

A survey of IND-CCA constructions

Ganyu (Bruce) Xu (g66xu)

CO 789, Winter 2024

1 Introduction

2 Preliminaries

3 The Fujisaki-Okamoto Transformation

The Fujisaki-Okamoto transformation [3] takes a IND-CPA public-key encryption scheme (PKE) and a few other cryptographic primitives of weaker security as inputs, and constructs a hybrid encryption scheme that achieves IND-CCA2 security.

In the conference paper, the security reduction is performed under the random oracle model, and the security of the hybrid scheme degrades linearly with the number of hash queries and the number of decryption queries. On the other hand, this scheme is desirable for its simplicity, and the low requirements for the input primitives: the PKE only needs to be one-time one-way secure, and the symmetric cipher only needs to be indistinguishable under one-time attack.

3.1 The hybrid scheme and security result

The hybrid encryption scheme contains three routines: key generation, encryption, and decryption. The input for the hybrid scheme includes a public-key encryption scheme $\text{KeyGen}^{\text{asym}}, E^{\text{asym}}, D^{\text{asym}}$, a symmetric encryption scheme $E^{\text{sym}}, D^{\text{sym}}$, and two hash functions $G : \mathcal{M}^{\text{asym}} \mapsto \mathcal{K}^{\text{sym}}, H : \mathcal{M}^{\text{asym}} \times \mathcal{C}^{\text{sym}} \mapsto \text{Coin}^{\text{asym}}$.

Algorithm 1 FO Key generation

- 1: $(\text{PK}^{\text{asym}}, \text{SK}^{\text{asym}}) \leftarrow \text{PKE.KeyGen}()$
 - 2: $\text{PK}^{\text{hy}} \leftarrow \text{PK}^{\text{asym}}, \text{SK}^{\text{hy}} \leftarrow \text{SK}^{\text{asym}},$
 - 3: **return** $(\text{PK}^{\text{hy}}, \text{SK}^{\text{hy}})$
-

Algorithm 2 FO Key encryption

Require: $m \in \mathcal{M}^{\text{sym}}$

- 1: Sample from the PKE's message space $\sigma \xleftarrow{\$} \mathcal{M}^{\text{asym}}$
 - 2: $a \leftarrow G(\sigma), c \leftarrow E_a^{\text{sym}}(m)$
 - 3: $h \leftarrow H(\sigma, c)$
 - 4: $e \leftarrow E^{\text{asym}}(\text{PK}^{\text{hy}}, \sigma, h)$
 - 5: **return** (e, c)
-

The decryption routine:

Algorithm 3 FO Key decryption

Require: The ciphertext (e, c)

- 1: $\hat{\sigma} \leftarrow D^{\text{asym}}(\text{SK}^{\text{hy}}, e)$
 - 2: $\hat{h} \leftarrow H(\hat{\sigma}, c)$
 - 3: $\hat{e} \leftarrow E^{\text{asym}}(\text{PK}^{\text{hy}})$
 - 4: **if** $\hat{e} \neq e$ **then**
 - 5: **return** \perp
 - 6: **end if**
 - 7: $\hat{a} \leftarrow G(\hat{\sigma})$
 - 8: $\hat{m} \leftarrow D_a^{\text{sym}}(c)$
 - 9: **return** \hat{m}
-

Theorem 3.1. *For all IND-CCA adversary \mathcal{A}^{hy} against the hybrid encryption scheme with advantage ϵ^{hy} , there exists a one-way one-time-encryption adversary against the public-key encryption scheme with advantage ϵ^{asym} and an one-time indistinguishability adversary against the symmetric encryption scheme with advantage ϵ^{sym} such that*

$$\epsilon^{\text{hy}} \leq q_H \epsilon^{\text{asym}} + \epsilon^{\text{sym}} + q_D 2^{-\gamma}$$

where γ is the spread of the public-key encryption scheme, q_H is the total number of hash queries, and q_D is the total number of decryption queries

3.2 Proof of security result

Theorem 3.1 is proved using a sequence of games that involves \mathcal{A}^{hy} as the main routine, and two games that involves \mathcal{A}^{hy} as a sub-routine. The sequence of games is as follows:

- Game 0 is the standard IND-CCA2 game
- Game 1 is identical to game 0, except that the decryption oracle \mathcal{O}_D is modified. Instead of using the true secret key SK^{hy} to decrypt the query (e_q, c_q) , the decryption oracle checks the tape of hash function H for the existence of hash query (σ_H, c_H, h_H) such that $c_q = c_H$ and $e_q = E^{\text{asym}}(\text{PK}^{\text{asym}}, \sigma_H, h_H)$. If such a query exists, then \mathcal{O}_D uses σ_H to derive the symmetric key $a_q \leftarrow G(\sigma_H)$ and decrypt the queried ciphertext c_q . If no such query exists, then \mathcal{O}_D will reject the queried ciphertext and output a decryption error. It is worth noting that this modified decryption oracle does not require the hybrid secret key SK^{hy} to process decryption queries.
- Game 2 is identical to game 1, except the routine of encrypting the challenge ciphertext is modified: $a^* \xleftarrow{\$} \mathcal{K}^{\text{sym}}$ is randomly sampled from the symmetric key space instead of being queried from G , and $h^* \xleftarrow{\$} \text{COIN}^{\text{asym}}$ is randomly sampled from the asymmetric coin space instead of being queried from H .

Let S_0, S_1, S_2 denote the event that \mathcal{A}^{hy} wins game 0, game 1, and game 2, respectively.

Lemma 3.1.1. *Let q_D denote the number of decryption queries, and γ denote the spread of the PKE, then*

$$P[S_0] - P[S_1] \leq q_D 2^{-\gamma}$$

Proof. For each decryption query (e_q, c_q) , there are three mutually exclusive possibilities:

1. The queried ciphertext is **honest**, meaning that there is a matching record on the tape of the hash function H
2. There is no matching record on the tape of the hash function H , and the check in step 4 in algorithm 3 will fail, outputting decryption error. Such a query is called **invalid**

3. There is no matching record on the tape of the hash function H , but the check in step 4 of algorithm 3 will succeed. Such a query is called **almost valid**

Observe that for both S_0, S_1 :

$$\begin{aligned} P[S] &= P[S \cap \text{all decryption queries are honest}] \\ &\quad + P[S \cap \text{some decryption queries are dishonest, but none is almost valid}] \\ &\quad + P[S \cap \text{at least one almost valid decryption queries}] \end{aligned}$$

When all decryption queries are honest, the decryption oracles will correctly decrypt the query in both game 0 and game 1. When all dishonest decryption queries are invalid, the decryption oracles will reject the query in both games. The only difference between the two games lies in how the decryption oracle processes almost valid decryption queries. Therefore:

$$\begin{aligned} P[S_0] - P[S_1] &= P[S_0 \cap \text{at least one almost valid decryption queries}] \\ &\quad - P[S_1 \cap \text{at least one almost valid decryption queries}] \\ &\leq P[\text{at least one almost valid decryption queries}] \end{aligned}$$

Let (e, c) be some decryption query made without querying H , then in the true decryption routine, $\hat{h} \leftarrow H(\hat{\sigma}, c)$ will be a truly random coin, and $\hat{e} \leftarrow E^{\text{asym}}(\hat{\sigma}, \hat{h})$ will be a truly random ciphertext for the given public key and $\hat{\sigma}$. Since the PKE has γ spread, we know that $P[e = \hat{e}] = P[(e, c) \text{ is almost valid}] \leq 2^{-\gamma}$. Among q_D decryption query, the probability of having at least one almost valid query is bounded by sum of probability of each decryption query being almost valid: $P[\text{at least one almost valid query}] \leq q_D 2^{-\gamma}$. \square

If during the IND-CCA game, $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ never makes hash query that involves σ^* , then under the random oracle model, there is no difference between sampling a^*, h^* randomly or pseudorandomly. On the other hand, if the adversary does make such a query, then there is an inconsistency between the challenge ciphertext and the results of such hash query. In other words, the difference between game 1 and game 2 is the even that the IND-CCA adversary makes such a hash query:

Lemma 3.1.2.

$$P[S_1] - P[S_2] = P[\text{query}]$$

Denote the challenge ciphertext in game 2 by (e^*, c^*) . Notice that e^* is the encryption of a truly random σ^* using a truly random coin h^* , while c^* is the encryption of m_b^* under a truly random key a^* . This allows a OW-CPA game for the PKE, and an IND-CPA game for the symmetric cipher to be perfectly simulated by some $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ and/or $\mathcal{A}_{\text{IND-CPA}}^{\text{sym}}$. Using this strategy, we can bound $P[S_2]$ and $P[\text{query}]$ using the advantage of some OW-CPA and/or IND-CPA adversaries.

Lemma 3.1.3. *For every IND-CCA adversary $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$, there exists an IND-OTE adversary against the underlying symmetric cipher $\mathcal{A}_{\text{IND-OTE}}^{\text{sym}}$ with advantage $\epsilon_{\text{IND-OTE}}^{\text{sym}}$ such that*

$$P[S_2] = \frac{1}{2} + \epsilon_{\text{IND-OTE}}^{\text{sym}}$$

Proof. $\mathcal{A}_{\text{IND-OTE}}^{\text{sym}}$ can perfectly simulate the hybrid key generation $\text{PK}^{\text{hy}} = \text{PK}^{\text{asym}}, \text{SK}^{\text{hy}} = \text{SK}^{\text{asym}}$, the random oracles G, H , as well as the modified decryption oracle.

When $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ submits the challenge plaintexts (m_0, m_1) , $\mathcal{A}_{\text{IND-OTE}}^{\text{sym}}$ passes them to the symmetric cipher challenger and receives the symmetric challenge ciphertext c^* . $\mathcal{A}_{\text{IND-OTE}}^{\text{sym}}$ then randomly samples $\sigma^* \xleftarrow{\$} \mathcal{M}^{\text{asym}}$ and $h^* \xleftarrow{\$} \text{COIN}^{\text{asym}}$ and computes $e^* \leftarrow E^{\text{asym}}(\text{PK}^{\text{asym}}, \sigma^*, h^*)$. (e^*, c^*) is given to $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ as the challenge ciphertext. It is easy to verify that from $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$'s perspective, this game is exactly game 2; in addition $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ wins the game if and only if $\mathcal{A}_{\text{IND-OTE}}^{\text{sym}}$ wins the game. \square

Lemma 3.1.4. *For every IND-CCA adversary $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$, there exists an OW-CPA adversary $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ with advantage $\epsilon_{\text{OW-CPA}}^{\text{asym}}$ such that*

$$\epsilon_{\text{OW-CPA}}^{\text{asym}} = P[\text{query}] \cdot \frac{1}{q_H}$$

where q_H is the number of hash queries $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ makes to either H or G .

Proof. Similar to the proof of lemma 3.1.3, $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ can perfectly simulate key generation, hash oracles, and decryption oracle.

Let e^* denote the asymmetric challenge ciphertext $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ receives from the OW-CPA challenger, let σ^* denote the plaintext that corresponds with e^* , and let (m_0, m_1) denote the challenge plaintexts submitted by $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$. $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ samples a random symmetric key $a^* \xleftarrow{\$} \mathcal{K}^{\text{sym}}$ and a coin flip $b \xleftarrow{\$} \{0, 1\}$, then computes $c^* \leftarrow E_{a^*}^{\text{sym}}(m_b)$. (e^*, c^*) is the IND-CCA challenge ciphertext.

After the $\mathcal{A}_{\text{IND-CCA}}^{\text{hy}}$ halts, $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ looks through the tape of the hash oracles $\mathcal{O}^H = \{(\tilde{\sigma}, \tilde{c})\}$ and $\mathcal{O}^G = \{\tilde{\sigma}\}$, then picks a random value among all possible $\tilde{\sigma}$'s to output. $\mathcal{A}_{\text{OW-CPA}}^{\text{asym}}$ wins if $\mathcal{A}_{\text{OW-CPA}}^{\text{hy}}$ makes a query $\tilde{\sigma} = \sigma^*$ and the correct query is chosen. \square

4 Tweaking FO Transform for IND-CCA KEM

While the FO transformation can construct IND-CCA public-key encryption scheme from OW-CPA PKE and IND-OTE symmetric cipher, there are a few drawbacks to the hybrid scheme as it is:

- The security proof is not tight. The security of the hybrid scheme degrades linearly with the number of hash queries.
- The hybrid scheme assumes the underlying PKE to be always correct. This is not the case in PKEs such as schemes based on "Learning with Errors" (LWE), which can have a small but non-zero probability of decryption error

In 2017, Hofheinz et al [4] improved the FO transformation by addressing the non-tight security and accounting for possible decryption errors. In addition, the paper proposed various IND-CCA KEM constructions from weak PKE's that does not require a symmetric cipher. In fact, both Kyber [2] and Classic McEliece [1] make direct use of transformations proposed in this paper to achieve IND-CCA security for their respective KEMs.

The strategy for achieving IND-CCA KEM consists of two parts

- Transform a OW-CPA and/or IND-CPA PKE into an OW-PCVA PKE
- Transform the OW-PCVA PKE into a IND-CCA KEM

4.1 Preliminaries

4.2 From OW-CPA to OW-PCVA

4.3 IND-CCA KEM

5 OAEP and RSA-OAEP

References

- [1] Martin R Albrecht, Daniel J Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo Von Maurich, Rafael Misoczki, Ruben Niederhagen, et al. Classic mceliece: conservative code-based cryptography. 2022.

- [2] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber algorithm specifications and supporting documentation. *NIST PQC Round*, 2(4):1–43, 2019.
- [3] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual international cryptology conference*, pages 537–554. Springer, 1999.
- [4] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.