Fast, memory-efficient IND-CCA KEM using encrypt-then-mac

Ganyu Xu[1], Guang Gong[1] and Kalikinkar Mandal[2]

[1] University of Waterloo, Waterloo, Canada, g66xu@uwaterloo.ca
[2] University of Waterloo, Waterloo, Canada, ggong@uwaterloo.ca
[3] University of New Brunswick, New Brunswick, Canada, kmandal@unb.ca

Abstract.

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

1 Introduction

2 Preliminaries and previous results

2.1 Public-key encryption scheme

We define a public key encryption scheme $\texttt{PKE}$ to be a collection of three routines $(\texttt{KeyGen}, \texttt{E}, \texttt{D})$ 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 $\texttt{E}(\texttt{pk}, m)$ 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 $\texttt{D}(\texttt{sk}, c)$ 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 $\texttt{PKE}$ to be perfectly correct, meaning that for all possible keypairs $(\texttt{pk}, \texttt{sk})$ and plaintext messages $m \in \mathcal{M}$, $\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m)) = m$ at all times. However, some encryption schemes, including many popular lattice-based schemes, admit a non-zero probability of decryption failure: $\texttt{D}(\texttt{sk}, \texttt{E}(\texttt{pk}, m)) \neq m$. 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 $\texttt{PKE}$ is δ-correct if

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

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

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.

Algorithm 1 The OW-ATK game

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

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:

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

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

Algorithm 2 Plaintext checking oracle PCO

Require: $(m \in \mathcal{M}, c \in \mathcal{C})$
return $[\![\mathtt{D}(\mathtt{sk}, c) = m]\!]$

Algorithm 3 Ciphertext validation oracle CVO

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

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.

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

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

2.2 Key encapsulation mechanism

A key encapsulation mechanism KEM is the collection of three routines (KeyGen, Encap, Decap) defined over some ciphertext space $\mathcal{C}$ and some key space $\mathcal{K}$. The key generation routine

Algorithm 4 The IND-CPA game

$(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$
$(m_0, m_1) \xleftarrow{\$} \mathcal{A}(a^\lambda, \mathtt{pk})$
$b \xleftarrow{\$} \{0, 1\}$
$c^* \xleftarrow{\$} \mathtt{E}(\mathtt{pk}, m_b)$
$\hat{b} \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathtt{pk}, c^*)$
return $[\![b = \hat{b}]\!]$

takes the security parameter 1^λ and outputs a keypair $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$. $\mathtt{Encap}(\mathtt{pk})$ is a probabilistic routine that takes a public key $\mathtt{pk}$ and outputs a pair of values (c, K) where $c \in \mathcal{C}$ is the encapsulation (or ciphertext) of the shared secret $k \in \mathcal{K}$. $\mathtt{Decap}(\mathtt{sk}, c)$ is a deterministic routine that takes the secret key $\mathtt{sk}$ 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

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

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

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

2.3 Message authentication code

A message authentication code MAC is a collection of routines $(\mathtt{S}, \mathtt{V})$ defined over some key space $\mathcal{K}$, some message space $\mathcal{M}$, and some tag space $\mathcal{T}$. The signing routine $\mathtt{S}(k, m)$ takes the secret key $k \in \mathcal{K}$ and some message, and outputs a tag t. The verification routine $\mathtt{V}(k, m, t)$ 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 $\mathcal{O}_{\mathtt{S}}(m)$, 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:

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

Algorithm 6 The EUF-CMA game

1: $k^* \xleftarrow{\$} \mathcal{K}$
2: $(\hat{m}, \hat{t}) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_S}()$
3: **return** $[\![\mathtt{V}(k^*, \hat{m}, \hat{t}) \ \mathtt{and} \ (\hat{m}, \hat{t}) \notin \mathcal{O}_S]\!]$

2.4 Modular Fujisaki-Okamoto transformation

The Fujisaki-Okamoto transformation [FO99] is a generic transformation that takes a PKE with weaker security (such as OW-CPA or IND-CPA) and outputs a 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 OW-CPA/IND-CPA PKE into OW-PCVA PKE by providing ciphertext integrity through re-encryption (the T transformation), then transforming the OW-PCVA PKE into an IND-CCA KEM (the U transformation).

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

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

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

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

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

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

The modularity of the T and U transformation allows us to tweak only the T transformation (see section 3), obtain OW-PCVA security, then automatically get 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).

3 Achieving ciphertext integrity using encrypt-then-mac

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

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

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

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

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

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

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

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

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

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

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

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 $\epsilon_0 = \epsilon_{\mathtt{OW\text{-}PCVA}}$

Game 1 is identical to Game 0, except PCO is replaced with $\mathtt{PCO}_1$. Because $\mathtt{E}_{\mathtt{EtM}}$ 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 $\mathtt{D}(\mathtt{sk}, \mathtt{E}(\mathtt{pk}, m; r)) \neq m$. 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 $\mathcal{A}_{\mathtt{OW\text{-}PCVA}}$'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 $\mathtt{CVO}_1$. $\mathtt{CVO}_1$ replaces the decryption routine with the deterministic encryption routine and the MAC key derivation

Algorithm 9 Sequence of games

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

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

$\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
$(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$
if $\mathtt{V}(\hat{k}, c, t) = 0$ **then**
    return 0
end if
return $[\![\hat{m} = m]\!]$

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

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

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

$\hat{m} \leftarrow \mathtt{D}(\mathtt{sk}, c)$
$(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$
return $\mathtt{V}(\hat{k}, c, t)$

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

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

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) as a matching hash query $(\tilde{m}, \tilde{r}, \tilde{k})$, but $(\tilde{m}, \tilde{r})$ causes decryption failure: $\mathtt{D}(\mathtt{sk}, \mathtt{E}(\mathtt{pk}, m; r)) \neq m$. 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 $q_V \cdot \epsilon_{\mathtt{MAC}}$.

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

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

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 $P[\mathtt{QUERY}^*]$, and the adversary's advantage in `Game 3` by ϵ_3, then by the difference lemma:

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

A standard `OW-CPA` adversary against the underlying `PKE` can simulate `Game 3` for an `OW-PCVA` adversary, since $\mathtt{PCO}_1$ and $\mathtt{CVO}_1$ 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`:

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

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 $\mathtt{PCO}_1$. If $\mathtt{QUERY}^*$ occurs, then the `OW-CPA` adversary wins the game with probability $\frac{1}{q_G + q_P}$. In other words:

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

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 $\epsilon_{\mathtt{OW\text{-}CPA}}$ there exists an `IND-CPA` adversary with advantage $\epsilon_{\mathtt{IND\text{-}CPA}}$ such that*

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

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

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

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

4 Experimental results

5 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.

[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.