0, m_1(|m_0| = |m_1|)$ *and two identities* ID_A *and* ID_B *on which he wishes to be challenged. He cannot have asked the private key corresponding to* ID_B *in the first stage.*

(4) *The challenger randomly takes a bit* $d \in \{0, 1\}$ *and computes* σ=Signcrypt (m_d, s_{ID_A}, ID_B) *which is sent to* $\mathcal{A}$.

(5) $\mathcal{A}$ *asks again a polynomially bounded number of queries just like in the first stage. This time, he cannot make a* Keygen *query on* ID_B *and he cannot ask the plaintext corresponding to* σ.

(6) *Finally,* $\mathcal{A}$ *produces a bit* $d^{'}$ *and wins the game if* $d^{'} = d$.

The adversary $\mathcal{A}$*'s advantage is defined to be* $Adv(A)$:=$|2Pr[d^{'} = d] - 1|$

Definition 3. *An identity-based signcryption scheme is said to have the EUF-IBSC-ACMIA property if no adversary has a non-negligeable advantage in the following game.*

(1) *The challenger runs the setup algorithm and gives the system parameters to the adversary* $\mathcal{A}$.

(2) *The adversary* $\mathcal{A}$ *performs a polynomially bounded number of queries just like in the previous definition 2.*

(3) *Finally,* $\mathcal{A}$ *produces a new triple* (σ^*, ID_A, ID_B) (*i.e. a triple that was not produced by the signcryption oracle*)*, where the private key of* ID_A *was not asked in the first stage and wins the game if the result of* Unsigncrypt(σ, ID_A, d_{ID_B}) *is not the* $\perp$ *symbol.*

The adversary's advantage is simply its probability of victory.

In this definition, to obtain the non-repudiation property and to prevent a dishonest recipient to send a ciphertext to himself on Alice's behalf and to try to convince a third party that Alice was the sender, it is necessary for the adversary to be allowed to make a *Keygen* query on the forged message's recipient ID_B.

3 Proposed Signcryption Scheme

3.1 Description of the Scheme

- *Setup:* Given a security parameter $l \in Z^+$, the private key generator(PKG) chooses groups G_1 and G_2 of prime order q($l = |q| = l_1 + l_2$, here $l_1 = [\frac{l+1}{2}], l_2 = [\frac{l}{2}]$), a generator P of G_1, an bilinear map $\widehat{e} : G_1 \times G_1 \rightarrow G_2$ and cryptographic hash functions H_1:$\{0, 1\}^* \rightarrow G_1$, H_2:$G_2 \rightarrow \{0, 1\}^n$ (here n is the key length of symmetric cipher), H_3:$\{0, 1\}^* \rightarrow Z_q^*$, F_1:$\{0, 1\}^{l_2} \rightarrow \{0, 1\}^{l_1}$, F_2:$\{0, 1\}^{l_1} \rightarrow \{0, 1\}^{l_2}$. It also choose a secure symmetric cipher $(\mathcal{E}, D)$ and a master-key $s \in Z_q^*$, and computes $P_{pub} = sP$ and g=$\widehat{e}(P$,$P_{pub})$. The system's public parameters are $\mathcal{P}$= $\{q, G_1, G_2, n, \widehat{e}, P, P_{pub}, g, H_1, H_2, H_3, F_1, F_2, \mathcal{E}, \mathcal{D}\}$.

- *Keygen:* Given identity ID, the PKG computes $Q_{ID} = H_1(ID)$ and the private key $d_{ID} = s^{-1}Q_{ID}$, $s_{ID} = sQ_{ID}$.
- *Signcrypt:* To send a message m ($|\mathcal{E}_{(\cdot)}(m)| \geq l_2$) to Bob, Alice follows the steps below.
  1. Compute $Q_{ID_B} = H_1(ID_B) \in G_1$.
  2. Randomly choose $x \in Z_q^*$, compute $k_1{=}g^x$, $k{=}H_2(\widehat{e}(P{,}Q_{ID_B})^x)$.
  3. Compute $c = c_1||c_2 {=}\mathcal{E}_k(m)$, $f = F_1(c_2)||(F_2(F_1(c_2)) \bigoplus c_2)$. Here if $|c| = l_2$, $c_2 = c$; if $|c| > l_2$, $c_1{=}[c]^{|c|-l_2}$, $c_2{=}[c]_{l_2}$.
  4. Compute $r = H_3(k_1) + f$ and $r_0 = H_3(r||c_1)$.
  5. Compute $S = xP_{pub} - r_0 s_{ID_A}$.
  6. The ciphertext is $\sigma = (c_1, r, S)$.
- *Unsigncrypt:* When receiving $\sigma = (c_1, r, S)$, Bob follows the steps below.
  1. Compute $Q_{ID_A} = H_1(ID_A) \in G_1$ and $r_0 = H_3(r||c_1)$.
  2. Compute $k_1 = \widehat{e}(S, P)\widehat{e}(Q_{ID_A}, P_{pub})^{r_0}$.
  3. Compute $\tau = \widehat{e}(S, d_{ID_B})\widehat{e}(Q_{ID_A}, Q_{ID_B})^{r_0}$ and $k = H_2(\tau)$.
  4. Compute $f = r - H_3(k_1)$.
  5. Compute $c_2 = [f]_{l_2} \bigoplus F_2([f]^{l_1})$ and $m {=}\mathcal{D}_k(c_2)$.
  6. Accept σ if and only if $[f]^{l_1} = F_1(c_2)$.
  7. Given (k, m, σ) to third party.
- *TPVerify:* On receiving (k, m, σ), the third party follows the steps below.
  1. Compute $r_0 = H_3(r||c_1)$ and $k_1 = \widehat{e}(S, P)\widehat{e}(Q_{ID_A}, P_{pub})^{r_0}$.
  2. Compute $f = r - H_3(k_1)$.
  3. Compute $c_2 = [f]_{l_2} \bigoplus F_2([f]^{l_1})$.
  4. Accept the origin of ciphertext if and only if $[f]^{l_1} = F_1(c_2)$.
  5. Accept the message authenticity if and only if $m {=}\mathcal{D}_k(c_2)$.
- **Remark 1:** If $|\mathcal{E}_{(\cdot)}(m)| < l_2$, we need some redundancy to signcrypt message m. For example, we choose a hash function $H : \{0,1\}^* \to \{0,1\}^{l_2}$ and set $c^{'} {=}\mathcal{E}_{(\cdot)}(m)||H(\mathcal{E}_{(\cdot)}(m))$, then we sign message $c^{'}$ by Fangguo Zhang et al's identity-based signature scheme. Since the length of paper is limited, we don't discuss it any more here. Throughout this paper, we assume $|\mathcal{E}_{(\cdot)}(m)| \geq l_2$ if message m need to be signcrypted.
- **Remark 2:** In the unsigncryption process, $f \in Z^+$ is turned into a bit string f. If $|f| < l$, we will fill $(l - |f|)$ zeros in the left of bit string f.

The consistency of this scheme is easy to verify by the bilinear pairing. It is forward-secure, in the sense that only Bob (and PKG) can recover m: knowledge of Alice's private keys s_{ID_A} and d_{ID_A} is insufficient to compute k. It is also publicly verifiable because, when verifying the messages origin by *TPVerify* algorithm, any third party does not depend on any private information. In order to convince someone that Alice is the sender of plaintext m, the receiver just have to forward the ephemeral decryption k to the third party.

3.2 Security Result

Theorem 1. *In the random oracle model (the hash functions are modeled as random oracles), if there is an IND-IBSC-CCIA adversary $\mathcal{A}$ that succeeds with an* *advantage ϵ when running in a time t and asking at most q_{H_1} H_1 queries, at most q_E Keygen queries, at most q_R H_3 queries, q_R Signcrypt queries and q_U Unsigncrypt queries, then there is a distinguisher $\mathcal{B}$ that can solve the MDBDH problem in $O(t+((6q_R+2)q_R+4q_U)T_{\hat{e}}+((3q_R+1)q_R+2q_U)T_{pm})$ time with an advantage*

$$Adv_{\mathcal{B}}^{MDBDHP(G_1,G_2,P)}(l) > (\epsilon(2^{[l/2]}-q_U)-q_U)/(q_{H_1})^2 2^{[l/2]+1}$$

where $T_{\hat{e}}$ denotes the computation time of the bilinear pairing, T_{pm} denote the computation time of exponentiation over G_2

Proof. see the appendix.

The existential unforgeability against adaptive chosen messages and identity attacks derives from the security of Fangguo Zhang ea al's identity-based signature scheme [15]. By arguments similar to those in [17], one can show that an attacker that is able to forge a signcrypted message must be able to forge a signature for Fangguo Zhang ea al's identity-based signature scheme.

4 Comparison of Schemes

Among these schemes [7],[9], [10],[11],[12],[13],[14], only schemes [7],[14] use the more general symmetric cipher and seems to process messages of arbitrary length. So, in table 1 below, we compare our scheme with schemes [7],[14] in terms of the length of the ciphertext which they produce and the number of the dominant operations required by them. In table we use mls, exps, and pcs as abbreviations for point multiplications in G_1, exponentiations in G_2 and pairing computations respectively. In table, we denote all the ciphertexts, which are produced by encrypting the plaintext m with symmetric cipher in different and equal length keys and which are of equal length, as c for convenience, since we only consider the ciphertext length not the content of the ciphertext.

Table 1. Comparison of Schemes

Schemes	Ciphertext Size		Efficiency					
	$\lvert c\rvert^*=l_2$	$\lvert c\rvert^*>l_2$	Signcrypt			Signcrypt		
		$(c_1=[c]^{\lvert c\rvert-l_2})$	mls	exps	pcs	mls	exps	pcs
Libert-Quisquater[10]♣	$\lvert c\rvert+\lvert q\rvert+\lvert G_1\rvert$	$\lvert c\rvert+\lvert q\rvert+\lvert G_1\rvert$	1	2	$2^{‡}$		2	$4^{‡}$
Chow-Yiu-Hui-Chow[15]	$\lvert c\rvert+\lvert q\rvert+\lvert G_1\rvert$	$\lvert c\rvert+\lvert q\rvert+\lvert G_1\rvert$	1	2	$2^{‡}$		2	$4^{‡}$
Our scheme	$\lvert q\rvert+\lvert G_1\rvert$	$\lvert c_1\rvert+\lvert q\rvert+\lvert G_1\rvert$	1	2	$1^{†}$		2	$4^{‡}$

(∗) c is produced by encrypting plaintext m with symmetric cipher.
(†) One pairing is precomputable
(‡) Two pairings are precomputable
(♣) This scheme has no forward-secure property

5 Conclusion

We proposed a new identity-based signcryption scheme. It produces shorter length ciphertext than any one of schemes [7],[14] for the same plaintext. It has the IND-IBSC-CCIA property in random oracle with assumption that MDBDHP is hard to decide. Additionally, it is an interesting problem to construct an identity-based signcryption schemes which produces shorter length ciphertext than ours for the same plaintext.

References

1. Y. Zheng, Digital Signcryption or How to Achieve Cost (Signature & Encryption)$\ll$ Cost (Signature)+ Cost (Encryption), Advances in Cryptology - Crypto'97, LNCS 1294, Springer, pp.165-179, 1997.
2. Y. Zheng, Identification, Signature and Signcryption using High Order Residues Modulo an RSA Composite, Proc. of PKC'01, LNCS 1992, Springer, pp. 48-63, 2001.
3. Y. Zheng, Signcryption and its applications in efficient public key solutions, Proc. of ISW'97,pp. 291-312, 1998.
4. Y.Zheng, H. Imai, Efficient Signcryption Schemes On Elliptic Curves, Proc. of IFIP/SEC'98, Chapman & Hall, 1998.
5. R. Steinfeld, Y. Zheng, A Signcryption Scheme Based on Integer Factorization, Proc. of ISW'00,pp. 308-322, 2000.
6. B.H. Yum, P.J. Lee, New Signcryption Schemes Based on KCDSA, Proc. of ICISC'01, LNCS2288, Springer, pp. 305-317, 2001.
7. B. Libert and J.-J. Quisquater. New identity based signcryption schemes based on pairings. In IEEE Information Theory Workshop, Paris, France, 2003.
8. J. Malone-Lee, Identity Based Signcryption, available at http://eprint.iacr.org/2002/098/.
9. D. Nalla and K. C. Reddy. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2002. http://eprint.iacr.org/2003/066.
10. R. Sakai and M. Kasahara. Id based cryptosystems with pairing on elliptic curve. In 2003 Symposium on Cryptography and Information Security - SCIS'2003, Hamamatsu, Japan, 2003. See also http://eprint.iacr.org/2003/054.
11. X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In Advances in Cryptology - Crypto'2003, volume 2729 of Lecture Notes in Computer Science, pages 383-399. Springer-Verlag, 2003.
12. L. Chen and J.Malone-Lee. Improved identity-based signcryption. Cryptology ePrint Archive, Report 2004/114, 2004. http://eprint.iacr.org/2003/114.
13. Noel McCullagh and Paulo S.L.M Barreto Efficient and Forward-Secure Identity-Based Signcryption , available at http://eprint.iacr.org/2004/117.
14. Sherman S.M. Chow, S.M. Yiu, Lucas C.K. Hui, and K.P. Chow. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. In: Information Security and Cryptology - ICISC 2003, LNCS 2971, pp. 352-369. Springer-Verlag, 2004
15. Fangguo Zhang, Willy Susilo, and Yi Mu. Identity-based Partial Message Recovery Signatures (or How to Shorten ID-based Signatures). Financial Cryptography and Data Security (FC'05), Lecture Notes in Computer Science, Springer Verlag, 2005, pp47-59
16. A. Shamir, identity-based Cryptosystems and Signature Schemes, Advances in Cryptology -Crypto' 84, LNCS 0196, Springer, 1984.
17. G. Gamage, J. Leiwo, Y. Zheng. Encrypted message authentication by Firewalls, Proc. of PKC'99, LNCS 1560, Springer, pp. 69-81, 1999.

Appendix: Proof of Theorem 1

The distinguisher $\mathcal{B}$ receives a random instance $(P, a_1P, a_2P, a_3P, a_3^{-1}P, h)$ of the MDBDH problem. Its goal is to decide whether $h = \widehat{e}(P,P)^{a_1a_2a_3^{-1}}$ or not. $\mathcal{B}$ will run $\mathcal{A}$ as a subroutine and act as $\mathcal{A}$'s challenger in the IND-IBSC-CCIA game. Here note that we only discuss the case $\mathcal{E}_{(.)}(m) = l_2$, the discussion of the case $\mathcal{E}_{(.)}(m) > l_2$ is similar to that of the case $\mathcal{E}_{(.)}(m) = l_2$ and is omitted. $\mathcal{B}$ needs to maintain lists L_1, L_2, L_3, L_4, and L_5 that are initially empty and are used to keep track of answers to queries asked by $\mathcal{A}$ to oracle H_1, H_2, H_3, F_1 and F_2. We assume that the following assumptions are made.

(1) $\mathcal{A}$ will ask for $H_1(ID)$ before ID is used in any *Signcrypt*, *Unsigncrypt* and *Keygen* queries.

(2) $\mathcal{A}$ will not ask for *Keygen*(ID) again if the query *Keygen*(ID) has been already issued before.

(3) Ciphertext returned from a *Signcrypt* query will not be used by $\mathcal{A}$ in an *Unsigncrypt* query.

At the beginning of the game, $\mathcal{B}$ gives $\mathcal{A}$ the system parameters with $P_{pub} = a_3P$ (a_3 is unknown to $\mathcal{B}$ and plays the role of the PKG's master key in the game).

– H_1 **queries:** When $\mathcal{A}$ makes an H_1 query on identity, $\mathcal{B}$ checks the list L_1, If an entry for the query is found, the same answer will be given to $\mathcal{A}$; otherwise, a value d_j from F_q^* will be randomly chosen and d_jP will be used as the answer, the query and the answer will then be stored in the list L_1. The only exception is that $\mathcal{B}$ has to randomly choose one of the H_1 queries from $\mathcal{A}$, say the i^{th} query, and answers $H_1(ID_i) = a_2P$ for this query. Since a_2P is a value in a random instance of the MDBDH problem, it does not affect the randomness of the hash function H_1.

– H_2**,** H_3**,** F_1 **and** F_2 **queries:** When $\mathcal{A}$ makes queries on these hash functions, $\mathcal{B}$ checks the corresponding list. If an entry for the query is found, the same answer will be given to $\mathcal{A}$; otherwise, a randomly generated value will be used as an answer to $\mathcal{A}$, the query and the answer will then be stored in the list.

– ***Keygen*** **queries:** When $\mathcal{A}$ asks a query *Keygen*(ID), if $ID = ID_i$, then $\mathcal{B}$ fails and stops. if $ID \neq ID_i$, then the list L_1 must contain a pair (ID, d) for some d. The private keys corresponding to ID is $d_{ID} = da_3^{-1}P$ and $s_{ID} = da_3P$ which $\mathcal{B}$ knows how to compute.

– ***Signcrypt*** **queries:** At any time $\mathcal{A}$ can perform a *Signcrypt* query for a plaintext m and identities ID_A, ID_B (Let ID_A, ID_B be the identity of the sender and that of the recipient respectively).

For case $ID_A \neq ID_i$, $\mathcal{B}$ can compute the private key s_{ID_A} correspondingly and the query can be answered by a call to *Signcrypt*(m, s_{ID_A}, Q_{ID_B}).

For the case $ID_A = ID_i$ and $ID_B \neq ID_i$, $\mathcal{B}$ answers *Signcrypt*(m, s_{ID_A}, Q_{ID_B}) query as follows. $\mathcal{B}$ randomly picks $S \in G_1^*$. Then $\mathcal{B}$ randomly choose $r_0 \in Z_q$ and computes $k_1 = (\widehat{e}(S, P)\widehat{e}(Q_{ID_A}, P_{pub})^{r_0})$. If L_3 contains $(k_1, \cdot)$, $\mathcal{B}$ has to repeat the same process using another r_0 until the corresponding $(k_1, \cdot)$ is not any entry in

L_3 (Note that: this process repeats at most $3q_R$ times as L_3 can contain at most $3q_R$. $\mathcal{B}$ needs to compute two pairings at most for each iteration of the process).

Then $\mathcal{B}$ computes $\tau= \hat{e}(S,\, d_{ID_B})\hat{e}(Q_{ID_A},\, Q_{ID_B})^{r_0}$, where d_{ID_B} is the private decryption key of ID_B. $\mathcal{B}$ finds $k = H_2(\tau)$ by running the simulation for H_2 and computes $c = \mathcal{E}_k(m)$. Then $\mathcal{B}$ finds $f_1 = F_1(c)$ by running the simulation for F_1 and $f_2 = F_2(f_1)$ by running the simulation for F_2, and computes $f{=}f_1||f_2 \bigoplus c$. Then $\mathcal{B}$ randomly picks $r_1 {\in} Z_q^* \backslash A$ (here $A{=}\{x{-}f|\ L_3$ contains $(x,\, \cdot)\} \bigcup \{x|L_3$ contains $(\cdot,\, x)\}\ \bigcup\{k_1 - f\}$) and puts $(k_1{,}r_1)$ and $(r_1{+}f,\ r_0)$ into L_3. The ciphertext $(r_1 + f, S)$ appears to be valid from $\mathcal{A}$'s viewpoint.

For case $ID_A = ID_B = ID_i$, $\mathcal{B}$ signcrypts m as follows. $\mathcal{B}$ randomly chooses $\tau^* \in G_2$ and $k^* \in \{0,1\}^n$ such that entries $(\tau^*, .)$ and $(., k^*)$ are not in L_2 and computes $c^* = \mathcal{E}_{k^*}(m)$. Then $\mathcal{B}$ finds $f_1^*{=}F_1(c^*)$ by running the simulation for F_1, and $f_2^* = F_2(f_1^*)$ by running the simulation for F_2 and computes $f^* = f_1^*||f_2^* \bigoplus c^*$. $\mathcal{B}$ randomly picks $S^* \in G_1^*$. Then $\mathcal{B}$ randomly choose $r_0^* \in Z_q$ and computes $k_1^*{=}(\hat{e}(S^*,\ P)\hat{e}(Q_{ID_A},\ P_{pub})^{r_0^*})$. If L_3 contains $(k_1^*,\ \cdot)$, $\mathcal{B}$ has to repeat the same process using another r_0^* until the corresponding $(k_1^*,\ \cdot)$ is not any entry in L_3 (Note that: $\mathcal{B}$ needs to compute two pairings at most for each iteration of the process). Then $\mathcal{B}$ randomly picks $r_1^* {\in} Z_q^* \backslash A$ (here $A{=}\{x{-}f^*|$ L_3 contains $(x,\ \cdot)\} \bigcup \{x|L_3$ contains $(\cdot,\ x)\} \bigcup \{k_1^* - f^*\})$ and puts $(k_1^*{,}r_1^*)$ and $(r_1^*{+}f^*,\ r_0^*)$ into L_3. $\mathcal{B}$ gives the ciphertext $\sigma^*{=}(r_1^*{+}f^*,\ S^*)$ to $\mathcal{A}$. As $\mathcal{A}$ will not ask for the unsigncryption of σ^*, he will never see that σ^* is not a valid ciphertext of the plaintext m where $ID_A = ID_B = ID_i$ (since τ^* may not equal to $\hat{e}(S^*, d_{ID_B})\hat{e}(Q_{ID_A}, Q_{ID_B})^{r_0^*}$).

– ***Unsigncrypt* queries:** When $\mathcal{A}$ makes a *Unsigncrypt* query for ciphertext $\sigma' = (r', S')$ from ID_A to ID_B, we consider the two cases below:

For the case $ID_B = ID_i$, $\mathcal{B}$ always answers $\mathcal{A}$ that σ' is invalid. So in the following case, $\mathcal{B}$ always notifies $\mathcal{A}$ the ciphertext is invalid: if the list L_3 contains (r', r'_0) and ($\hat{e}(S', P)\hat{e}(Q_{ID_A}, P_{pub})^{r'_0}$, y), the list L_5 contains an entry $([r' {-} y]^{l_1}$, $f'_2)$, and $\mathcal{A}$ previously asked the hash value $F_1([r' {-} y]_{l_2} \bigoplus f'_2)$, there is a probability of at most $1/2^{l_1}$ that $\mathcal{B}$ answered $[r' {-} y]^{l_1}$ (and that σ' was actually valid from $\mathcal{A}$'s point of view). The simulation fails if the list L_4 contains an entry $([r' {-} y]_{l_2} \bigoplus f'_2, [r' {-} y]^{l_1})$ (as $\mathcal{B}$ rejected a valid ciphertext).

For the case $ID_B \neq ID_i$, $\mathcal{B}$ rejects the ciphertext σ' if $(r', \cdot)$ isn't be found in L_3. Otherwise, it finds (r', r'_0) in L_3. $\mathcal{B}$ rejects the ciphertext σ' if $(k'_1, \cdot)$ is not be found in the list L_3 (here $k'_1 {=} \hat{e}(S', P)\hat{e}(Q_{ID_A}, P_{pub})^{r'_0}$). Otherwise, it finds $(k'_1$, $r'_1)$. $\mathcal{B}$ rejects the ciphertext σ' if $([r' {-} r'_1]^{l_1}, \cdot)$ isn't be found in L_5. Otherwise, it finds $([r' {-} r'_1]^{l_1}, f'_2\ )$ and computes $c' {=} [r' {-} r'_1]_{l_2} \bigoplus f'_2$. $\mathcal{B}$ rejects the ciphertext σ' if the list L_4 contains an entry (c', x) with $x {\neq} [r' {-} r'_1]^{l_1}$ or the list L_4 doesn't contain $(c'$, $[r' {-} r'_1]^{l_1})$. Otherwise, $\mathcal{B}$ computes $\tau' = \hat{e}(S', d_{ID_B})\hat{e}(Q_{ID_A}, Q_{ID_B})^{r'_0}$, then he searches for an entry $(\tau', \cdot)$ in list L_2; If no such entry is found, $\mathcal{B}$ randomly picks $k' \in \{0,1\}^n$ such that no entry with k' already exists in L_2 and inserts (τ', k') in L_2. $\mathcal{B}$ can use the corresponding k' to find $m' = \mathcal{D}_{k'}([r' {-} r'_1]_{l_2} \bigoplus f'_2)$ and returns m'. Apparently, under this case, the probability to reject at least one valid ciphertext doesn't exceed $\frac{q_U}{2^{\lceil l/2 \rceil}} (= Max\{\frac{q_U}{2^l}, \frac{q_U}{2^{\lceil (l+1)/2 \rceil}}, \frac{q_U}{2^{\lceil l/2 \rceil}}\})$.

By analyzing the two cases above, It is easy to see that, for all queries, the probability to reject at least one valid ciphertext does exceed $\frac{q_U}{2^{[l/2]}}$ ($=Max$ $\{\frac{q_U}{2^{[(l+1)/2]}}, \frac{q_U}{2^{[l/2]}}\}$).

After the first stage, $\mathcal{A}$ picks a pair of identities on which he wishes to be challenged. Note that $\mathcal{B}$ fails if $\mathcal{A}$ has asked an *Keygen* query on ID_i during the first stage. It is easy to see that the probability for $\mathcal{B}$ not to fail in this stage is greater than $\frac{1}{q_{H_1}}$. Further, with a probability exactly $\frac{2}{q_{H_1}}$ ($=(q_{H_1}-1)/\binom{q_{H_1}}{2}$), $\mathcal{A}$ chooses to be challenged on the pair (ID_j, ID_i) with $j \neq i$. Hence the probability that $\mathcal{A}$'s response is helpful to $\mathcal{B}$ is greater than $\frac{1}{(q_{H_1})^2}$. Note that if $\mathcal{A}$ has submitted an *Keygen* query on ID_i, then $\mathcal{B}$ fails because he is unable to answer the question. On the other hand, if $\mathcal{A}$ does not choose (ID_j, ID_i) as target identities, $\mathcal{B}$ fails too.

Then $\mathcal{A}$ produces two plaintexts m_0 and m_1 ($|m_0|$=$|m_1|$), $\mathcal{B}$ randomly picks a bit $b \in \{0,1\}$ and signcrypts m_b. To do so, he sets $S^* = a_1P$ and randomly chooses $r_0^* \in F_q^*$. Suppose $ID_j = dP$, setting $S^* = a_1P$ implies $(x - r_0^*)da_3 = a_1$, i.e. $x = a_1a_3^{-1}d^{-1} + r_0^*$. Since a_1P and a_3P belong to a random instance of the MDBDH problem, x is random and this will not modify $\mathcal{A}$'s view. $\mathcal{B}$ computes k_1^*=$(\widehat{e}(S^*,\ P)\widehat{e}(Q_{ID_A},\ P_{pub})^{r_0^*})$. If L_3 contains $(k_1^*,\ \cdot)$, $\mathcal{B}$ has to repeat the same process using another r_0^* until the corresponding $(k_1^*,\ \cdot)$ is not any entry in L_3. $\mathcal{B}$ computes τ^*=$h\widehat{e}(Q_{ID_j},\ a_2P)^{r_0^*}$, where h is $\mathcal{B}$'s candidate for the MDBDH problem, obtains $k^* = H_2(\tau^*)$ by running the simulation for H_2, and computes c_b =$\mathcal{E}_{k^*}(m_b)$. Then $\mathcal{B}$ finds f_1 =$F_1(c_b)$ by running the simulation for F_1 and f_2 =$F_2(f_1)$ by running the simulation for F_2, and compute $f = f_1||f_2 \bigoplus c_b$. Then $\mathcal{B}$ randomly picks $r_1^* \in Z_q^* \backslash A$ (here A=$\{x-f|\ L_3$ contains $(x,\ \cdot)\}\bigcup\{x|L_3$ contains $(\cdot,\ x)\}\bigcup\{k_1^* - f\}$) and puts (k_1^*, r_1^*) and $(r_1^*+f,\ r_0^*)$ into L_3. $\mathcal{B}$ sends the ciphertext σ^*=$(r_1^*+f,\ S^*)$ to $\mathcal{A}$.

$\mathcal{A}$ then performs a second series of queries, $\mathcal{B}$ can handle these queries as in the first stage. At the end, $\mathcal{A}$ will produce a bit $b^{'}$ for which he believes relation $\sigma = Signcrypt(m_{b'}, s_{ID_j}, Q_{ID_i})$ holds . If $b = b^{'}$, $\mathcal{B}$ then answers 1 as the result to the MDBDH problem since he has produced a valid signcrypted message of m_b using the knowledge of h. Otherwise, $\mathcal{B}$ should answer 0.

Taking into account all the probabilities that $\mathcal{B}$ will not fail its simulation, the probability that $\mathcal{A}$ chooses to be challenged on the pair (ID_j, ID_i), and also the probability that $\mathcal{A}$ wins the IND-IBSC-CCIA game, we have

$$Adv_{\mathcal{B}}^{MDBDHP(G_1,G_2,P)}(l) > (\tfrac{\epsilon+1}{2}(1 - \tfrac{q_U}{2^{[l/2]}}) - 1/2)(1/(q_{H_1})^2) = \frac{\epsilon(2^{[l/2]}-q_U)-q_U}{(q_{H_1})^2 2^{[l/2]+1}}$$

Regarding the time complexity, it can be verified by counting the number of pairing operations required to answer all queries.