Applying Fujisaki-Okamoto to Identity-Based Encryption

Peng Yang[1], Takashi Kitagawa[2], Goichiro Hanaoka[2], Rui Zhang[1], Kanta Matsuura[1], and Hideki Imai[1,2]

[1] Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
{pengyang, kanta, imai}@iis.u-tokyo.ac.jp, zhang@imailab.iis.u-tokyo.ac.jp
[2] Research Centre for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), 1102 Akihabara-Daibiru, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
{t-kitagawa, hanaoka-goichiro}@aist.go.jp

Abstract. The Fujisaki-Okamoto (FO) conversion is widely known to be able to generically convert a weak public key encryption scheme, say one-way against chosen plaintext attacks (OW-CPA), to a strong one, namely, indistinguishable against adaptive chosen ciphertext attacks (IND-CCA). It is not known that if the same holds for identity-based encryption (IBE) schemes, though many IBE and variant schemes are in fact specifically using the FO conversion. In this paper, we investigate this issue and confirm that the FO conversion is generically effective also in the IBE case. However, straightforward application of the FO conversion only leads to an IBE scheme with a loose (but polynomial) reduction. We then propose a simple modification to the FO conversion, which results in considerably more efficient security reduction.

1 Introduction

Background. Identity based encryption (IBE) [11] is a public key encryption scheme where the encryption public key can be an arbitrarily string, such as the recipient's identity, thus the distribution of public key certificates can be avoided for an IBE scheme. This was first motivated by applications to encrypt emails under the recipient's email address, however, it found more applications ever since, e.g. [8, 4].

It has been shown [1, 7] that the strongest security notion for IBE is *indistinguishability against adaptive chosen ID and adaptive chosen ciphertext attacks* (IND-ID-CCA). Nevertheless, many IBE schemes, other than (IND-ID-CCA), first build a "basic scheme" which is *one-way against adaptive chosen ID and chosen plaintext attacks* (OW-ID-CPA), then *specifically* use the famous Fujisaki-Okamoto (FO) conversion [6] to upgrade the basic scheme to a scheme with IND-ID-CCA security. However, it is still unknown whether the FO conversion can *generically* upgrade OW-ID-CPA security to IND-ID-CCA security.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 183–192, 2006.

It is crucial to note that the FO conversion is a generic conversion to enhance a public key encryption scheme with security of *one-wayness under chosen plaintext attacks* (OW-CPA) to security of *indistinguishability against adaptive chosen ciphertext attacks* (IND-CCA) [10] in the random oracle model. Many practical PKE schemes are based on it.

Our Contributions. Our contributions are three-fold:
First, we investigate the generic security of the IBE obtained by applying the FO conversion to an underlying OW-ID-CPA secure IBE and confirm the IND-ID-CCA security of the IBE can be polynomially reduced to the OW-ID-CPA security of the underlying IBE.

Additionally, we find that the straightforward application of the FO conversion yields a significantly inefficient reduction cost. To be more precise, the simulator's time complexity is more than $2^{100}(> 2^{80})$ times re-encryption computation (in addition to an IND-ID-CCA adversary's running time).

Finally, we slightly modify the FO conversion so that the simulator's time complexity is reduced to be $2^{60}(< 2^{80})$ times re-encryption computation (in addition to an adversary's running time) which can be dealt with in practice.

2 Preliminary

In this section, we present the definitions of IBE, OW-ID-CPA, IND-ID-CCA and γ-uniformity.

ID-Based Encryption. ID-Based encryption (IBE) [11] is a public key encryption scheme where the encryption public keys can be arbitrary strings. It is formally defined as follows:

Definition 1 (ID-Based Encryption). *Formally, an identity-based encryption (IBE) scheme* $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ *consists of the four algorithms.*

- $\mathcal{S}$, the setup algorithm, takes security parameter $k \in \mathbf{Z}$ as input, and outputs system parameters params and the master-key master-key. $\mathcal{S}$ is a probabilistic algorithm. params consists of descriptions of a finite message space MSPC, and a finite ciphertext space CSPC.
- $\mathcal{X}$, the extract algorithm, takes as inputs params, master-key and an arbitrary $\mathsf{ID} \in \{0,1\}^*$, and outputs a private key d. ID is an arbitrary string and used as a public key. d is the corresponding private key(decryption key). This algorithm extracts a private key corresponding to ID.
- $\mathcal{E}$, the encryption algorithm, takes as input params, ID and $M \in$ MSPC. Let $\mathsf{COIN}(k) \subseteq \{0,1\}^*$ be a finite set. $\mathcal{E}$ chooses a random string $coin \in \mathsf{COIN}(k)$ and outputs a ciphertext $C \in$ CSPC. $\mathcal{E}$ is a probabilistic algorithm. We denote the result of running this algorithm $\mathcal{E}(\mathsf{params}, \mathsf{ID}, M; coin)$.
- $\mathcal{D}$, the decryption algorithm, takes as input params, $C \in$ CSPC and a private key d, and outputs $M \in$ MSPC. The algorithm decrypts a ciphertext C using the private key d.

These algorithms must satisfy the standard consistency constraint,

$$\forall M \in \mathsf{MSPC},\ \mathcal{D}(\mathsf{params}, d, C) = M \text{ where } C = \mathcal{E}(\mathsf{params}, \mathsf{ID}, M).$$

One-Way Identity-Based Encryption. A notion of security called one-way encryption(OWE) is an even weaker notion. Roughly speaking, this notion means that when given the encryption of a random plaintext the adversary cannot produce the plaintext in its entirety. Originally OWE is defined for standard public key encryption schemes. Boneh and Franklin [3] extended the definition of OWE for IBE schemes. An IBE scheme is an one-way encryption scheme if no polynomial adversary $\mathcal{A}$ has a non-negligible advantage against the challenger in the following game:

Setup: The challenger takes a security parameter k and runs the setup algorithm $\mathcal{S}$. It gives the adversary the resulting system parameters params. It keeps the master-key to itself.

Phase 1: The adversary issues private key extraction queries $\mathsf{ID}_1, \ldots, \mathsf{ID}_m$. The challenger responds by running $\mathcal{X}$ to extract the private key d_i corresponding to the public key ID_i. It sends d_i to the adversary. These queries may be asked adaptively.

Challenge: Once the adversary decides that Phase 1 is over it outputs a public key $\mathsf{ID} \notin \{\mathsf{ID}_1, \ldots, \mathsf{ID}_m\}$ on which it wishes to be challenged. The challenger picks a random $M \in \mathsf{MSPC}$ and encrypts M using ID as the public key. It then sends the resulting ciphertext C to the adversary.

Phase 2: The adversary issues more extraction queries $\mathsf{ID}_{m+1}, \ldots, \mathsf{ID}_n$. The only constraint is that $\mathsf{ID}_i \neq \mathsf{ID}$. The challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $M' \in \mathsf{MSPC}$ and wins the game if $M = M'$.

We refer to such an adversary $\mathcal{A}$ as an OW-ID-CPA adversary. $\mathcal{A}$'s advantage in attacking the scheme is defined as: $Adv_{\mathcal{A}}(k) = \Pr[M = M']$. The probability is taken over the random bits used by the challenger and the adversary.

Definition 2 (OW-ID-CPA)**.** *We say that an IBE scheme is secure in the sense of* OW-ID-CPA *if $Adv_{\mathcal{A}}$ is negligible for any polynomial time algorithm $\mathcal{A}$.*

Chosen Ciphertext Security. Boneh and Franklin [3] defined chosen ciphertext security for IBE systems. In their model, security for an IBE system is defined by the following IND-ID-CCA game:

Setup: The challenger takes a security parameter k and runs setup algorithm $\mathcal{S}$. It gives the adversary the resulting system parameters params and keeps the master-key to itself.

Phase 1: The adversary issues queries $q_1, \cdots, q_m$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm $\mathcal{E}$ to generate decryption key d_i which corresponds to the public key $\langle \mathsf{ID}_i \rangle$. It sends d_i to the adversary.

– Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm $\mathcal{E}$ to generate the decryption key d_i corresponding to the public key $\langle \mathsf{ID}_i \rangle$. Then it runs algorithm $\mathcal{D}$ to decrypt the ciphertext C_i using d_i. It sends the adversary the resulting plaintext.

The query may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs two equal length plaintext $M_0, M_1 \in \mathsf{MSPC}$ and an ID on which it wishes to be challenged. The only constraint is that the ID did not appear in any Extraction query in Phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and sets $C = \mathcal{E}(\mathsf{params}, \mathsf{ID}, M_b)$. It sends C to the adversary.

Phase 2: The adversary issues more queries $q_{m+1}, \cdots, q_n$ where query q_i is one of:

– Extraction query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}$. The challenger responds as in Phase 1.

– Decryption query $\langle \mathsf{ID}_i, C_i \rangle$ where $\langle \mathsf{ID}_i, C_i \rangle \neq \langle \mathsf{ID}, C \rangle$. The challenger responds as in Phase 1.

These queries may be asked adaptively as in Phase 1.

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

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. An advantage of an IND-ID-CCA adversary is defined as follows: $Adv_{\mathcal{A}}(k) = |\Pr[b = b'] - \frac{1}{2}|$. The probability is taken over the random bits used by the challenger and the adversary.

Definition 3 (IND-ID-CCA). *We say that an IBE system is secure in sense of* IND-ID-CCA *if $Adv_{\mathcal{A}}$ is negligible for any polynomial time algorithm $\mathcal{A}$.*

γ-Uniformity. A property γ-uniformity is originally defined for conventional public key encryption schemes [6]. Here, we define γ-uniformity for IBE schemes.

Definition 4 (γ-uniformity). *Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IBE scheme. For a given $\mathsf{ID} \in \{0,1\}^*$, the corresponding decryption key d, $x \in \mathsf{MSPC}$ and $y \in \mathsf{CSPC}$, define*

$$\gamma(x, y) = \Pr[h \leftarrow_R \mathsf{COIN}(k) : y = \mathcal{E}(\mathsf{params}, \mathsf{ID}, x; h)].$$

We say that Π is γ-uniform, if, for any $\mathsf{ID} \in \{0,1\}^*$, any $x \in \mathsf{MSPC}$ and any $y \in \mathsf{CSPC}$, $\gamma(x, y) \leq \gamma$.

3 Fujisaki-Okamoto Conversion for IBE Schemes

In this section, we discuss the security of the FO conversion for OW-ID-CPA secure IBE. As far as we know, this is the first formal analysis which proves that FO generically converts any OW-ID-CPA secure IBE into an IND-ID-CCA secure IBE. We also give an observation that the straightforward application of FO to achieve a strong security is insufficient.

Straightforward Application of FO. Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an OW-ID-CPA IBE. Then, we can construct another IBE $\Pi' = \{\mathcal{S}', \mathcal{X}', \mathcal{E}', \mathcal{D}'\}$ as follows:

Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π' and $\mathsf{COIN}(k)$ be Π's coin-flipping space.

- $\mathcal{S}'$, the setup algorithm. It is as $\mathcal{S}$. In addition, we pick two hash functions $G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
- $\mathcal{X}'$, the extraction algorithm. It is as $\mathcal{X}$.
- $\mathcal{E}'$, the encryption algorithm. It is defined as follows:

$$\mathcal{E}'(\mathsf{params}, \mathsf{ID}, M; \sigma) = \mathcal{E}\big(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M)\big) \| H(\sigma) \oplus M$$

- $\mathcal{D}'$, the decryption algorithm. Let $C = C_1 \| C_2$ be a ciphertext to decrypt. Algorithm $\mathcal{D}'$ works in the following steps:
  1. Computes $\mathcal{D}(\mathsf{params}, d, C_1) = \sigma$.
  2. Computes $H(\sigma) \oplus C_2 = M$
  3. Sets $r = G(\sigma, M)$. Tests that $\mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; r) = C_1$. If not, outputs "reject".
  4. Outputs M as the decryption of C

Theorem 1. *Suppose the hash functions G and H are random oracles and Π is a γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an* IND-ID-CCA *adversary which has advantage $\epsilon(k)$ against Π' and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H queries, q_G G queries, q_E Extraction queries and q_D Decryption queries. Suppose that running time of $\mathcal{E}$ is at most τ. Then there is an* OW-ID-CPA *adversary $\mathcal{A}$ which has advantage at least $\frac{1}{q_H + q_G}\big(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\big)$ against Π. Its running time is $t(k) + q_G \cdot q_D \cdot \tau$.*

Proof. We show how to construct adversary $\mathcal{A}$ by using adversary $\mathcal{B}$ as an oracle. The challenger starts an OW-ID-CPA game by executing $\mathcal{S}$ and generates params and master-key. The master-key is kept secret by the challenger. $\mathcal{A}$ works by interacting with $\mathcal{B}$ in an IND-ID-CCA game as follows:

Setup: $\mathcal{A}$ gives params to $\mathcal{B}$.

Responses to G-Queries: $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, M_i, g_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, M_i)$, $\mathcal{A}$ responds as follows:

1. If the query σ_i and M_i already appears on the G^{list} in a tuple $\langle \sigma_i, M_i, g_i \rangle$ then $\mathcal{A}$ responds with $G(\sigma_i, M_i) = g_i$.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from $\mathsf{COIN}(k)$ of Π.
3. $\mathcal{A}$ adds the tuple $\langle \sigma_i, M_i, g_i \rangle$ to the G^{list} and returns g_i.

Responses to H-Queries: $\mathcal{A}$ maintains a list of tuples $\langle M_i, h_i \rangle$ to respond the queries. We refer to this list as H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(M_i)$, $\mathcal{A}$ responds as following:

1. If the query M_i already appears on the H^{list} in a tuple $\langle M_i, h_i \rangle$ then $\mathcal{A}$ responds with $H(M_i) = h_i$.
2. Otherwise, $\mathcal{A}$ picks a string h_i from $\{0,1\}^{l_2}$ randomly.
3. $\mathcal{A}$ adds the tuple $\langle M_i, h_i \rangle$ to the H^{list} and returns h_i.

Responses to Extraction Queries: Let $\langle \mathsf{ID}_i \rangle$ be an Extraction query issued by $\mathcal{B}$. $\mathcal{A}$ inputs $\langle \mathsf{ID}_i \rangle$ to its own extraction oracle and gets the corresponding decryption key d_i. $\mathcal{A}$ passes d_i to $\mathcal{B}$ as the answer of the query.

Responses to Decryption Queries: Let $\langle \mathsf{ID}_i, C_i \rangle$ be a Decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds as follows:

1. Find a pair of tuples $\langle \sigma, M, g \rangle$ and $\langle \sigma, h \rangle$ from the G^{list} and H^{list}, respectively, such that $\mathcal{E}(\mathsf{params}, \mathsf{ID}_i, \sigma; g) \| h \oplus M_j = C_i$.
2. Outputs M if there exists such a pair of tuples, or outputs "reject" otherwise.

Challenge: Once $\mathcal{B}$ decides that Phase 1 is over it outputs a public key ID and two messages M_0, M_1 on which it wishes to be challenged. $\mathcal{A}$ sends ID to the challenger and receives a ciphertext C. Then, $\mathcal{A}$ generates $C_{ch1} \| C_{ch2}$ where $C_{ch1} = C$ and C_{ch2} is a random string whose length is l_2. $\mathcal{A}$ gives $C_{ch1} \| C_{ch2}$ as the challenge to $\mathcal{B}$.

Guess: Once $\mathcal{B}$ decides that Phase 2 is over it outputs a guess b'.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, M, g \rangle$ or $\langle \sigma, h \rangle$ from the G^{list} or the H^{list}, respectively. Then, $\mathcal{A}$ outputs σ in the tuple as the answer of the OW-ID-CPA game.

We first define the following three events:

SuccB the event that $\mathcal{B}$ wins the IND-ID-CCA game.

AskB the event that $\mathcal{B}$ asks a query for $G(\mathcal{D}(\mathsf{params}, d, C_{ch1}), *)$ or $H(\mathcal{D}(\mathsf{params}, d, C_{ch1}))$ at some point during the game, where $d := \mathcal{X}(\mathsf{params}, \mathsf{master\text{-}key}, \mathsf{ID})$ and $*$ denotes any l_2-bit string.

Fail the event that the simulation fails before $\mathcal{B}$ submits a query for $G(\mathcal{D}(\mathsf{params}, d, C_{ch1}), *)$ or $H(\mathcal{D}(\mathsf{params}, d, C_{ch1}))$.

Then, we have that

$$\Pr[\mathsf{SuccB} | \neg \mathsf{Fail}] \cdot \Pr[\neg \mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}].$$

Since $\Pr[\mathsf{SuccB} | \neg \mathsf{Fail}, \neg \mathsf{AskB}] = 1/2$, we also have

$$\begin{aligned} \Pr[\mathsf{SuccB} | \neg \mathsf{Fail}] &= \Pr[\mathsf{SuccB} | \neg \mathsf{Fail} \wedge \mathsf{AskB}] \cdot \Pr[\mathsf{AskB}] + \frac{1}{2}\Big(1 - \Pr[\mathsf{AskB}]\Big) \\ &\leq \frac{1}{2} \Pr[\mathsf{AskB}] + \frac{1}{2}. \end{aligned}$$

Hence, we have that

$$\Big(\frac{1}{2} \Pr[\mathsf{AskB}] + \frac{1}{2}\Big) \cdot \Pr[\neg \mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}],$$

and therefore,

$$\Pr[\mathsf{AskB}] \geq 2\epsilon(k) - \Pr[\mathsf{Fail}].$$

Next, we estimate $\Pr[\mathsf{Fail}]$. The event Fail occurs only when either

Case 1. $\mathcal{B}$ submits a Decryption query $\langle \mathsf{ID}, C_1 \| H(\sigma) \oplus M \rangle$ such that $C_1 = \mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M))$ without asking $G(\sigma, M)$, or

Case 2. $\mathcal{B}$ submits a Decryption query $\langle \mathsf{ID}, \mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M)) \| C_2 \rangle$ such that $C_2 = H(\sigma) \oplus M$ without asking $H(\sigma)$.

Case 1 and **2** happen with probability at most γ and $1/2^{l_2}$, respectively, and therefore, we have that $\Pr[\mathsf{Fail}] \leq 1 - (1 - \gamma - 1/2^{l_2})^{q_D}$.

Hence, we have that

$$\begin{aligned} Adv_{\mathcal{A}}(k) &\geq \frac{1}{q_G + q_H} \Pr[\mathsf{AskB}] \\ &\geq \frac{1}{q_G + q_H} \left(2\epsilon(k) - \left(1 - \left(1 - \gamma - \frac{1}{2^{l_2}} \right)^{q_D} \right) \right) \\ &\simeq \frac{1}{q_G + q_H} \left(2\epsilon(k) - q_D \gamma - \frac{q_D}{2^{l_2}} \right). \end{aligned}$$

Finally, we estimate $\mathcal{A}$'s running time. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times for responding to each Decryption query, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot q_D \cdot \tau$. □

Discussion: Running Time of $\mathcal{A}$. As shown in Theorem 1, there exists a polynomial time reduction from $\mathcal{B}$ to $\mathcal{A}$, and consequently, any polynomial time adversary cannot break Π' in IND-ID-CCA sense if any polynomial time adversary cannot break Π in OW-ID-CPA sense. However, this result does not immediately imply that any realistic adversary cannot break Π' in IND-ID-CCA sense if any realistic adversary cannot break Π in OW-ID-CPA sense. Suppose that $\mathcal{A}$'s computational time is significantly larger than $\mathcal{B}$'s. Then, it might be still infeasible to break Π in practice even if $\mathcal{B}$ can break Π' in IND-ID-CCA sense. Bellare and Rogaway [2] proposed the notion of *exact security* for formally dealing with this issue.

Now, we focus on the running times of $\mathcal{A}$ and $\mathcal{B}$ (rather than their advantages). As in Theorem 1, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot q_D \cdot \tau$, where $t(k)$ denotes $\mathcal{B}$'s running time. This means that $\mathcal{A}$ has to run the encryption algorithm $\mathcal{E}$ for $q_G \cdot q_D$ times in addition to $\mathcal{B}$'s running time. Consequently, assuming that q_G and q_D are estimated as 2^{60} and 2^{40} respectively, $\mathcal{A}$ has to run $\mathcal{E}$ for 2^{100} times! (Notice that a Decryption query requires on-line computation while a G-query only requires off-line hash computation.) It is believed that more than 2^{80} operations are computationally infeasible in the real world, and therefore, $\mathcal{A}$ cannot break OW-ID-CPA security of Π in practice (even if $\mathcal{B}$ works in a practical time).

Hence, the above straightforward application of the FO conversion is insufficient for achieving a strong security. In the next section, we propose an improved version of the FO conversion for IBE, which provides an efficient simulator with less time complexity.

4 Modified Fujisaki-Okamoto for IBE Schemes

In this section, we propose a modified FO conversion with an improved reduction cost, i.e. the simulator needs shorter running time but still obtains the same

advantage when compared with the simulator in the straightforward FO. The difference between our modification and the original FO is only that we take σ, M and ID as input to G instead of σ and M.

Basic Idea. The huge running time of $\mathcal{A}$ in Theorem 1 is caused by the following reason. In order to respond to a Decryption query $\langle \mathsf{ID}, C \rangle$, $\mathcal{A}$ has to find a pair of tuples from G^{list} and H^{list} such that its corresponding ciphertext with public key ID is identical to C. Since $\mathcal{A}$ does not know ID in advance, it is required to carry out re-encryption with public key ID for all tuples in G^{list} for every Decryption query. This results in $q_G \cdot q_D$ times of re-encryption operations. For solving this problem, we add ID as one of the inputs to G.

Modified FO Conversion. Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IBE scheme which is secure in the sense of OW-ID-CPA. We denote the new encryption scheme as $\Pi'' = \{\mathcal{S}'', \mathcal{X}'', \mathcal{E}'', \mathcal{D}''\}$. Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π'' and $\mathsf{COIN}(k)$ be Π's coin-flipping space.

- $\mathcal{S}''$, the setup algorithm. It is as $\mathcal{S}$. In addition we pick two hash functions $G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \times \{0,1\}^* \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
- $\mathcal{X}''$, the extraction algorithm. It is as $\mathcal{X}$.
- $\mathcal{E}''$, the encryption algorithm. It takes system parameter params, public key $\mathsf{ID} \in \{0,1\}^*$, random coin $\sigma \in \{0,1\}^{l_1}$ and a message $M \in \{0,1\}^{l_2}$.
  It is defined as follows:

$$\mathcal{E}''(\mathsf{params}, \mathsf{ID}, \sigma, M) = \mathcal{E}\big(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M, \mathsf{ID})\big) \| H(\sigma) \oplus M$$

- $\mathcal{D}''$, the decryption algorithm. Let $C = C_1 \| C_2$ be a ciphertext to decrypt. This algorithm works in the following four steps:
  1. Computes $\mathcal{D}(\mathsf{params}, d, C_1) = \sigma$
  2. Computes $H(\sigma) \oplus C_2 = M$
  3. Sets $r = G(\sigma, M, \mathsf{ID})$. Test that $\mathcal{E}(\mathsf{params}, \mathsf{ID}, M; r) = C_1$. If not, outputs "reject".
  4. Outputs M as the decryption of C.

Theorem 2. *Suppose the hash functions G and H are random oracles and Π is γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an* IND-ID-CCA *adversary which has advantage $\epsilon(k)$ against Π'' and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_G G-queries, q_H H-queries, q_E Extraction queries and q_D Decryption queries. Suppose that encrypting one message needs time τ. Then there is an* OW-ID-CPA *adversary $\mathcal{A}$ which has advantage at least $\frac{1}{q_H + q_G}\big(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\big)$ against Π. Its running time is $t(k) + q_G \cdot \tau$*

Proof. To prove Theorem 2, almost same strategy as the proof of Theorem 1 can be used. That is, assuming IND-ID-CCA adversary $\mathcal{B}$ for Π'', constructing OW-ID-CPA adversary $\mathcal{A}$ for Π which uses $\mathcal{B}$ as an oracle.

There are two different points between the proof of Theorem 1 and 2. The points are how to answer G-queries and Decryption-queries in the IND-ID-CCA game between $\mathcal{A}$ and $\mathcal{B}$. Due to the space limitation, we describe only these different points.

Responses to G-Queries: $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, M_i, \mathsf{ID}_i)$, $\mathcal{A}$ responds as follows:

1. If the query σ_i, M_i and ID_i already appears on the G^{list} in a tuple $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ then $\mathcal{A}$ responds with $G(\sigma_i, M_i, \mathsf{ID}_i) = a_i$.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from $\mathsf{COIN}(k)$.
3. $\mathcal{A}$ generates a ciphertext $C_i = \mathcal{E}(\mathsf{params}, \mathsf{ID}_i, \sigma_i; g_i) \| H(\sigma_i) \oplus M_i$.
4. $\mathcal{A}$ adds the tuple $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ to the G^{list} and responds to $\mathcal{B}$ with $G(\sigma_i, M_i, \mathsf{ID}_i) = g_i$.

Responses to Decryption Queries: Let $\langle \mathsf{ID}_i, C_i \rangle$ be a decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds this query in the following steps:

1. Finds a tuple $\langle \sigma_j, M_j, \mathsf{ID}_j, g_j, C_j \rangle$ from the G^{list} such that $\mathsf{ID}_i = \mathsf{ID}_j$ and $C_i = C_j$.
2. Outputs M_j if there exists such a tuple, or outputs "reject" otherwise.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, M, \mathsf{ID}, g, C \rangle$ or $\langle \sigma, h \rangle$ from the G^{list} or the H^{list} randomly and outputs σ in the tuple as the answer of the $\mathsf{OW\text{-}ID\text{-}CPA}$ game.

The advantage of $\mathcal{A}$ can be evaluate in the same way as in Theorem 1. So, we omit to describe the detail of the evaluation here.

Finally, we estimate $\mathcal{A}$'s running time. In addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times to make the G^{list}. Thus, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot \tau$. □

Comparison. Here, we compare the running times of simulators for Π' and Π''. In the comparison, we especially focus on times to run the encryption algorithm $\mathcal{E}$ which is required for each simulation. It is believed that if a simulator has to run $\mathcal{E}$ for more than 2^{80} times, then it does not properly work in a realistic time. Now, we have that

$$\#_{\mathcal{E}}(\Pi')(= 2^{100}) \gg 2^{80} \gg \#_{\mathcal{E}}(\Pi'')(= 2^{60}),$$

where $\#_{\mathcal{E}}(\cdot)$ denotes the times to run $\mathcal{E}$ in the simulation. This implies that the running time of the simulator for Π'' is considered realistic, and on the other hand, that for Π' is not.

However, it should be noticed that existence of an adversary which can break Π'' does not always imply existence of another adversary which can break Π in practice. This is due to its non-tight reduction cost in terms of advantage, i.e. $\frac{2}{q_G + q_H}\epsilon(k)$.

5 Conclusion

In this paper, we confirmed the generic security of FO conversion in IBE schemes, and investigated the fact that there exists a significantly inefficient reduction cost in the straightforward application, say, the additional 2^{100} times re-encryption computation. Under this circumstance, we modified FO and reduced the additional time down to 2^{60} times re-encryption computation.

Our discussion started from the OW-ID-CPA schemes, and we can also address the case starting from the IND-ID-CPA schemes. When we apply REACT [9] and the *PKC '99* version of FO [5] to IBE, some similar but more interesting results will appear. We will present them in the full version of this paper.

Acknowledgements

Rui Zhang is supported by a JSPS Fellowship. We appreciate the anonymous referees of AAECC for helpful comments and suggestions. And we are also grateful to Yang Cui for the constructive discussions.

References

1. N. Attrapadung, Y. Cui, G. Hanaoka, H. Imai, K. Matsuura, P. Yang, and R. Zhang. Relations among notions of security for identity based encryption schemes. Cryptology ePrint Archive, Report 2005/258, 2005. `http://eprint.iacr.org/2005/258`.
2. M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with rsa and rabin. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
4. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - EUROCRYPT '03*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.
5. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography - PKC '99*, volume 1560 of *LNCS*, pages 53–68. Springer, 1999.
6. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.
7. D. Galindo and I. Hasuo. Security notions for identity based encryption. Cryptology ePrint Archive, Report 2005/253, 2005. `http://eprint.iacr.org/2005/253`.
8. C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT '02*, volume 2501 of *LNCS*, pages 548–566. Springer, 2002.
9. T. Okamoto and D. Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology – CT-RSA '01*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–174. Springer, 2001.
10. C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.
11. A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.