1 Preliminaries

1.1 Plaintext awareness

Plaintext awareness (PA)[BR95, BDPR98] describes the idea that no efficient algorithm can produce a valid ciphertext without knowing the corresponding decryption.

Let $\mathtt{PKE} = (\mathtt{KeyGen}, \mathtt{E}, \mathtt{D})$ be a public-key encryption scheme. Let H be a hash function. Let $\mathcal{E}^H_{\mathtt{pk}}$ be an encryption oracle that takes no argument and returns valid ciphertexts when queried. Under the random oracle model, hash queries made to the oracle $\mathcal{O}^H$ can be logged to a tape $\mathcal{L}^H = \{(h_i, H(h_i))\}$, and ciphertexts obtained from the encryption oracle are also logged to a separate tape $C = \{c : c \leftarrow \mathcal{E}^H_{\mathtt{pk}}(\cdot)\}$.

A plaintext-awareness adversary B is a probabilistic algorithm that is given some public key and access to the two oracles, then output some ciphertext c. Note that the encryption oracle here is not redundant because obtaining ciphertexts from the encryption oracle will not log any corresponding hash queries in the hash oracle. This models a PA adversary's ability to obtain valid ciphertexts without running the encryption routine, such as by eavesdropping. The PA adversary B outputs a ciphertext c and the transcript $\mathcal{L}^H, C$ of its interactions with the oracles.

Let K be some algorithm that outputs a decryption of c using the corresponding transcript $\mathcal{L}^H, C$. We restrict $c \notin C$ to prevent trivially turning K into a decryption oracle. The plaintext awareness game is defined in figure 1.

Algorithm 1 Plaintext awareness game

1: $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$
2: $(\mathcal{L}^H, C, c) \leftarrow B^{H, \mathcal{E}^H_{\mathtt{pk}}}(1^\lambda, \mathtt{pk})$
3: $m \leftarrow K(1^\lambda, \mathtt{pk}, \mathcal{L}^H, C, c)$
4: **return** $[\![c \notin C \wedge m = D(\mathtt{sk}, c)]\!]$

Figure 1: The plaintext awareness game

Definition 1.1. *A public-key encryption scheme is* plaintext-aware *if there exists an efficient K such that for all efficient PA adversaries B, the probability of K failing to extract the correct decryption is negligible.*

$$P[\mathit{PA(K,\ B)} \neq 1] \leq \mathit{negl}(\lambda)$$

Note that our definition of plaintext awareness deviates from [BDPR98] in that we do not require the PKE to also be IND-CPA secure. This is because in constructing a key encapsulation mechanism using the Fujisaki-Okamoto transformation, we only require the input PKE to be one-way secure, though under a stronger attack model. Restricting the definition to only the plaintext-awareness game allows us to combine PA with other security definition, such as OW-CPA, to prove stronger security result.

2 Encrypt-then-MAC transformations

Let $\mathtt{PKE}(\mathtt{KeyGen}, \mathtt{E}, \mathtt{D})$ be a probabilistic public-key encryption scheme defined over message space $\mathcal{M}_{\mathtt{PKE}}$, ciphertext space $\mathcal{C}_{\mathtt{PKE}}$, and coin space $\mathcal{R}_{\mathtt{PKE}}$. Where the encryption routine is deterministic, we simply set the coin space to contain a single element $\mathcal{R} = \{r\}$. Let $\mathtt{MAC}(\mathtt{Sign}, \mathtt{Verify})$ be a message authentication code defined over key space $\mathcal{K}_{\mathtt{MAC}}$. The message space of the MAC should contain the ciphertext space of the PKE: $\mathcal{C}_{\mathtt{PKE}} \subseteq \mathcal{M}_{\mathtt{MAC}}$. Let $G : \mathcal{M}_{\mathtt{PKE}} \to \mathcal{R}_{\mathtt{PKE}}$ and $H : \mathcal{M}_{\mathtt{PKE}} \to \mathcal{K}_{\mathtt{MAC}}$ be hash functions.

The "encrypt-then-MAC" transformation $\mathtt{PKE}_{\mathtt{EtM}}(\mathtt{KeyGen}, \mathtt{E}_{\mathtt{EtM}}, \mathtt{D}_{\mathtt{EtM}}) = T_{\mathtt{EtM}}(\mathtt{PKE}, \mathtt{MAC}, H)$ outputs a public-key encryption scheme where the key generation routine is identical to the input PKE's key generation routine. The de-randomized "encrypt-then-MAC" transformation $\mathtt{PKE}^{\$}_{\mathtt{EtM}}(\mathtt{KeyGen}, \mathtt{E}^{\$}_{\mathtt{EtM}}, \mathtt{D}^{\$}_{\mathtt{EtM}}) = T^{\$}_{\mathtt{EtM}}(\mathtt{PKE}, \mathtt{MAC}, G, H)$ similarly outputs a public-key encryption scheme. In both transformations, the key generation routine remains unchanged. The modified encryption and decryption routines are described in figure 2 and 3.

Algorithm 2 $\mathtt{E}_{\mathtt{EtM}}(\mathtt{pk}, m)$

$r \xleftarrow{\$} \mathcal{R}_{\mathtt{PKE}}$ ▷ If $\mathtt{E}$ is randomized, then $\mathtt{E}_{\mathtt{EtM}}$ is randomized
$k_{\mathtt{MAC}} \leftarrow H(m)$
$c \leftarrow \mathtt{E}(\mathtt{pk}, m; r)$
$t \leftarrow \mathtt{Sign}(k_{\mathtt{MAC}}, c)$
return (c, t)

Algorithm 3 $\mathtt{D}_{\mathtt{EtM}}(\mathtt{sk}, (c, t))$

$\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
$\hat{k}_{\mathtt{MAC}} \leftarrow G(\hat{m})$
if $\mathtt{Verify}(\hat{k}_{\mathtt{MAC}}, c, t) \neq 1$ **then**
    return $\perp$
end if
return $\hat{m}$

Figure 2: "encrypt-then-MAC" transformation

Algorithm 4 $\mathtt{E}^{\$}_{\mathtt{EtM}}(\mathtt{pk}, m)$

$k_{\mathtt{MAC}} \leftarrow H(m)$
$r \leftarrow G(m)$
$c \leftarrow \mathtt{E}(\mathtt{pk}, m; r)$
$t \leftarrow \mathtt{Sign}(k_{\mathtt{MAC}}, c)$
return (c, t)

Algorithm 5 $\mathtt{D}^{\$}_{\mathtt{EtM}}(\mathtt{sk}, (c, t))$

$\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
$\hat{k}_{\mathtt{MAC}} \leftarrow G(\hat{m})$
if $\mathtt{Verify}(\hat{k}_{\mathtt{MAC}}, c, t) \neq 1$ **then**
    return $\perp$
end if
return $\hat{m}$

Figure 3: de-randomized "encrypt-then-MAC" transformation

Theorem 2.1. *If the MAC is one-time unforgeable and PKE is OW-CPA secure, then EtM and EtM$^{\$}$ are both OW-CCA secure*

We will prove theorem 2.1 in two steps. First we show that EtM and EtM$^{\$}$ are both plaintext-aware in section 2.1, then we show that plaintext awareness and OW-CPA security imply OW-CCA security in section 2.2.

The KEM transformations from [HHK17] require the input PKE to be OW-PCVA secure, although OW-PCVA security is implied by OW-CCA security

Theorem 2.2. *For every OW-PCVA adversary $\mathcal{A}_{PCVA}$ against some PKE, there exists an OW-CCA adversary $\mathcal{A}_{CCA}$ with identical advantage:*

$$\epsilon_{OW\text{-}PCVA} = \epsilon_{OW\text{-}CCA}$$

Proof. The OW-CCA adversary can perfectly simulate both the PCO and the CVO by replacing "running decryption routine" with "querying the decryption oracle". When the OW-PCVA adversary halts, the OW-CCA adversary passes OW-PCVA adversary's output as its own. Thus, OW-CCA adversary wins if and only if OW-PCVA adversary wins. □

2.1 EtM is plaintext aware

2.2 Plaintext awareness and OW-CPA imply OW-CCA

In this section we show that if a PKE is both plaintext aware and OW-CPA secure, then it is OW-CCA secure.

Theorem 2.3. *For every OW-CCA adversary $\mathcal{A}_{OW\text{-}CCA}$ with advantage $\epsilon_{OW\text{-}CCA}$, there exists an ϵ_K knowledge extractor K and an OW-CPA adversary with advantage $\epsilon_{OW\text{-}CPA}$ such that*

$$\epsilon_{OW\text{-}CCA} \leq \epsilon_{OW\text{-}CPA} + q_D \cdot \epsilon_K$$

Proof. We will prove theorem 2.3 by constructing an OW-CPA adversary $\mathcal{A}_{\mathtt{OW\text{-}CPA}}$ that runs the OW-CCA adversary $\mathcal{A}_{\mathtt{OW\text{-}CCA}}$ as a sub-routine. $\mathcal{A}_{\mathtt{OW\text{-}CPA}}$ will answer $\mathcal{A}_{\mathtt{OW\text{-}CCA}}$'s decryption query using the knowledge extractor.

After receiving the security parameter 1^λ and public key pk, $\mathcal{A}_{\mathtt{OW\text{-}CPA}}$ initializes the hash tape $\mathcal{L}^H = \{\}$ to be an empty list, the runs $\mathcal{A}_{\mathtt{OW\text{-}CCA}}$ until the sub-routine is waiting for the challenge ciphertext. During this phase:

1. When $\mathcal{A}_{\texttt{OW-CCA}}$ makes a hash query h, $\mathcal{A}_{\texttt{OW-CPA}}$ queries $\mathcal{O}^H$ to obtain the hash value $H(h)$, appends $(h, H(h))$ to $\mathcal{L}^H$, then answers with $H(h)$

2. When $\mathcal{A}_{\texttt{OW-CCA}}$ makes a decryption query c, $\mathcal{A}_{\texttt{OW-CPA}}$ runs the knowledge extractor $m \leftarrow K(\mathcal{L}^H, C = \{\}, c)$, then answers with m

After receiving c^*, $\mathcal{A}_{\texttt{OW-CPA}}$ passes c^* to $\mathcal{A}_{\texttt{OW-CCA}}$, then continues simulating the `OW-CCA` game for the sub-routine.

1. Hash queries are answered in the same way as in the first phase

2. When $\mathcal{A}_{\texttt{OW-CCA}}$ makes a decryption query c:

   - If $c = c^*$, answer with $\perp$
   - If $c \neq c^*$, run the knowledge extractor $m \leftarrow K(\mathcal{L}^H, C = \{c^*\}, c)$, then answer with m

We borrow the argument from [BDPR98] to analyze how well $\mathcal{A}_{\texttt{OW-CPA}}$ simulates the decryption oracle. Let q_D be the total number of decryption queries made throughout both phases and let q_1 denote the number of decryption queries made in phase 1. We can thus label the decryption queries by $(c_1, c_2, \ldots, c_{q_1}, c_{q_1+1}, \ldots, c_{q_D})$. Each decryption query corresponds to a plaintext awareness adversary B_i playing an instance of the plaintext awareness game using $\mathcal{A}_{\texttt{OW-CCA}}$ as a sub-routine. Each B_i answers hash queries from $\mathcal{A}_{\texttt{OW-CCA}}$ in the same way as described above. For $j < i$, B_i answers the decryption query c_j using what K returns on B_j's output. When $\mathcal{A}_{\texttt{OW-CCA}}$ makes the i-th decryption query c_i, B_i outputs c_i and halts. For $1 \leq j \leq q_1$, there is no ciphertext obtained without running the encryption routine, so K is run with $C = \{\}$. For $q_1 < j \leq q_D$, $\mathcal{A}_{\texttt{OW-CCA}}$ has access to c^*, which is obtained without running the encryption routine, so K is run with $C = \{c^*\}$.

The simulated `OW-CCA` game deviates from the true `OW-CCA` game if and only if the knowledge extractor returns an incorrect decryption at one or more decryption queries. Since the probability of any one of decryption query being incorrect is bounded by ϵ_K, the probability that at least one decryption query being incorrect is bounded by $q_D \cdot \epsilon_K$:

$$\epsilon_{\texttt{OW-CCA}} - \epsilon_{\texttt{OW-CPA}} \leq q_D \cdot \epsilon_K$$

□

References

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18*, pages 26–45. Springer, 1998.

[BR95] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings 13*, pages 92–111. Springer, 1995.

[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.