A survey of generic IND-CCA2 transformations

Ganyu Xu

July 5, 2024

1 Preliminaries

1.1 Public-key encryption schemes

A public key encryption scheme $\texttt{PKE} = (\texttt{KeyGen}, \texttt{E}, \texttt{D})$ is a collection of three routines. $\texttt{KeyGen}(1^\lambda)$ takes the security parameteras input and returns a keypair $(\texttt{pk}, \texttt{sk})$. $\texttt{E}(\texttt{pk}, m)$ takes some public key and some plaintext message $m \in \mathcal{M}_{\texttt{PKE}}$ and output a ciphertext $c \in \mathcal{C}_{\texttt{PKE}}$. Where the encryption routine is probabilistic, we model the randomness using a coin $r \in \mathcal{R}$ such that $E(\texttt{pk}, m; r)$ is deterministic with an explicit r. Finally, $\texttt{D}(\texttt{sk}, c)$ uses the secret key to decrypt the ciphertext.

1.1.1 Correctness

Conventionally we require a $\texttt{PKE}$ to be perfectly correct. This means that for all possible key pairs $(\texttt{pk}, \texttt{sk})$ and plaintexts m, the decryption routine always correctly inverts the encryption routine: $\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m)) = m$.

Where perfect correctness is not achieved, such as with most lattice-based encryption schemes, we need to account for the possiblity that decryption can fail. If under some keypair $(\texttt{pk}, \texttt{sk})$, a plaintext-ciphertext pair (m, c) is such that $c \xleftarrow{\$} \texttt{E}(\texttt{pk}, m)$ is obtained from encrypting m (probabilistically) but $m \neq \texttt{D}(\texttt{sk}, c)$, we call it a decryption failure. For probabilistic encryption routines where the coin is uniformly sampled from the coin space, we can quantify the probability that m triggers a decryption failure. From here, we can take the distribution of all keypairs and quantify the "correctness" of a (possibly imperfectly correct) encryption scheme. The following definition of δ-correctness is directly taken from [BDK+18].

Definition 1.1 (δ-correctness)**.** *A probabilistic public-key encryption scheme $\mathtt{PKE} = (\mathtt{KeyGen}, \mathtt{E}, \mathtt{D})$ is δ-correct if the expected maximal probability of decryption failure taken across the distribution of keypairs is at most δ:*

$$E\left[\max_{m \in \mathcal{M}} P\left[\mathtt{D}(\mathtt{sk}, \mathtt{E}(\mathtt{pk}, m))\right]\right] \leq \delta$$

where the expectation is taken over the distribution of keypairs and the probability is taken over the distribution of coins.

[HHK17] also defined an adversarial game in which the adversary's goal is to find some plaintext message to trigger a decryption failure. This adversarial game meaningfully models the real-world scenario in which decryption failure can reveal information about the secret key. Notice that when evaluating the win condition, the coin is uniformly random instead of being chosen by the adversary.

```
Algorithm 1 CORS
$(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \texttt{KeyGen}(1^\lambda)$
$m \xleftarrow{\$} A_{\texttt{CORS}}(1^\lambda, \texttt{pk}, \texttt{sk})$
return $[\![\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m)) \neq m]\!]$
```

Figure 1: The correctness game CORS

The definition of δ-correctness sets an explicit upper bound on the probability that any plaintext triggers decryption failure. This means that even if the `CORS` adversary actually finds the message m that is the most likely to trigger decryption failure, the probability of winning the `CORS` game is still upper-bounded by δ.

Lemma 1.0.1. *If `PKE` is δ-correct, then for all `CORS` adversaries A, even computationally unbounded ones, the probability of winning the `CORS` game is at most δ*

The values of δ for Kyber are taken directly from [ABD$^+$19]

security level	δ
Kyber512	2^{-139}
Kyber768	2^{-164}
Kyber1024	2^{-174}

Table 1: Concrete δ for Kyber

1.1.2 Security

An one-way adversary $A = (A_1, A_2)$ consists of two sub-routines A_1 and A_2. $s \overset{\$}{\leftarrow} A_1^{\mathcal{O}_1}(1^\lambda, \texttt{pk})$ takes the security parameter, some public-key, access to some oracle(s) $\mathcal{O}_1$, and outputs some intermediate state s. $\hat{m} \leftarrow A_2^{\mathcal{O}_2}(1^\lambda, \texttt{pk}, c^*)$ resumes from the output of A_1 and takes some challenge ciphertext, then tries to guess the corresponding decryption.

The advantage $\texttt{Adv}_{\texttt{OW-ATK}}(A)$ of an `OW-ATK` adversary is the probability that its guess is correct.

Definition 1.2. *A `PKE` is `OW-ATK` secure if for all efficient adversaries A, the advantage in the `OW-ATK` game is neligigble with respect to the security parameter:*

$$\texttt{Adv}_{\texttt{OW-ATK}}(A) \leq \text{negl}(\lambda)$$

An **indistinguishability** adversary $A = (A_1, A_2)$ similarly consists of two sub-routines. The first subroutine adversarially chooses two distinct plaintext messages, and the second sub-routine tries to distinguish which of the two plaintext messages is the decryptino of the challenge encryption. The advantage of an indistinguishability adversary is defined by $\texttt{Adv}_{\texttt{IND-ATK}}(A) = P[\hat{b} = b] - \frac{1}{2}$

Definition 1.3. *A `PKE` is `IND-ATK` secure if for all efficient adversaries A, the advantage in the indistinguishability game is negligible with respect to the security parameter*

$$\texttt{Adv}_{\texttt{IND-ATK}}(A) \leq \text{negl}(\lambda)$$

The security games are described in details in figure 2

Algorithm 2 `OW-ATK` game

```
$(\texttt{pk}, \texttt{sk}) \overset{\$}{\leftarrow} \texttt{KeyGen}(1^\lambda)$
$s \overset{\$}{\leftarrow} A_1^{\mathcal{O}_1}(1^\lambda, \texttt{pk})$
$m^* \overset{\$}{\leftarrow} \mathcal{M}$
$c^* \overset{\$}{\leftarrow} \texttt{E}(\texttt{pk}, m^*)$
$\hat{m} \leftarrow A_2^{\mathcal{O}_2}(1^\lambda, \texttt{pk}, s, c^*)$
return $[\![\hat{m} = m^*]\!]$
```

Algorithm 3 `IND-ATK` game

```
$(\texttt{pk}, \texttt{sk}) \overset{\$}{\leftarrow} \texttt{KeyGen}(1^\lambda)$
$(m_0, m_1) \overset{\$}{\leftarrow} A_1^{\mathcal{O}_1}(1^\lambda, \texttt{pk})$
$b \overset{\$}{\leftarrow} \{0, 1\}$
$c^* \overset{\$}{\leftarrow} \texttt{E}(\texttt{pk}, m_b)$
$\hat{b} \leftarrow A_2^{\mathcal{O}_2}(1^\lambda, \texttt{pk}, s, c^*)$
return $[\![\hat{b} = b]\!]$
```

Figure 2: The indistinguishability security game

The capabilities of the adveraries are modeled using different collections of oracles. In standard security requirements, adversaries with access to no additional oracles can only mount chosen plaintex attacks (CPA), adversaries with access to decryption oracle $\mathcal{O}^{\mathtt{D}}$ only before the receiving challenge ciphertext can mount non-adaptive chosen ciphertext attacks (CCA1), adversaries with access to decryption oracle both before and after receiving the challenge ciphertext can mount adaptive chosen ciphertext attacks (CCA2).

[HHK17] also defined two non-standard oracles and the corresponding attacks. The plaintext checking oracle $\mathtt{PCO}(m, c)$ returns 1 if m is a decryption of c and 0 otherwise. The ciphertext validation oracle $\mathtt{CVO}(c)$ returns 1 if c is a valid ciphertext and 0 otherwise.

Algorithm 4 $\mathtt{PCO}(m, c)$

```
  return $[\![\mathtt{D}(\mathtt{sk}, c) = m]\!]$
```

Algorithm 5 $\mathtt{CVO}(c)$

```
  return $[\![\mathtt{D}(\mathtt{sk}, c) \in \mathcal{M}]\!]$
```

Figure 3: PCO and CVO

Here is an overview of the various kinds of attacks and their associated oracles

ATK	$\mathcal{O}_1$	$\mathcal{O}_2$
CPA	–	
CCA1	$\mathcal{O}^{\mathtt{D}}$	-
CCA2	$\mathcal{O}^{\mathtt{D}}$	
PCVA	PCO, CVO	
PCA	PCO	
VA	CVO	

Table 2: Attacks and associated oracle access

1.1.3 Spread and rigidity

The spread of a public key encryption scheme measures the diffusion the encryption routine's output. The higher the spread, the lower the probability of obtaining any specific ciphertext.

Definition 1.4 (γ-spread)**.** *For a given keypair* $(\mathtt{pk}, \mathtt{sk})$ *and plaintext message* $m \in \mathcal{M}$*, the* min-entropy *of the encryption routine is:*

$$\text{min-entropy}(\mathtt{pk}, m) := -\log_2 \left(\max_{c \in \mathcal{C}} P\left[c = \mathtt{E}(\mathtt{pk}, m)\right] \right)$$

A PKE has γ-spread if for all keypairs $(\mathtt{pk}, \mathtt{sk})$ *and plaintext* $m \in \mathcal{M}$*:*

$$\text{min-entropy}(\mathtt{pk}, m) \leq \gamma$$

Having γ sperad means that for any keypair $(\mathtt{pk}, \mathtt{sk})$, plaintext m, and ciphertext c:

$$P\left[c = \mathtt{E}(\mathtt{pk}, m)\right] \leq 2^{-\lambda}$$

Finally, *rigidity* conveys the idea that a ciphertext cannot be perturbed without becoming either invalid or decrypting to another plaintext

Definition 1.5 (rigidity)**.** *$\mathtt{PKE}(\mathtt{KeyGen}, \mathtt{E}, \mathtt{D})$ is* rigid *if for all keypairs* $(\mathtt{pk}, \mathtt{sk})$ *and ciphertext c, either* $\mathtt{D}(\mathtt{sk}, c) = \perp$ *or* $\mathtt{E}(\mathtt{pk}, \mathtt{D}(\mathtt{sk}, c)) = c$

2 Modular Fujisaki-Okamoto transformation

The Fujisaki-Okamoto transformation [FO99] and its KEM variations [HHK17].

References

[ABD$^+$19] 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.

[BDK$^+$18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual international cryptology conference*, pages 537–554. Springer, 1999.

[HHK17] 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.