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 symetric 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

```
$(\text{PK}^{\text{asym}}, \text{SK}^{\text{asym}}) \leftarrow \text{PKE.KeyGen}()$
$\text{PK}^{\text{hy}} \leftarrow \text{PK}^{\text{asym}}, \text{SK}^{\text{hy}} \leftarrow \text{SK}^{\text{asym}}$,
return $(\text{PK}^{\text{hy}}, \text{SK}^{\text{hy}})$
```

Algorithm 2 FO Key encryption

```
Require: $m \in \mathcal{M}^{\text{sym}}$
Sample from the PKE's message space $\sigma \xleftarrow{\$} \mathcal{M}^{\text{asym}}$
$a \leftarrow G(\sigma)$, $c \leftarrow E_a^{\text{sym}}(m)$
$h \leftarrow H(\sigma, c)$
$e \leftarrow E^{\text{asym}}(\text{PK}^{\text{hy}}, \sigma, h)$
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: $\hat{m} \leftarrow D^{\text{sym}}_{\hat{a}}(c)$
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*

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

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 $\mathcal{A}^{\text{hy}}$ as the main routine, and two games that involves $\mathcal{A}^{\text{hy}}$ 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 $\mathcal{A}^{\text{hy}}$ 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, outputing 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:

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

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:

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

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 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: $P[\text{ at least one almost valid query }] \leq q_D 2^{-\gamma}$. □

If during the IND-CCA game, $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$ 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.

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

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 $\mathcal{A}^{\text{asym}}_{\text{OW-CPA}}$ and/or $\mathcal{A}^{\text{sym}}_{\text{IND-CPA}}$. 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 $A^{hy}_{IND\text{-}CCA}$, there exists an IND-OTE adversary against the underlying symmetric cipher $\mathcal{A}^{sym}_{IND\text{-}OTE}$ with advantage $\epsilon^{sym}_{IND\text{-}OTE}$ such that*

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

Proof. $\mathcal{A}^{\text{sym}}_{\text{IND-OTE}}$ 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 $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$ submits the challenge plaintexts (m_0, m_1), $\mathcal{A}^{\text{sym}}_{\text{IND-OTE}}$ passes them to the symmetric cipher challenger and receives the symmetric challenge ciphertext c^*. $\mathcal{A}^{\text{sym}}_{\text{IND-OTE}}$ 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 $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$ as the challenge ciphertext. It is easy to verify that from $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$'s perspective, this game is exactly game 2; in addition $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$ wins the game if and only if $\mathcal{A}^{\text{sym}}_{\text{IND-OTE}}$ wins the game. □

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

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

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

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

Let e^* denote the asymmetric challenge ciphertext $\mathcal{A}^{\text{asym}}_{\text{OW-CPA}}$ 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 $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$. $\mathcal{A}^{\text{asym}}_{\text{OW-CPA}}$ samples a random symmetric key $a^* \xleftarrow{\$} \mathcal{K}^{\text{sym}}$ and a coin flip $b \xleftarrow{\$} \{0,1\}$, then computes $c^* \leftarrow E^{\text{sym}}_{a^*}(m_b)$. (e^*, c^*) is the IND-CCA challenge ciphertext.

After the $\mathcal{A}^{\text{hy}}_{\text{IND-CCA}}$ halts, $\mathcal{A}^{\text{asym}}_{\text{OW-CPA}}$ 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. $\mathcal{A}^{\text{asym}}_{\text{OW-CPA}}$ wins if $\mathcal{A}^{\text{hy}}_{\text{OW-CPA}}$ makes a query $\tilde{\sigma} = \sigma^*$ and the correct query is chosen. □

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

The intermediary security definition PCVA is defined by two oracles: the **plaintext checking oracle (PCO)** and the **ciphertext validation oracle (CVO)**. The details of these two oracles will be discussed in the following section

4.1 From OW/IND-CPA to OW-PCVA

Let (KeyGen, E, D) denote the input public-key encryption scheme, let $(\text{KeyGen}^T, E^T, D^T)$ denote the transformed public-key encryption scheme, let G be a hash function. The transformed routines are as follows:

Algorithm 4 KeyGen^T

```
$\text{pk}, \text{sk} \xleftarrow{\$} \text{KeyGen}()$
**return** $\text{pk}^T \leftarrow \text{pk}, \text{sk}^T \leftarrow \text{sk}$
```

Algorithm 5 Encryption routine: E^T

```
Require: The message $m \in \mathcal{M}$
$r \leftarrow G(m)$                                  ▷ $r$ serves as the coin
$c \leftarrow E(\mathrm{pk}^T, m, r)$
return $c$
```

Algorithm 6 Decryption routine: D^T

```
Require: ciphertext $c \in \mathcal{C}$
$\hat{m} \leftarrow D(\mathrm{sk}, c)$
$\hat{r} \leftarrow G(\hat{m})$
if $E(\mathrm{pk}, \hat{m}, \hat{r}) \neq c$ then
    return $\perp$
end if
return $\hat{m}$
```

To discuss the security of the transformed scheme, we first need to lay out the implementations of the PCO and the CVO. In the vanilla PCVA game, the two oracles have access to the challenge secret key and perform decryption routines:

Algorithm 7 Plaintext checking oracle PCO_0

```
Require: A query plaintext-ciphertext pair $(\tilde{m}, \tilde{c})$
$\hat{m} \leftarrow D(\mathrm{sk}, \tilde{c})$
if $\hat{m} \neq \tilde{m}$ then
    return 0                                         ▷ query is not valid
end if
$\tilde{r} \leftarrow G(\tilde{m})$
if $E(\mathrm{pk}, \tilde{m}, \tilde{r}) \neq \tilde{c}$ then
    return 0                                         ▷ query is also not valid
end if
return 1                                             ▷ query is valid
```

Algorithm 8 Ciphertext validation oracle CVO_0

```
Require: A query ciphertext $\tilde{c}$
$\hat{m} \leftarrow D(\text{sk}, \tilde{c})$
if $\hat{m} = \perp$ then
    return 0                                   ▷ ciphertext is not valid
else if $E(\text{pk}, \hat{m}, G(\hat{m})) \neq \tilde{c}$ then
    return 0                                   ▷ ciphertext is not valid
end if
return 1                                       ▷ ciphertext is valid
```

To link the advantage of an OW-PCVA adversary, a sequence of games is devised in which PCO and CVO are incrementally modified to not require the secret key and the challenge encryption to be performed with a truly random coin. These modifications will allow the OW-PCVA game to be perfectly simulated by an OW/IND-CPA adversary, thus establishing the security of the transformed encryption scheme upon the security of the input encryption scheme.

Theorem 4.1 (IND-CPA to OW-PCVA)**.** *Under the random-oracle model, for every OW-PCVA adversary $\mathcal{A}^T_{OW\text{-}PCVA}$ against the transformed PKE with advantage $\epsilon^T_{OW\text{-}PCVA}$, there exists an IND-CPA adversary $\mathcal{A}_{IND\text{-}CPA}$ against the input PKE with advantage $\epsilon_{IND\text{-}CPA}$ such that*

$$\epsilon^T_{OW\text{-}PCVA} \leq q_V \cdot 2^{-\gamma} + q_G \cdot \delta + \frac{1}{|\mathcal{M}|} + 3 \cdot \epsilon_{IND\text{-}CPA}$$

Where γ is the spread of ciphertext of the input PKE, δ is the probability of decryption error of the input PKE, q_V is the number of ciphertext validation queries, and q_G is the number of hash queries to G.

Proof. Admittedly this is only a sketch. Some nuance is omitted for the sake of clarity, but the overall result will be off by no more than some negligible constants.

We device a sequence of games that $\mathcal{A}^T_{\text{OW-PCVA}}$ plays. Game 0 is the standard OW-PCVA game.

Game 1 is identical to game 0, except that CVO has been modified to check the tape of the hash function G for plaintext $\tilde{m}$ that encrypts into the queried ciphertext $\tilde{c}$ under E^T:

Algorithm 9 CVO_1

```
Require: queried ciphertext $\tilde{c}$
for $(\tilde{m}, \tilde{r})$ on the tape of $G$ do
    if $E(\text{pk}, \tilde{m}, \tilde{r}) = \tilde{c}$ then
        return 1                               ▷ queried ciphertext is valid
    end if
end for
return 0                                       ▷ queried ciphertext is invalid
```

Following similar logic in the FO transformation's security reduction, it's easy to see that game 0 and game 1 diverge when $\mathcal{A}^T_{\text{OW-PCVA}}$ makes at least one CVO query $\tilde{c}$ whose decryption encrypts correctly (aka passes CVO in game 0) but without querying G (aka fails CVO in game 1)

$$P[\text{wins game 0}] - P[\text{wins game 1}] \leq q_V \cdot 2^{-\gamma} \tag{1}$$

Game 2 is identical to game 1, except that PCO is modified to check only the encryption instead of checking both encryption and decryption

Algorithm 10 PCO$_2$

```
Require: query $(\tilde{m}, \tilde{c})$
for $(m, r)$ from the tape of $G$ do
    if $m = \tilde{m}$ and $E(\text{pk}, m, r) = \tilde{c}$ then
        return 1
    end if
end for
return 0
```

Game 2 diverges from game 1 if $\mathcal{A}^T_{\text{OW-PCVA}}$ makes at least one hash query $(\tilde{m}, \tilde{r})$ such that

$$D(\text{sk}, E(\text{pk}, \tilde{m}, \tilde{r})) \neq \tilde{m}$$

Therefore

$$P[\text{wins game 1}] - P[\text{wins game 2}] \leq q_G \cdot \delta \tag{2}$$

Game 3 is identical to game 2, except that when encrypting the challenge ciphertext c^*, a truly random coin ($r^* \xleftarrow{\$} \text{Coin}$) is used instead of a pseudorandom coin $r^* \leftarrow G(m^*)$. Game 3 diverges from game 2 if $\mathcal{A}^T_{\text{OW-PCVA}}$ ever queries the hash of the challenge plaintext m^*:

$$P[\text{wins game 2}] - P[\text{wins game 3}] \leq P[\text{query } m^*] \tag{3}$$

In game 3, PCO and CVO can be simulated without the secret key, and the challenge ciphertext is computed using a truly random coin instead of the hash of the challenge plaintext. This allows game 3 to be perfectly simulated by some OW-CPA adversary $\mathcal{A}_{\text{OW-CPA}}$ against the input PKE:

- When $\mathcal{A}_{\text{OW-CPA}}$ receives pk, the public key is passed to $\mathcal{A}^T_{\text{OW-PCVA}}$
- $\mathcal{A}_{\text{OW-CPA}}$ simulates G, PCO$_2$, CVO$_1$ for $\mathcal{A}^T_{\text{OW-PCVA}}$
- When $\mathcal{A}_{\text{OW-CPA}}$ receives the challenge ciphertext c^*, it is passed to $\mathcal{A}^T_{\text{OW-PCVA}}$ as the OW-PCVA challenge ciphertext
- When $\mathcal{A}^T_{\text{OW-PCVA}}$ outputs m, $\mathcal{A}_{\text{OW-CPA}}$ outputs the same value

It's easy to see that from $\mathcal{A}^T_{\text{OW-PCVA}}$'s perspective, it is playing game 3, and $\mathcal{A}^T_{\text{OW-PCVA}}$ wins if and only if $\mathcal{A}_{\text{OW-CPA}}$ wins:

$$P[\mathcal{A}^T_{\text{OW-PCVA}} \text{ wins game 3}] = \epsilon_{\text{OW-CPA}} = \epsilon_{\text{IND-CPA}} + \frac{1}{|\mathcal{M}|} \tag{4}$$

Note that $\epsilon_{\text{OW-CPA}} = \epsilon_{\text{IND-CPA}} + \frac{1}{|\mathcal{M}|}$ is a well-known results

The final piece of this puzzle is to bound the probability of the event that the $\mathcal{A}^T_{\text{OW-PCVA}}$ queries the hash of the challenge plaintext m^*. Similar to the argument above, game 3 can be perfectly simulated by an IND-CPA adversary $\mathcal{A}_{\text{IND-CPA}}$ against the underlying PKE:

- The IND-CPA pk is directly passed to $\mathcal{A}^T_{\text{OW-PCVA}}$
- $\mathcal{A}_{\text{IND-CPA}}$ simulates G, PCO$_2$, CVO$_1$ for $\mathcal{A}^T_{\text{OW-PCVA}}$
- $\mathcal{A}_{\text{IND-CPA}}$ submits randomly sampled messages (m_0, m_1) as the challenge plaintexts. Upon receiving the IND-CPA challenge ciphertext c^*, $\mathcal{A}_{\text{IND-CPA}}$ passes c^* to $\mathcal{A}^T_{\text{OW-PCVA}}$ as the OW-PCVA challenge ciphertext
- After $\mathcal{A}^T_{\text{OW-PCVA}}$ halts, $\mathcal{A}_{\text{IND-CPA}}$ checks the tape of the hash function G. If either m_0 or m_1 is on the tape, then it is returned; if neither is on the tape, then $\mathcal{A}_{\text{IND-CPA}}$ randomly returns 0 or 1

Intuitively we can argue the following: if $\mathcal{A}^{T}_{\text{OW-PCVA}}$ indeed queries m^*, then $\mathcal{A}_{\text{IND-CPA}}$ wins with probability 1; otherwise, $\mathcal{A}_{\text{IND-CPA}}$ wins with probability $\frac{1}{2}$:

$$P[\text{IND-CPA adversary wins}] = P[\text{query } m^*] + (1 - P[\text{query } m^*]) \cdot \frac{1}{2}$$

Which transforms into

$$P[\text{query } m^*] = 2\epsilon_{\text{IND-CPA}} \tag{5}$$

Putting equations 1, 2, 3, 4, 5 together gives us the result

□

4.2 IND-CCA KEM

Let H be a hash function, (KeyGen, E, D) be an IND-CPA public-key encryption scheme, and (E^T, D^T) be the OW-PCVA public-key encryption obtained by running the IND-CPA PKE through the transformation described above. We first present the KEM, then discuss its security properties

The KEM's key generation routine is identical to the key generation routine of the IND-CPA PKE.

Algorithm 11 Encapsulate

```
Require: pk
Sample random $m \xleftarrow{\$} \mathcal{M}$
$c \leftarrow E^T(\text{pk}, m)$                    ▷ the ciphertext
$K \leftarrow H(m, c)$                              ▷ the shared secret
return $(c, K)$
```

Algorithm 12 Decapsulate

```
Require: sk, the ciphertext $c$
$m \leftarrow D^T(\text{sk}, c)$
if $m = \perp$ then
    return $\perp$
end if
return $H(m, c)$
```

We denote the KEM by its routines $(\text{KeyGen}, E^{U^{\perp}}, D^{U^{\perp}})$

Theorem 4.2. *For every IND-CCA adversary $\mathcal{A}^{U^{\perp}}_{\textit{IND-CCA}}$ against the KEM $(\textit{KeyGen}, E^{U^{\perp}}, D^{U^{\perp}})$ with advantage $\epsilon^{U^{\perp}}_{\textit{IND-CCA}}$, there exists an OW-PCVA adversary $\mathcal{A}^{T}_{\textit{OW-PCVA}}$ against the transformed PKE (E^T, D^T) with advantage $\epsilon^{T}_{\textit{OW-PCVA}}$ such that*

$$\epsilon^{U^{\perp}}_{\textit{IND-CCA}} \leq \epsilon^{T}_{\textit{OW-PCVA}} \tag{6}$$

This security result is obtained through a sequence of games. When we start at the IND-CCA game, there is a decapsulation oracle that uses the true secret key, but through the sequence of games, the decapsulation oracle is modified to not require the secret key, so it can be simulated by an OW-PCVA adversary.

Proof. Game 0 is the KEM IND-CCA game

Game 1 is identical to game 0, but we modify both the hash oracle H and the decapsulation oracle $\mathcal{O}^D$ to remove the use of the secret key.

Algorithm 13 H_1, the hash oracle in game 1

```
Require: input $\tilde{m}, \tilde{c}$
Require: plaintext checking oracle PCO
if $\exists(m, c, K) \in \text{tape}_H$ such that $m = \tilde{m}, c = \tilde{c}$ then
    return $K$
end if
$\tilde{K} \xleftarrow{\$} \{0,1\}^n$
if PCO validates $\tilde{m}, \tilde{c}$ then
    Add $(\tilde{c}, \tilde{K})$ to the tape of $\mathcal{O}^D$
end if
Add $(\tilde{m}, \tilde{c}, \tilde{K})$ to $\text{tape}_H$
return $\tilde{K}$
```

Algorithm 14 $\mathcal{O}^D_1$, the decapsulation oracle in game 1

```
Require: the queried ciphertext $\tilde{c}$
Require: a ciphertext validity oracle CVO
if CVO fails to validate $\tilde{c}$ then
    return $\perp$
end if
if $\exists(c, K)$ on $\mathcal{O}^D$'s tape such that $c = \tilde{c}$ then
    return $K$
end if
Sample $\tilde{K} \xleftarrow{\$} \{0,1\}^n$
Add $(\tilde{c}, \tilde{K})$ to the tape of $\mathcal{O}^D$
return $\tilde{K}$
```

Intuitively, we can argue that the PCO and the CVO ensures the integrity of the ciphertext, and that the hash tape and the decapsulation tape ensure the consistency of the queries. Therefore, from the adversary's point of view, game 1 and game 0 are perfectly identical.

$$P[S_0] - P[S_1] = 0 \tag{7}$$

Game 2 is identical to game 1, except that when queried on the challenge plaintext/ciphertext pair (m^*, c^*) by the adversary, H consistently returns a separately randomly sampled value K. Game 2 diverges from game 1 when $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$ makes such a query

$$P[S_1] - P[S_2] \leq P[\text{query}] \tag{8}$$

In game 2, since $H(m^*, c^*)$ returns truly random value instead of pseudorandom value, the KEM adversary can obtain no information that correlates with K^*. In other words, no adversary can have any advantage in game 2:

$$P[S_2] = \frac{1}{2} \tag{9}$$

On the other hand, $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$ making query to m^* can be taken advantage of by the OW-PCVA adversary to recover m^* in the OW-PCVA game:

- The OW-PCVA public key is passed directly to $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$
- The hash function H and the decapsulation oracle (implemented using PCO and CVO) can all be perfectly simulated by $\mathcal{A}^{T}_{\text{OW-PCVA}}$
- When $\mathcal{A}^{T}_{\text{OW-PCVA}}$ receives the OW-PCVA challenge ciphertext c^*, it samples a random key $K^* \xleftarrow{\$} \{0,1\}^n$ and passes c^*, K^* to $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$

- When $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$ halts, $\mathcal{A}^{T}_{\text{OW-PCVA}}$ checks the tape of H. If there exists $(\tilde{m}, \tilde{c})$ such that $\tilde{c} = c^*$, then $\mathcal{A}^{T}_{\text{OW-PCVA}}$ returns $\tilde{m}$. If no such query exists, a random value is returned

Using the simulation above, we know that if $\mathcal{A}^{U^\perp}_{\text{IND-CCA}}$ makes the query (m^*, c^*), then $\mathcal{A}^{T}_{\text{OW-PCVA}}$ will win its game. Therefore, the probability of "query" cannot be more than the advantage of the OW-PCVA adversary:

$$P[\text{query}] \leq \epsilon^{T}_{\text{OW-PCVA}} \tag{10}$$

Combining equations 7, 8, 9, 10 provides the desired result. □

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, Tancrède 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.