Fast Fujisaki-Okamoto transformation using encrypt-then-mac and applications to Kyber

Anonymous Submission

Abstract. The modular Fujisaki-Okamoto (FO) transformation takes public-key encryption with weaker security and constructs a key encapsulation mechanism (KEM) with indistinguishability under adaptive chosen ciphertext attacks. While the modular FO transform enjoys tight security bound and quantum resistance, it also suffers from computational inefficiency due to using de-randomization and re-encryption for providing ciphertext integrity. In this work, we propose an alternative modular FO transformation that replaces re-encryption with a message authentication code (MAC) and prove the security bound of our construction. We then instantiate a concrete instance with ML-KEM and show that when re-encryption incurs significant computational cost, our construction provides substantial runtime speedup and reduced memory footprint.

Keywords: Key encapsulation mechanism, post-quantum cryptography, lattice cryptography, Fujisaki-Okamoto transformation

1 Introduction

The Fujisaki-Okamoto transformation [FO99] is a generic construction that takes cryptographic primitives of lesser security and constructs a public-key encryption scheme with indistinguishability under adaptive chosen ciphertext attacks. Later works extended the original transformation to the construction of key encapsulation mechanism, which has been adopted by many post-quantum schemes such as Kyber [BDK+18] (standardized by NIST into ML-KEM [KE23]).

The current state of the FO transformation enjoys tight security bound and quantum resistance [HHK17], but also leaves many open questions. One such problem is the use of re-encryption for providing ciphertext integrity [BP18], which requires the decryption/decapsulation to run the encryption routine as a subroutine. In many post-quantum schemes, such as Kyber, the encryption routine is substantially computationally more expensive than the decryption routine.

The problem of ciphertext integrity was solved in symmetric cryptography. Given a semantically secure symmetric cipher and an existentially unforgeable message authentication code, combining them using "encrypt-then-mac" provides authenticated encryption [BN00]. We took inspiration from this strategy and applied a similar technique to provide ciphertext integrity for a public-key encryption scheme, which then translates to an `IND-CCA` secure `KEM`. Using a message authentication code for ciphertext integrity replaces the re-encryption step in decryption with the computation of a tag, which should offer significant performance improvements while maintaining comparable level of security.

The main challenge in applying "encrypt-then-mac" to public-key cryptography is the lack of a pre-shared `MAC` key. We proposed to derive the shared `MAC` key by hashing the plaintext message. We will prove in section 3 that, under the random oracle model, the `MAC` key is securely hidden behind the hash function, and producing a valid pair of ciphertext and tag without full knowledge of the plaintext constitutes a forgery attack on the message authentication code. Thanks to the modular construction in [HHK17], providing ciphertext integrity in the underlying encryption scheme gives us an IND-CCA secure KEM for free.

In section 4.2, we instantiate concrete instances of our proposed transformation by modifying ML-KEM. We will demonstrate that, at the cost of small increase in encryption runtime and ciphertext size, our construction reduces both the runtime and memory footprint of the decryption routine.

2 Preliminaries and previous results

2.1 Public-key encryption scheme

We define a public key encryption scheme PKE to be a collection of three routines $(\mathtt{KeyGen}, \mathtt{E}, \mathtt{D})$ defined over a finite message space $\mathcal{M}$ and some ciphertext space $\mathcal{C}$. Many encryption routines are probabilistic, and we define their source of randomness to come from some coin space $\mathcal{R}$.

The encryption routine $\mathtt{E}(\mathtt{pk}, m)$ takes a public key, a plaintext message, and outputs a ciphertext $c \in \mathcal{C}$. Where the encryption routine is probabilistic, specifying a pseudorandom seed $r \in \mathcal{R}$ will make the encryption routine behave deterministically. The decryption routine $\mathtt{D}(\mathtt{sk}, c)$ takes a secret key, a ciphertext, and outputs the decryption $\hat{m}$ if the ciphertext is valid under the given secret key, or the rejection symbol $\perp$ if the ciphertext is invalid.

2.1.1 Correctness

It is common to require a PKE to be perfectly correct, meaning that for all possible keypairs $(\mathtt{pk}, \mathtt{sk})$ and plaintext messages $m \in \mathcal{M}$, $\mathtt{D}(\mathtt{sk}, \mathtt{E}(\mathtt{pk}, m)) = m$ at all times. However, some encryption schemes, including many popular lattice-based schemes, admit a non-zero probability of decryption failure: $\mathtt{D}(\mathtt{sk}, \mathtt{E}(\mathtt{pk}, m)) \neq m$. Furthermore, [HHK17] and [ABD+19] explained how decryption failure played a role in an adversary's advantage. In this paper, we inherit the definition for correctness from [HHK17]:

Definition 1 (δ-correctness)**.** A public key encryption scheme PKE is δ-correct if

$$\mathbf{E}[\max_{m \in \mathcal{M}} P[\mathtt{D}(\mathtt{sk}, c) \neq m \mid c \xleftarrow{\$} \mathtt{E}(\mathtt{pk}, m)]] \leq \delta$$

Where the expectation is taken over the probability distribution of keypairs $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathtt{KeyGen}()$

2.1.2 Security

We discuss the security of a PKE using the sequence of games described in [Sho04]. Specifically, we first define the OW-ATK and the IND-CPA game as they pertain to a public key encryption scheme. In later section we will define the IND-CCA game as it pertains to a key encapsulation mechanism.

In the OW-ATK game, an adversary's goal is to recover the decryption of a randomly generated ciphertext.

The adversary $\mathcal{A}$ with access to oracle(s) $\mathcal{O}_{\mathtt{ATK}}$ wins the game if its guess $\hat{m}$ is equal to the challenge plaintext m^*. The *advantage* $\epsilon_{\mathtt{OW\text{-}ATK}}$ of an adversary in this game is the probability that it wins the game.

The choice of oracle(s) $\mathcal{O}_{\mathtt{ATK}}$ depends on the choice of ATK. Specifically:

Algorithm 1 The OW-ATK game

1: $(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \texttt{KeyGen}(1^\lambda)$
2: $m^* \xleftarrow{\$} \mathcal{M}$
3: $c^* \xleftarrow{\$} \texttt{E}(\texttt{pk}, m)^*$
4: $\hat{m} \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\texttt{ATK}}}(1^\lambda, \texttt{pk}, c^*)$
5: **return** $[\![m^* = \hat{m}]\!]$

Figure 1: The OW-ATK game

Algorithm 2 $\texttt{PCO}(m \in \mathcal{M}, c \in \mathcal{C})$

1: **return** $[\![\texttt{D}(\texttt{sk}, c) = m]\!]$

Figure 2: The Plaintext-Checking Oracle PCO

Algorithm 3 $\texttt{CVO}(c \in \mathcal{C})$

1: **return** $[\![\texttt{D}(\texttt{sk}, c) \in \mathcal{M}]\!]$

Figure 3: the Ciphertext-Validation Oracle CVO

$$\mathcal{O}_{\texttt{ATK}} = \begin{cases} - & \texttt{ATK} = \texttt{CPA} \\ \texttt{PCO} & \texttt{ATK} = \texttt{PCA} \\ \texttt{CVO} & \texttt{ATK} = \texttt{VA} \\ \texttt{PCO, CVO} & \texttt{ATK} = \texttt{PCVA} \end{cases}$$

Where the definitions of plaintext-checking oracle PCO and the ciphertext-validation oracle CVO are inherited from [HHK17]

In the IND-CPA game, an adversary's goal is to distinguish the encryption of one message from the encryption of another message. Given the public key, the adversary outputs two adversarially chosen messages and obtains the encryption of a random choice between these two messages. The adversary wins the IND-CPA game if it correctly identifies which message the encryption is obtained from.

Algorithm 4 The IND-CPA game

1: $(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \texttt{KeyGen}(1^\lambda)$
2: $(m_0, m_1) \xleftarrow{\$} \mathcal{A}(a^\lambda, \texttt{pk})$
3: $b \xleftarrow{\$} \{0, 1\}$
4: $c^* \xleftarrow{\$} \texttt{E}(\texttt{pk}, m_b)$
5: $\hat{b} \xleftarrow{\$} \mathcal{A}(1^\lambda, \texttt{pk}, c^*)$
6: **return** $[\![b = \hat{b}]\!]$

The *advantage* $\epsilon_{\texttt{IND-CPA}}$ of an IND-CPA adversary $\mathcal{A}$ is defined by

$$\epsilon_{\texttt{IND-CPA}} = \left| P[\hat{b} = b] - \frac{1}{2} \right|$$

2.2 Key encapsulation mechanism

A key encapsulation mechanism KEM is a collection of three routines ($\texttt{KeyGen}, \texttt{Encap}, \texttt{Decap}$) defined over some ciphertext space $\mathcal{C}$ and some key space $\mathcal{K}$. The key generation routine takes the security parameter 1^λ and outputs a keypair $(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \texttt{KeyGen}(1^\lambda)$. $\texttt{Encap}(\texttt{pk})$ is a probabilistic routine that takes a public key pk and outputs a pair of values (c, K) where $c \in \mathcal{C}$ is the encapsulation (or ciphertext) of the shared secret $k \in \mathcal{K}$. $\texttt{Decap}(\texttt{sk}, c)$ is a deterministic routine that takes the secret key sk and the encapsulation c and returns the shared secret k if the ciphertext is valid, or the rejection symbol $\perp$ if the ciphertext is invalid.

The IND-CCA security of a KEM is defined by an adversarial game in which an adversary's goal is to distinguish pseudorandom shared secret (generated by running the Encap routine) and a truly random value.

Algorithm 5 IND-CCA game for KEM

1: $(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \texttt{KeyGen}(1^\lambda)$
2: $(c^*, k_0) \xleftarrow{\$} \texttt{Encap}(\texttt{pk})$
3: $k_1 \xleftarrow{\$} \mathcal{K}$
4: $b \xleftarrow{\$} \{0, 1\}$
5: $\hat{b} \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\texttt{Decap}}}_{\texttt{IND-CCA}}(1^\lambda, \texttt{pk}, c^*, k_b)$
6: **return** $[\![\hat{b} = b]\!]$

The decapsulation oracle $\mathcal{O}^{\texttt{Decap}}$ takes a ciphertext c and returns the output of the Decap routine using the secret key. The advantage $\epsilon_{\texttt{IND-CCA}}$ of an IND-CCA adversary $\mathcal{A}_{\texttt{IND-CCA}}$ is defined by

$$\epsilon_{\texttt{IND-CCA}} = \left| P[\hat{b} = b] - \frac{1}{2} \right|$$

2.3 Message authentication code

A message authentication code MAC is a collection of routines ($\texttt{MAC}, \texttt{MAC.Verify}$) defined over some key space $\mathcal{K}$, some message space $\mathcal{M}$, and some tag space $\mathcal{T}$. The signing routine $\texttt{MAC}(k, m)$ takes the secret key $k \in \mathcal{K}$ and some message, and outputs a tag t. The verification routine $\texttt{MAC.Verify}(k, m, t)$ takes the triplet of secret key, message, and tag, and outputs 1 if the message-tag pair is valid under the secret key, or 0 otherwise.

The security of a MAC is defined in an adversarial game in which an adversary, with access to some signing oracle $\mathcal{O}_{\texttt{MAC}}(m)$, tries to forge a new valid message-tag pair that has never been queried before. The existential unforgeability under chosen message attack (EUF-CMA) game is shown below:

Algorithm 6 The EUF-CMA game

1: $k^* \xleftarrow{\$} \mathcal{K}$
2: $(\hat{m}, \hat{t}) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\texttt{MAC}}}()$
3: **return** $[\![\texttt{MAC.Verify}(k^*, \hat{m}, \hat{t}) \texttt{ and } (\hat{m}, \hat{t}) \notin \mathcal{O}_{\texttt{MAC}}]\!]$

The advantage $\epsilon_{\texttt{EUF-CMA}}$ of the existential forgery adversary is the probability that it wins the EUF-CMA game.

2.4 Modular Fujisaki-Okamoto transformation

The Fujisaki-Okamoto transformation (FOT) [FO99] is a generic transformation that takes a $\texttt{PKE}$ with weaker security (such as $\texttt{OW-CPA}$ or $\texttt{IND-CPA}$) and outputs a $\texttt{PKE}$ with stronger security. A later variation [HHK17] improved the original construction in [FO99] by accounting for decryption failures, tightening security bounds, and providing a modular construction that first transforms $\texttt{OW-CPA/IND-CPA PKE}$ into $\texttt{OW-PCVA PKE}$ by providing ciphertext integrity through re-encryption (the $\texttt{T}$ transformation), then transforming the $\texttt{OW-PCVA PKE}$ into an $\texttt{IND-CCA KEM}$ (the $\texttt{U}$ transformation).

Particularly relevant to our results are two variations of the $\texttt{U}$ transformation: $\texttt{U}^{\perp}$ (KEM with explicit rejection) and $\texttt{U}^{\not\perp}$ (KEM with implicit rejection). If $\texttt{PKE}$ is $\texttt{OW-PCVA}$ secure, then $\texttt{U}^{\perp}$ transforms $\texttt{PKE}$ into an $\texttt{IND-CCA}$ secure $\texttt{KEM}^{\perp}$:

Theorem 1. *For any $\texttt{IND-CCA}$ adversary $\mathcal{A}_{\texttt{KEM}}$ against $\texttt{KEM}^{\perp}$ with advantage $\epsilon_{\texttt{KEM}}$ issuing at most q_D decapsulation queries and at most q_H hash queries, there exists an $\texttt{OW-PCVA}$ adversary $\mathcal{A}_{\texttt{PKE}}$ against the underlying $\texttt{PKE}$ with advantage $\epsilon_{\texttt{PKE}}$ that makes at most q_H queries to $\texttt{PCO}$ and $\texttt{CVO}$ such that*

$$\epsilon_{\texttt{KEM}} \leq \epsilon_{\texttt{PKE}}$$

Similarly, if $\texttt{PKE}$ is $\texttt{OW-PCA}$ secure, then $\texttt{U}^{\not\perp}$ transforms $\texttt{PKE}$ into an $\texttt{IND-CCA}$ secure $\texttt{KEM}^{\not\perp}$

Theorem 2. *For any $\texttt{IND-CCA}$ adversary $\mathcal{A}_{\texttt{KEM}}$ against $\texttt{KEM}^{\not\perp}$ with advantage $\epsilon_{\texttt{KEM}}$ issuing at most q_D decapsulation queries and at most q_H hash queries, there exists an $\texttt{OW-CPA}$ adversary $\mathcal{A}_{\texttt{PKE}}$ against the underlying $\texttt{PKE}$ with advantage $\epsilon_{\texttt{PKE}}$ issuing at most q_H queries to $\texttt{PCO}$ such that:*

$$\epsilon_{\texttt{KEM}} \leq \frac{q_H}{|\mathcal{M}_{\texttt{PKE}}|} + \epsilon_{\texttt{PKE}}$$

The modularity of the $\texttt{T}$ and $\texttt{U}$ transformation allows us to tweak only the $\texttt{T}$ transformation (see section 3), obtain $\texttt{OW-PCVA}$ security, then automatically get $\texttt{IND-CCA}$ security for free. This means that we can directly apply our contribution to existing KEM's already using this modular transformation, such as ML-KEM [KE23], and obtain performance improvements while maintaining comparable levels of security (see section 4.2).

3 FO transform with encrypt-then-mac for achieving ciphertext integrity

In this section, we present two different encrypt-then-mac modes using symmetric-key approaches to achieve ciphertext integrity, which result in two different fast FOTs. The first method is to apply an MAC over ciphertext [???], referred to as *standard Enc-then-MAC*, and the second method is adopted from GCM [??] mode in authenticated encryption (AE) using polynomial hash, referred to as *poly Enc-then-MAC*. In other words, we will replace the public-key re-encryption in FOT by symmetric-key MAC approaches, and the resulting transform is referred to as *FOT+*.

3.1 FOT+ in standard encrypt-then-MAC

Let $\texttt{PKE}(\texttt{KeyGen}, \texttt{E}, \texttt{D})$ be a probabilistic public-key encryption scheme defined over message space $\mathcal{M}_{\texttt{PKE}}$, ciphertext space $\mathcal{C}$, and coin space $\mathcal{R}$. Let $\texttt{MAC}$ be a deterministic and perfectly correct message authentication code defined over key space $\mathcal{K}_{\texttt{MAC}}$, message space $\mathcal{M}_{\texttt{MAC}}$, and tag space $\mathcal{T}_{\texttt{MAC}}$. Let $G : \mathcal{M}_{\texttt{PKE}} \to \mathcal{R} \times \mathcal{K}_{\texttt{MAC}}$ be a hash function that hashes a plaintext message into a pseudorandom coin and a $\texttt{MAC}$ key. The $\texttt{T}_{\texttt{EtM}}[\texttt{PKE}, \texttt{MAC}, G]$ transformation takes the input PKE, MAC, and hash function G, and outputs a public-key encryption scheme $\texttt{PKE}_{\texttt{EtM}}(\texttt{KeyGen}, \texttt{E}_{\texttt{EtM}}, \texttt{D}_{\texttt{EtM}})$ where as the key generation routine remains unchanged, and the encryption/decryption routines are as follows:

Algorithm 7 $\texttt{E}_{\texttt{EtM}}$($\texttt{pk}$, m)

$(r, k) \leftarrow G(m)$
$c \leftarrow \texttt{E}(\texttt{pk}, m; r)$
$t \leftarrow \texttt{MAC}(k, c)$
return (c, t)

Algorithm 8 $\texttt{D}_{\texttt{EtM}}$($\texttt{sk}$, (c, t))

$\hat{m} \leftarrow \texttt{D}(\texttt{sk}, c)$
$(\hat{r}, \hat{k}) \leftarrow G(m)$
if $\texttt{MAC.Verify}(\hat{k}, c, t) = 0$ **then**
  return $\perp$
end if
return $\hat{m}$

We claim that if the input PKE is OW-CPA secure and MAC is existentially unforgeable, then under the random oracle model, $\texttt{PKE}_{\texttt{EtM}}$ is OW-PCVA secure with non-tight security reduction.

Theorem 3. *If PKE is δ-correct, then $\texttt{PKE}_{\texttt{EtM}}$ is δ-correct. In addition, for every OW-PCVA adversary $\mathcal{A}_{\texttt{OW-PCVA}}$ against $\texttt{PKE}_{\texttt{EtM}}$ that makes q_P PCO queries, q_V CVO queries, q_G hash queries to G, and that has advantage $\epsilon_{\texttt{OW-PCVA}}$ there exists an existential forgery adversary $\mathcal{A}_{\texttt{MAC}}$ against the underlying MAC with advantage $\epsilon_{\texttt{MAC}}$ and some OW-CPA adversary $\mathcal{A}_{\texttt{OW-CPA}}$ against the underlying PKE with advantage $\epsilon_{\texttt{OW-CPA}}$ such that*

$$\epsilon_{\texttt{OW-PCVA}} \leq (q_G + q_P) \cdot \delta + q_V \cdot \epsilon_{\texttt{MAC}} + (q_G + q_P + 1) \cdot \epsilon_{\texttt{OW-CPA}}$$

Furthermore, if the input PKE is additionally IND-CPA secure, then $\texttt{PKE}_{\texttt{EtM}}$ is OW-PCVA secure with tight security reduction.

Corollary 1. *For every OW-PCVA adversary $\mathcal{A}_{\texttt{OW-PCVA}}$ against $\texttt{PKE}_{\texttt{EtM}}$ that makes q_P PCO queries, q_V CVO queries, q_G hash queries to G, and that has advantage $\epsilon_{\texttt{OW-PCVA}}$ there exists an existential forgery adversary $\mathcal{A}_{\texttt{MAC}}$ against the underlying MAC with advantage $\epsilon_{\texttt{MAC}}$ and some IND-CPA adversary $\mathcal{A}_{\texttt{IND-CPA}}$ against the underlying PKE with advantage $\epsilon_{\texttt{IND-CPA}}$ such that*

$$\epsilon_{\texttt{OW-PCVA}} \leq (q_G + q_P) \cdot \delta + q_V \cdot \epsilon_{\texttt{MAC}} + \frac{1 + 2q_G}{|\mathcal{M}_{\texttt{PKE}}|} + 3\epsilon_{\texttt{IND-CPA}}$$

Proof. Since no modification was made to the internals of the input PKE, the correctness of the transformed scheme is trivially identical to the correctness of the input scheme.

We will prove the security claim using a sequence of games [Sho04], then prove the security claim in the corollary 1 by making a few modifications. This proof borrows heavily from the proof presented in [HHK17].

Game 0 is the OW-PCVA game. Let ϵ_0 denote the OW-PCVA adversary's advantage in Game 0, then $\epsilon_0 = \epsilon_{\texttt{OW-PCVA}}$

Game 1 is identical to Game 0, except PCO is replaced with $\texttt{PCO}_1$. Because $\texttt{E}_{\texttt{EtM}}$ is a deterministic encryption routine, the two games differ from the adversary's perspective if and only if any of the PCO query $(m, (c, t))$ causes decryption failure $\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m; r)) \neq m$.

Algorithm 9 Sequence of games

1. $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$
2. $m^* \xleftarrow{\$} \mathcal{M}_{\mathtt{PKE}}$
3. $(r^*, k^*) \leftarrow G(m^*)$ ▷ Game 0-2
4. $r^* \xleftarrow{\$} \mathcal{R}, k^* \xleftarrow{\$} \mathcal{K}_{\mathtt{MAC}}$ ▷ Game 3
5. $c^* \leftarrow \mathtt{E}(\mathtt{pk}, m^*; r^*)$
6. $t^* \leftarrow \mathtt{MAC}(k^*, c^*)$
7. $\hat{m} \leftarrow \mathcal{A}_{\mathtt{OW\text{-}PCVA}}^{\mathcal{O}^G, \mathtt{PCO}, \mathtt{CVO}}(1^\lambda, \mathtt{pk}, (c^*, t^*))$ ▷ Game 0
8. $\hat{m} \leftarrow \mathcal{A}_{\mathtt{OW\text{-}PCVA}}^{\mathcal{O}^G, \mathtt{PCO}_1, \mathtt{CVO}}(1^\lambda, \mathtt{pk}, (c^*, t^*))$ ▷ Game 1
9. $\hat{m} \leftarrow \mathcal{A}_{\mathtt{OW\text{-}PCVA}}^{\mathcal{O}^G, \mathtt{PCO}_1, \mathtt{CVO}_1}(1^\lambda, \mathtt{pk}, (c^*, t^*))$ ▷ Game 2-3
10. **return** $[\![\hat{m} = m^*]\!]$

Figure 4: Sequence of `Game 0` to `Game 3` in proof of theorem 3

Algorithm 10 $\mathtt{PCO}(m, (c, t))$

1. $\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
2. $(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$
3. **if** $\mathtt{MAC.Verify}(\hat{k}, c, t) = 0$ **then**
4.   **return** 0
5. **end if**
6. **return** $[\![\hat{m} = m]\!]$

Algorithm 11 $\mathtt{PCO}_1(m, (c, t))$

1. $(r, k) \leftarrow G(\hat{m})$
2. **return** $[\![\mathtt{E}(\mathtt{pk}, m; r) = c]\!]$ `and` $[\![\mathtt{MAC.Verify}(k, c, t) = 1]\!]$

Algorithm 12 $\mathtt{CVO}(c, t)$

1. $\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
2. $(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$
3. **return** $\mathtt{MAC.Verify}(\hat{k}, c, t)$

Algorithm 13 $\mathtt{CVO}_1(c, t)$

1. **if** $\exists(\tilde{m}, \tilde{r}, \tilde{k}) \in \mathcal{O}^G$ such that $\mathtt{E}(\mathtt{pk}, \tilde{m}; \tilde{r}) = c$ and $\mathtt{MAC.Verify}(\tilde{k}, c, t)$ **then**
2.   **return** 1
3. **end if**
4. **return** 0

The probability of decryption failure for any single PCO query is bounded by δ, so the overall probability of having at least one query causing decryption failure is at most $q_P \cdot \delta$. Let ϵ_0 and ϵ_1 respectively denote $\mathcal{A}_{\texttt{OW-PCVA}}$'s advantage in Game 0 and Game 1 respectively, then by the difference lemma [Sho04]:

$$\epsilon_0 - \epsilon_1 \leq q_P \cdot \delta \tag{1}$$

Game 2 is identical to Game 1 except CVO is replaced with $\texttt{CVO}_1$. $\texttt{CVO}_1$ replaces the decryption routine with the deterministic encryption routine and the MAC key derivation with "searching through hash oracle records". Therefore, there are exactly two scenarios in which Game 2 differ from Game 1 from the adversary's perspective.

In the first scenario, the queried ciphertext (c, t) has a matching hash query $(\tilde{m}, \tilde{r}, \tilde{k})$, but $(\tilde{m}, \tilde{r})$ causes decryption failure: $\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m; r)) \neq m$. For each hash query, the probability that $(\tilde{m}, \tilde{r})$ causes decryption failure is bounded by δ, so the probability of having at least one such hash query is at most $q_G \cdot \delta$.

In the second scenario, the queried ciphertext (c, t) has no matching hash query. Under the random oracle model, this means that the MAC key k used to sign c is an unknown and uniformly random key from the adversary's perspective. In other words, (c, t) is an existential forgery. The probability of producing a single forgery is bounded by the advantage of a MAC adversary, so the probability of having at least one dishonest CVO query is at most $q_V \cdot \epsilon_{\texttt{MAC}}$.

Denote the OW-PCVA adversary's advantage in game 2 by ϵ_2, then by the difference lemma:

$$\epsilon_1 - \epsilon_2 \leq q_G \cdot \delta + q_V \cdot \epsilon_{\texttt{MAC}} \tag{2}$$

In Game 3, the challenge encryption routine is modified. Instead of pseudorandomly deriving the coin r^* and the MAC key k^* from G, the coin and the MAC key are uniformly sampled from their respective domain. Under the random oracle model, Game 3 and Game 2 are indistinguishable from the adversary's perspective unless the adversary queries G or PCO with the value m^*. Denote the probability of "adversary querying G or PCO with m^*" by $P[\texttt{QUERY}^*]$, and the adversary's advantage in Game 3 by ϵ_3, then by the difference lemma:

$$\epsilon_2 - \epsilon_3 \leq P[\texttt{QUERY}^*] \tag{3}$$

A standard OW-CPA adversary against the underlying PKE can simulate Game 3 for an OW-PCVA adversary, since $\texttt{PCO}_1$ and $\texttt{CVO}_1$ only make use of the public key pk and the hash oracle $\mathcal{O}^G$, the challenge encryption c^* is obtained using a truly random coin, and the MAC key can be uniformly sampled. After the OW-PCVA adversary outputs a guess, the OW-CPA adversary can simply pass OW-PCVA's output. The OW-CPA adversary wins if and only if the OW-CPA adversary wins Game 3:

$$\epsilon_3 = \epsilon_{\texttt{OW-CPA}} \tag{4}$$

We can construct another OW-CPA adversary that simulates Game 3 for an OW-PCVA adversary. After the OW-PCVA adversary halts, this OW-CPA adversary picks and outputs a random value $\tilde{m}$ from all possible values recorded on the tape of the hash oracle $\mathcal{O}^G$ and $\texttt{PCO}_1$. If $\texttt{QUERY}^*$ occurs, then the OW-CPA adversary wins the game with probability $\frac{1}{q_G + q_P}$. In other words:

$$\epsilon_{\texttt{OW-CPA}} = \frac{1}{q_G + q_P} \cdot P[\texttt{QUERY}^*] \tag{5}$$

Combining equations 1, 2, 3, 4, and 5 gives the security bound in theorem 3.

We make two modifications to prove corollary 1. First, we invoke a well-known result that the IND-CPA security of a PKE with sufficiently large message space implies its OW-CPA security.

Lemma 1. *For every OW-CPA adversary with advantage $\epsilon_{OW\text{-}CPA}$ there exists an IND-CPA adversary with advantage $\epsilon_{IND\text{-}CPA}$ such that*

$$\epsilon_{OW\text{-}CPA} \leq \frac{1}{|\mathcal{M}_{PKE}|} + \epsilon_{IND\text{-}CPA} \tag{6}$$

Second, we borrow results from [HHK17] and construct an IND-CPA adversary to bound $P[\mathtt{QUERY}^*]$:

$$\frac{1}{2}P[\mathtt{QUERY}^*] \leq \epsilon_{\mathtt{IND\text{-}CPA}} + \frac{q_G}{|\mathcal{M}_{\mathtt{PKE}}|} \tag{7}$$

Combining equations 1, 2, 3, 4, 6, and 7 into theorem 3 completes the proof of the corollary. □

3.2 FOT+in poly Enc-then-MAC

4 Applications to Kyber

4.1 Kyber using FOT+

4.2 Performance comparisons

Comparisons will do for Kyber and Kyber$^+$, define it.

4.3 Experimental results

5 Conclusions and future works

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.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 531–545. Springer, 2000.

[BP18] Daniel J Bernstein and Edoardo Persichetti. Towards kem unification. *Cryptology ePrint Archive*, 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.

[KE23] NIST Module-Lattice-Based Key-Encapsulation. Mechanism standard. *NIST Post-Quantum Cryptography Standardization Process; NIST: Gaithersburg, MD, USA*, 2023.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *cryptology eprint archive*, 2004.