ID-based Encryption Scheme Secure against Chosen Ciphertext Attacks

Rongxing Lu and Zhenfu Cao

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200030, P. R. China
{cao-zf, rxlu}@cs.sjtu.edu.cn
http://tdt.sjtu.edu.cn

Abstract. ID-based encryption allows for a sender to encrypt a message to an identity without access to a public key certificate. Based on the bilinear pairing, Boneh and Franklin proposed the first practical ID-based encryption scheme and used the padding technique of Fujisaki-Okamto to extend it to be a chosen ciphertext secure version. In this letter, we would like to use another padding technique to propose a new ID-based encryption scheme secure against chosen ciphertext attacks. The security of our scheme is based on the Gap bilinear Diffie-Hellman assumption in the random oracle model.

Keywords: ID-based encryption, chosen ciphertext security, gap bilinear Diffie-Hellman problem.

1 Introduction

The concept of ID-based cryptosystem was first introduced by Shamir [1] in 1984. In such an ID-based cryptosystem, the public key of a user is derived from his identity information and his private key is generated by a trusted third party called Private Key Generator (PKG). The advantage of an ID-based cryptosystem is that it simplifies the key management process which is a heavy burden in the traditional certificate based cryptosystem. In an ID-based cryptosystem, if Alice wants to send an encrypted message to Bob, she only needs to use Bob's identity information as public key to encrypt the message.

In 2001, Boneh and Franklin [2] proposed the first full functional ID-based encryption scheme `BasicIdent` from bilinear pairings and applied the padding technique of Fujisaki-Okamoto [3] to extend `BasicIdent` to `FullIdent`, which is secure against chosen ciphertext attacks. However, the security reduction of `FullIdent` is far from tight.

In this letter, based on the Gap bilinear Diffie-Hellman problem, we would like to use another padding technique [4] to propose a new ID-based encryption scheme secure against chosen ciphertext attacks and use the technique from provable security [5] to analyze its security. The advantage of our proposed scheme is that the security reduction is quite tight.

2 Definitions

2.1 Notations

We let $\mathbb{N} = \{1, 2, 3, \ldots\}$ be the set of positive integers. If x is a string, then $|x|$ denotes its length, while if $\mathbb{S}$ is a set then $|\mathbb{S}|$ denotes its size and $s \xleftarrow{R} \mathbb{S}$ denotes the operation of picking a random element s of $\mathbb{S}$ uniformly. If m is a string, $[m]^{k_1}$ denotes the most significant k_1 bits of m and $[m]_{k_2}$ denotes the least significant k_2 bits of m. Besides, $\oplus$ denotes XOR operation and $\|$ denotes a concatenation throughout this letter.

2.2 Definition of ID-based Encryption Scheme

An ID-based encryption scheme $\mathcal{E}$ uses four algorithms: *Setup*, *Extract*, *Encrypt* and *Decrypt*. The functions of these algorithms are described as follows:

Setup: Given a security parameter k, it returns the system parameters `params` and the master key `master-key`. The `params` will be publicly known, while the `master-key` will be known only to PKG.

Extract: Given `params`, `master-key` and an arbitrary $id \in \{0, 1\}^*$, it returns a private key S_{id}. Here id will be used as the public key.

Encrypt: Given `params`, an identity id, and a message m, it returns a ciphertext c.

Decrypt: Given `params`, a private key S_{id}, and a ciphertext c, it returns a message m.

These algorithms must satisfy the standard consistency constraint of ID-based encryption, i.e. the private key S_{id} is generated by *Extract* when it is given id as the public key, then

$$\forall m \; \mathit{Decrypt}(\texttt{params}, c, S_{id}) = m; \quad \text{where } c = \mathit{Encrypt}(\texttt{params}, id, m).$$

2.3 Chosen Ciphertext Security

Boneh and Franklin [2] strengthened the IND-CCA model to deal with an adversary who possesses private keys corresponding to identities of its choice and attacks an identity id in an ID-based encryption scheme. They called it IND-ID-CCA model. The IND-ID-CCA model is described through the following game between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

SETUP: The challenger $\mathcal{C}$ takes a security parameter k and runs the *Setup* algorithm. It gives the adversary $\mathcal{A}$ the resulting system parameters `params`, and keeps the `master-key` itself.

PHASE 1: The adversary $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_m$ where query q_i is one of:

– Extraction query $< id_i >$. The challenger $\mathcal{C}$ responds by running *Extract* to generate the private key S_{id_i} corresponding to the pubic key $< id_i >$, and sends S_{id_i} back to $\mathcal{A}$.

– Decryption query $< id_i, c_i >$. The challenger $\mathcal{C}$ responds by running algorithm *Extract* to generate the private key S_{id_i} corresponding to id_i. It then runs algorithm *Decrypt* to decrypt the ciphertext c_i using the private key S_{id_i}, and returns the resulting plaintext to $\mathcal{A}$.

CHALLENGE: Once $\mathcal{A}$ decides that Phase 1 is over it outputs two equal length plaintexts m_0, m_1, an identity id on which it wishes to be challenged. The only constraint is that id did not appear in any private key extraction queries in Phase 1. The challenger $\mathcal{C}$ picks a random bit $b \in \{0,1\}$ and sets $c = \mathit{Encrypt}(\texttt{params}, id, m_b)$. It responds c to $\mathcal{A}$.

PHASE 2: $\mathcal{A}$ issues more queries $q_{m+1}, \ldots, q_n$ where q_i is one of:

– Extraction query $< id_i >$ where $id_i \neq id$. $\mathcal{C}$ responds as in Phase 1.
– Decryption query $< id_i, c_i > \neq < id, c >$. $\mathcal{C}$ responds as in Phase 1.

GUESS: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$.

We define $\mathcal{A}$'s advantage as the following function of security k, where k is given as input to $\mathcal{C}$:

$$Adv_{\mathcal{A}}(k) = \left|2\Pr[b = b'] - 1\right|.$$

We say that an ID-based encryption scheme is IND-ID-CCA secure, if no polynomially bounded adversary $\mathcal{A}$ has non-negligible advantage against the challenger $\mathcal{C}$.

3 Basic Concepts on Bilinear Pairings

Bilinear pairing is an important cryptographic primitive [2]. Let $\mathbb{G}_1$ be a cyclic additive group generated by P, whose order is a prime q, $|q| = k$, and $\mathbb{G}_2$ be a cyclic multiplicative group of the same order q. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties:

– *Bilinear*: For any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, we have $e(aP, bQ) = e(P,Q)^{ab}$.
– *Non-degenerate*: There exists $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_1$ such that $e(P,Q) \neq 1$.
– *Computable*: There is an efficient algorithm to compute $e(P,Q)$ for all $P, Q \in \mathbb{G}_1$.

We note that the Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such bilinear maps [2].

Next, we describe three mathematical problems in $\mathbb{G}_1$, $\mathbb{G}_2$, namely the Bilinear Diffie-Hellman (BDH) Problem, the Decisional Bilinear Diffie-Hellman (DBDH) Problem and the Gap Bilinear Diffie-Hellman (GBDH) Problem.

– BDH Problem: For $a, b, c \in \mathbb{Z}_q^*$, given $P, aP, bP, cP \in \mathbb{G}_1$, compute $e(P,P)^{abc} \in \mathbb{G}_2$.
– DBDH Problem: For $a, b, c \in \mathbb{Z}_q^*$, given P, aP, bP, $cP \in \mathbb{G}_1$ and $r \in G_2$, decide whether $r = e(P,P)^{abc}$ or not.

– GBDH Problem: For $a, b, c \in \mathbb{Z}_q^*$, given P, aP, bP, $cP \in \mathbb{G}_1$, compute $e(P,P)^{abc} \in \mathbb{G}_2$ with the help of a DBDH oracle (which answers whether a given tuple is a BDH tuple or not.)

We define by $Succ_{\mathbb{G}_1,\mathbb{G}_2}^{GBDH}(\mathcal{A})$ the success probability of an algorithm $\mathcal{A}$ in solving the GBDH Problem as

$$Succ_{\mathbb{G}_1,\mathbb{G}_2}^{GBDH}(\mathcal{A}) = \Pr[\mathcal{A}(P, aP, bP, cP) = e(P,P)^{abc}].$$

We say that the GBDH assumption holds if $Succ_{\mathbb{G}_1,\mathbb{G}_2}^{GBDH}(\mathcal{A})$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$.

4 Our Proposed Scheme

In this section, based on the formal definition in Section 2, we will introduce our ID-based encryption scheme from bilinear pairings.

– *Setup:* PKG chooses a random number $s \stackrel{R}{\longleftarrow} \mathbb{Z}_q^*$ and sets $P_{pub} = sP$, then defines three cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_1 : \mathbb{G}_2 \rightarrow \{0,1\}^{k_1+k_2}$ and $H_2 : \{0,1\}^{k_1} \times \mathbb{G}_2 \rightarrow \{0,1\}^{k_2}$, where k_1, k_2 are two security parameters. Finally, PKG publishes $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_{pub}, H_0, H_1, H_2\}$ and keeps s as the `master-key` secret.
– *Extract:* A user submits his identity id to PKG. PKG computes the user's public key as $Q_{id} = H_0(id)$, and returns $S_{id} = sQ_{id}$ to the user as his private key.
– *Encrypt:* To encrypt a message $m \in \{0,1\}^{k_1}$ for a user with the identity id do the followings:
  - Pick a random number $r \stackrel{R}{\longleftarrow} \mathbb{Z}_q^*$ and compute g^r, where $g = e(P_{pub}, Q_{id}) \in \mathbb{G}_2$.
  - Compute $c_1 = rP$ and $c_2 = H_1(g^r) \oplus (m \| H_2(m, g^r))$.
  - Output a ciphertext $c = (c_1, c_2)$.
– *Decrypt:* To decrypt the ciphertext $c = (c_1, c_2)$, the following steps will be run:
  - Use the private key S_{id} to compute g^r as follows,

$$e(c_1, S_{id}) = e(rP, sQ_{id}) = e(P_{pub}, Q_{id})^r = g^r$$

  - Compute $w = c_2 \oplus H_1(g^r)$;
  - Check whether $H_2([w]^{k_1}, g^r) = [w]_{k_2}$. If it holds, accept $c = (c_1, c_2)$ and define m as $[w]^{k_1}$ and output m. Otherwise, output "reject".

5 Security Analysis

In this section, based on GBDH assumption, we will show our proposed scheme is IND-ID-CCA secure.

Theorem 1. *Our proposed ID-based encryption scheme is secure in the sense of IND-ID-CCA in the random oracle model, providing that the GBDH Problem is intractable.*

Proof. Assume $\mathcal{A}$ be an IND-ID-CCA adversary, who can, with running time τ and advantage ϵ, break our proposed scheme after making at most q_0, q_1, q_2, q_e and q_d queries to the random oracles H_0, H_1, H_2, the extraction oracle and the decryption oracle, respectively. $\mathcal{A}$ is also allowed to access DBDH oracle $\mathcal{O}^{DBDH}$ to check whether a tuple is a BDH tuple or not. Then we can use $\mathcal{A}$ to construct another algorithm $\mathcal{C}$ to resolve the GBDH Problem with another probability ϵ' within time τ', where

$$\epsilon' \geq \epsilon - q_d \cdot (2^{-(k_1+k_2)} + 2^{-k_2})$$
$$\tau' \leq \tau + 2 \cdot (q_1 - 1) \cdot T_{pmul} + q_d \cdot T_{DBDH}$$

with T_{pmul} the time for point multiplication in $\mathbb{G}_1$ and T_{DBDH} the time for an $\mathcal{O}^{DBDH}$ operation.

Initially, $\mathcal{C}$ is given an instance (P, xP, yP, zP) of the GBDH Problem, and its goal is to compute $e(P,P)^{xyz} \in \mathbb{G}_2$. Then $\mathcal{C}$ runs $\mathcal{A}$ as a subroutine and simulates its attack environment.

Setup: $\mathcal{C}$ sets $P_{pub} = xP$ and gives pubic parameters $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_{pub}, H_0, H_1, H_2\}$ to $\mathcal{A}$. To illustrate our proof idea simply and clearly, we here assume the identity id of $\mathcal{A}$'s challenge is determined in advance. That is, $\mathcal{A}$ won't make decryption queries on other identity $id_i \neq id$ afterwards.

At the same time, without loss of generality, we assume all queries to the random oracles H_0, H_1, H_2, the extraction oracle and the decryption oracle are distinct, and the extraction query is preceded by an H_0 query. To avoid collision and consistently respond to these queries, $\mathcal{C}$ should maintain three lists Λ_{H_0}, Λ_{H_1} and Λ_{H_2}, which are initially empty. Note that, to resolve the GBDH Problem, $\mathcal{C}$ is allowed to query to the DBDH oracle $\mathcal{O}^{DBDH}$, when it processes decryption oracle query.

H_0-query: When $\mathcal{A}$ makes an H_0 query id_i, if $id_i = id$, then $\mathcal{C}$ returns $Q_{id} = H_0(id) = yP$. Otherwise, $\mathcal{C}$ chooses a random number $t_i \xleftarrow{R} \mathbb{Z}_q^*$, adds $< id_i, t_i, t_iP, t_ixP >$ to Λ_{H_0} and returns $Q_{id_i} = H_0(id_i) = t_iP$.

H_1-query: When $\mathcal{A}$ makes a new H_1 query g_i, $\mathcal{C}$ chooses a random number $h_{1i} \xleftarrow{R} \{0,1\}^{k_1+k_2}$, adds $< g_i, h_{1i} >$ to Λ_{H_1} and returns $H_1(g_i) = h_{1i}$.

H_2-query: When $\mathcal{A}$ makes a new H_2 query (m_i, g_i), $\mathcal{C}$ chooses a random number $h_{2i} \xleftarrow{R} \{0,1\}^{k_2}$, adds $< m_i, g_i, h_{2i} >$ to Λ_{H_2} and returns $H_2(m_i, g_i) = h_{2i}$.

Phase 1:

- Extraction query: When $\mathcal{A}$ asks an extraction query on $id_i \neq id$, $\mathcal{C}$ finds $< id_i, t_i, t_iP, t_ixP >$ in Λ_{H_0}. Then $\mathcal{C}$ returns t_ixP as the private key to $\mathcal{A}$.
- Decryption query: When $\mathcal{A}$ asks a decryption query on $(id, c_i = (c_{1i}, c_{2i}))$, $\mathcal{C}$ asks to the DBDH oracle $\mathcal{O}^{DBDH}$ to check whether a tuple $(Q_{id} = yP, P_{pub} = xP, c_{1i}, g_i)$ is a valid BDH tuple and then returns a right plaintext m_i to $\mathcal{A}$. More precisely, $\mathcal{C}$ does the followings:

- If there exists $< g_i, h_{1i} >$ in Λ_{H_1} such that $< Q_{id}, P_{pub}, c_{1i}, g_i >$ is a valid BDH tuple by asking $\mathcal{O}^{DBDH}$, $\mathcal{C}$ computes $w_i = c_{2i} \oplus h_{1i}$. Otherwise, $\mathcal{C}$ reports failure and terminates.
- If there exists $< m_i, g_i, h_{2i} >$ in Λ_{H_2} such that $m_i = [w_i]^{k_1}$ and $h_{2i} = [w_i]_{k_2}$, $\mathcal{C}$ outputs and returns m_i. Otherwise, $\mathcal{C}$ also reports failure and terminates.

Challenge: Once $\mathcal{A}$ decides the Phase 1 is over, he outputs id and two messages $m_0, m_1 \in \{0,1\}^{k_1}$ on which it wishes to be challenged. $\mathcal{C}$ will respond as follows:

- Set $c_1 = zP$, randomly choose $b \in \{0,1\}$ and select a random number $c_2 \xleftarrow{R} \{0,1\}^{k_1+k_2}$.
- Return $c = (c_1, c_2)$ as the ciphertext of m_b.

Phase 2:

- Extraction query: Same as in Phase 1.
- Decryption query: Same as in Phase 1, and the challenge $(id, c = (c_1, c_2))$ is excluded.

Guess: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

In an information theoretical sense, the adversary $\mathcal{A}$ cannot gain any advantage in distinguishing m_0, m_1 if it has not asked for $e(P,P)^{xyz}$ to H_1 or $(\star, e(P,P)^{xyz})$ to H_2. Therefore, we denote E_1 the event that $\mathcal{A}$ has asked $e(P,P)^{xyz}$ to H_1 and E_2 the event that $\mathcal{A}$ has asked $(\star, e(P,P)^{xyz})$ to H_2. Then,

$$
\begin{aligned}
& \Pr[b = b'] = \tfrac{1}{2} \pm \tfrac{1}{2} Adv_{\mathcal{A}}(k) \\
= \; & \Pr[b = b' \wedge (\mathrm{E}_1 \vee \mathrm{E}_2)] + \Pr[b = b' \wedge \neg(\mathrm{E}_1 \vee \mathrm{E}_2)] \\
= \; & \Pr[b = b' \wedge (\mathrm{E}_1 \vee \mathrm{E}_2)] + \tfrac{1}{2} \Pr[\neg(\mathrm{E}_1 \vee \mathrm{E}_2)] \\
= \; & \Pr[b = b' \wedge (\mathrm{E}_1 \vee \mathrm{E}_2)] + \tfrac{1}{2} - \tfrac{1}{2} \Pr[(\mathrm{E}_1 \vee \mathrm{E}_2)] \\
\Rightarrow \; & \pm Adv_{\mathcal{A}}(k) = \Pr[(\mathrm{E}_1 \vee \mathrm{E}_2)] - 2\Pr[b = b' \wedge (\mathrm{E}_1 \vee \mathrm{E}_2)] \\
\Rightarrow \; & \Pr[(\mathrm{E}_1 \vee \mathrm{E}_2)] \geq Adv_{\mathcal{A}}(k)
\end{aligned}
$$

During the decryption query, some decryptions may be incorrect, but only rejecting a valid ciphertext: a ciphertext is refused if the query g_i has not been asked to H_1 or (m_i, g_i) has not been asked to H_2. However, the adversary $\mathcal{A}$ might have guessed the right values for $H_1(g_i)$ and $H_2(m_i, g_i)$ without having asked for them, but only with probability $2^{-(k_1+k_2)} + 2^{-k_2}$.

Thus, by checking Λ_{H_1} and Λ_{H_2}, we can obtain the solution $e(P,P)^{xyz}$. Then,

$$
\begin{aligned}
& \Pr[(\mathrm{E}_1 \vee \mathrm{E}_2) \wedge \text{no incorrect decryption}] \\
& \geq Adv_{\mathcal{A}}(k) - q_d \cdot (2^{-(k_1+k_2)} + 2^{-k_2}) \\
\Rightarrow \; & \epsilon' \geq \epsilon - q_d \cdot (2^{-(k_1+k_2)} + 2^{-k_2})
\end{aligned}
$$

Here, if we only consider the time-consuming operations, namely the point multiplication operation and the $\mathcal{O}^{DBDH}$ operation, and neglect other operations, then the total running time of $\mathcal{C}$ in resolving the GBDH Problem is bounded by

$$\tau' \leq \tau + 2 \cdot (q_1 - 1) \cdot T_{pmul} + q_d \cdot T_{DBDH}.$$

Thus, the proof is completed.

From the above Theorem 1, it is clear that the security reduction of our proposed scheme is quite tight.

6 Conclusion

In this letter, based on GBDH Problem, we have proposed a new ID-based encryption scheme and used the techniques from provable security to analyze the security of our proposed scheme. By analysis, our proposed scheme is secure against chosen ciphertext attacks with tight reduction.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments.

References

1. A. Shamir, "Identity-based cryptosystems and signature schemes", In *Advances in Cryptology - Crypto'84*, LNCS 196, pp. 47 - 53, Springer-Verlag, 1984.
2. D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", In *Advances in Cryptology - Crypto'01*, LNCS 2139, pp. 213 - 229, Springer-Verlag, 2001.
3. E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes", In *Advances in Cryptology - Crypto'99*, LNCS 1666, pp. 537 - 554, Springer-Verlag, 1999.
4. Y. Zheng, "Improved public key cryptosystems secure agaisnt chosen ciphertext attacks", *Technical Note, University of Wollongong*, 1994.
5. T. Okamoto and D. Pointcheval, "The gap-problems: a new class of problems for the security of cryptographic schemes", In *Public key Cryptograhy - PKC'01*, LNCS 1992, pp. 104 - 118, Springer-Verlag, 2001.