Faster generic CCA secure KEM transformation using encrypt-then-MAC

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

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

Abstract. TODO: write abstract later

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

Key encapsulation mechanism (KEM) is a public-key cryptographic primitive that allows two parties to establish a shared secret over an insecure communication channel. The accepted security requirement of a KEM is *Indistinguishability under adaptive chosen ciphertext attack (IND-CCA)*. Intuitively speaking, IND-CCA security implies that no efficient adversary (usually defined as probabilistic polynomial time Turing machine) can distinguish a pseudorandom shared secret from a uniformly random bit string of identical length even with access to a decapsulation oracle. Unfortunately, CCA security is difficult to achieve from scratch. Early attempts at constructing CCA secure public-key cryptosystems using only heuristics argument and without using formal proof, such as RSA encryption in PKCS #1 [30] and RSA signature ISO 9796 [1], were badly broken with sophisticated cryptanalysis [12,15,17]. Afterwards, provable chosen ciphertext security became a necessity for new cryptographic protocols. There have been many provable CCA secure constructions since then. Notable examples include Optimal Asymmetric Encryption Padding (OAEP) [7], which is combined with RSA [21] into the widely adopted RSA-OAEP. The Fujisaki-Okamoto transformation [20,23] is another generic CCA secure transformation that was thoroughly studied and widely adopted, particularly by many KEM candidates in NIST's Post Quantum Cryptography (PQC) standardization project.

Chosen ciphertext security is a solved problem within the context of symmetric cryptography. It is well understood that authenticated encryption can be achieved by combining a semantically secure symmetric encryption scheme with an existentially unforgeable message authentication code (MAC) using either the "encrypt-then-MAC" (AES-GCM, ChaCha20-Poly1305) or "MAC-then-encrypt" pattern (AES-CCM)[6,27]. However, adapting this technique for public-key cryptosystems is challenging, since the two communicating parties do not have a pre-shared symmetric key. The first attempt at such adaption is the Diffie-Hellman integrated encryption scheme (DHIES) [3,4] proposed by Abdalla, Bellare, and Rogaway, who proved its chosen ciphertext security under a non-standard but

well studied assumption called "Gap Diffie-Hellman problem" [28]. DHIES and its variations appeared in international standards such as IEEE P1363a[2] and ANSI X9.63[5].

1.1 Our contributions

Our contributions are as follows:

Generic CCA secure KEM transformation. We propose the "encrypt-then-MAC" KEM transformation. Our transformation constructs a KEM with provable CCA security under the random oracle model using a public-key encryption scheme (PKE) with one-wayness under plaintext-checking attack and a message authentication code with existential unforgeability. Compared to the Fujisaki-Okamoto transformation, which is widely adopted by many KEM candidates in NIST's Post Quantum Cryptography (PQC) standardization project, our transformation replaces *de-randomization* (which might degrade the security of a randomized cryptosystem) and *re-encryption* (which is computationally inefficient and introduces additional risk of side channels) with computing MAC tag. We also provided concrete cryptanalysis on possible real-world attacks.

Instantiation with ElGamal and McEliece cryptosystem. We applied our KEM transformation to the ElGamal cryptosystem and the McEliece cryptosystems. We demonstrate that the "encrypt-then-MAC" KEM transformation is a generalization of DHIES by showing that the Gap Diffie-Hellman assumption is a special case of one-way security under plaintext checking attacks. We also surveyed plaintext checking attacks against many post quantum KEM candidates in the PQC standardizaton project.

C implementation of McEliece+. We implemented McEliece+ in C and benchmarked its performance. Compared to the reference implementation of Classic McEliece (which uses re-encryption), McEliece+ achieves significant decapsulation speedup at some minimal cost of encapsulation overhead, which results in 9-12% increase in throughput (encapsulation + decapsulation time).

1.2 Related works

OAEP *Optimal Asymmetric Encryption Padding (OAEP)* [7], proposed by Mihir Bellare and Phillip Rogaway in 1994, was one of the earliest provably secure CCA transformations. However, Victor Shoup identified a non-trivial gap in OAEP's security proof that cannot be filled under ROM[32], although Fujisaki et al. later proved that RSA-OAEP is secure under the RSA assumption [21]. RSA-OAEP is widely used in secure communication protocols such as TLS 1.2. The main drawback of OAEP is that it requires its input to be an one-way trapdoor permutation, which is difficult to find. To this day, RSA remains the only viable candidate to apply OAEP to.

REACT/GEM Okamoto and Pointchval proposed REACT [29] (Figure 1) in 2001, followed by GEM [16] in 2002. Both are generic CCA transformation with security proved under ROM. Okamoto and Pointcheval first defined the security notion of one-wayness under plaintext checking attack (OW-PCA) and reduced the CCA security of the transformation to the OW-PCA security of the input public-key cryptosystem.

$\mathtt{Enc}_{\mathrm{REACT}}(\mathtt{pk}, m)$

1: $w \leftarrow \mathcal{M}_{\mathrm{PKE}}$
2: $c_1 \leftarrow \mathtt{Enc}(\mathtt{pk}, w)$
3: $k \leftarrow G(w)$
4: $c_2 \leftarrow \mathcal{E}_k(m)$
5: $c_3 \leftarrow H(w, m, c_1, c_2)$
6: **return** (c_1, c_2, c_3)

$\mathtt{Dec}_{\mathrm{REACT}}(\mathtt{sk}, c)$

Require: $(c_1, c_2, c_3) \leftarrow c$
1: $\hat{w} \leftarrow \mathtt{Dec}(\mathtt{sk}, c_1)$
2: $\hat{k} \leftarrow G(\hat{w})$
3: $\hat{m} \leftarrow \mathcal{D}_{\hat{k}}(c_2)$
4: **if** $H(\hat{w}, \hat{m}, c_1, c_2) = c_3$ **then**
5: **return** $\hat{m}$
6: **else**
7: **return** $\perp$
8: **end if**

Fig. 1: Given PKE $(\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$, SKE $(\mathcal{E}, \mathcal{D})$, and hash functions G, H, REACT constructs a hybrid PKE $(\mathtt{KeyGen}_{\mathrm{REACT}}, \mathtt{Enc}_{\mathrm{REACT}}, \mathtt{Dec}_{\mathrm{REACT}})$

$\mathtt{Enc}_{\mathrm{GEM}}(\mathtt{pk}, m)$

1: $r \leftarrow \mathcal{R}$
2: $s \leftarrow F(m, r)$
3: $w \leftarrow s \| (r \oplus H(s))$
4: $c_1 \leftarrow \mathtt{Enc}(\mathtt{pk}, w)$
5: $k \leftarrow G(w, c_1)$
6: $c_2 \leftarrow \mathcal{E}_k(m)$
7: **return** (c_1, c_2)

$\mathtt{Dec}_{\mathrm{GEM}}(\mathtt{sk}, c)$

Require: $(c_1, c_2) \leftarrow c$
1: $\hat{w} \leftarrow \mathtt{Dec}(\mathtt{sk}, c_1)$
2: $(\hat{s}, \hat{t}) \leftarrow \hat{w}$
3: $\hat{r} \leftarrow \hat{t} \oplus H(\hat{s})$
4: $\hat{k} \leftarrow G(\hat{w}, c_1)$
5: $\hat{m} \leftarrow \mathcal{D}_{\hat{k}}(c_2)$
6: **if** $F(\hat{m}, \hat{r}) = \hat{s}$ **then**
7: **return** $\hat{m}$
8: **else**
9: **return** $\perp$
10: **end if**

Fig. 2: Given PKE $(\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$, SKE $(\mathcal{E}, \mathcal{D})$, and hash functions F, G, H, GEM constructs a hybrid PKE $(\mathtt{KeyGen}_{\mathrm{GEM}}, \mathtt{Enc}_{\mathrm{GEM}}, \mathtt{Dec}_{\mathrm{GEM}})$

Fujisaki-Okamoto transformation Fujisaki and Okamoto proposed to construct CCA secure hybrid PKE by combining a OW-CPA secure PKE and a semantically secure symmetric-key encryption (SKE) scheme [20]. The main techniques, namely *de-randomization* and *re-encrytion* were both introduced in the original proposal. Under ROM, Fujisaki and Okamoto reduced the CCA security of the hybrid PKE tightly to the semantic security of the input SKE and *non-tightly* to the OW-CPA security of the input PKE (with loss factor q, the number of hash oracle queries). Later works extended the original proposal to build CCA secure KEM: KEM's security model makes building secure KEM simpler than building secure PKE, and it is well-known that combining a CCA secure KEM with a CCA secure data encapsulation mechanism (DEM), such as some authenticated encryption scheme (e.g. AES-GCM, AES-CCM, ChaCha20-Poly1305), results in a CCA secure hybrid PKE [33,31]. Further studies [19,23,11,24,36,26] gave tighter security bounds, accounted for decryption failures in the underlying PKE, and analyzed the security under quantum random oracle model (QROM). To this day, the Fujisaki-Okamoto transformation is the only known generic CCA secure transformation that can convert OW-CPA/IND-CPA PKE into a CCA secure KEM. Because of the minimal input requirement and the simple construction, the Fujisaki-Okamoto transformation was widely adopted among post-quantum KEM candidates submitted to the PQC standardization project, including Kyber [14], Saber [18], FrodoKEM [13], and Classic McEliece [10].

Despite its widespread adoption, the Fujisaki-Okamoto transformation has many flaws:

- **Computational inefficiency.** In all variants of Fujisaki-Okamoto transformation, decapsulation routine needs to re-encrypt the decryption to ensure ciphertext non-malleability. For input PKE whose encryption routine carries significant computational cost, such as most lattice-based cryptosystems, re-encryption substantially slows down decapsulation.
- **Side-channel vulnerability.** Re-encryption introduces side-channels that can leak information about the decrypted PKE plaintext. As demonstrated in [35,34,25], these side-channels can be converted into efficient plaintext-checking attacks that can fully recover the secret key
- **Security degradation.** *de-randomization* can degrade the security of a randomized PKE. Where the security parameters did not account for this loss, the security of the KEM can fall below the expected level. Consequently, larger parameters are necessary to account for the security loss, which slows down the cryptosystem [8,9].

1.3 Paper organization

In Section 2, we review the preliminary definitions and theorems. In Section 3, we present the encrypt-then-MAC KEM transformation, proves its CCA security, and discusses practical attacks. In Section 4, we show that the encrypt-then-MAC transformation is a generalization of DHIES by applying it to the ElGamal cryptosystem. In Section 5, we present McEliece+, an instantiation of the

encrypt-then-MAC transformation, benchmark, and compare the performance of McEliece+ with Classic McEliece.

2 Preliminaries

2.1 Public-key encryption scheme

Syntax A public-key encryption scheme (PKE) is a collection fo three routines $(\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec})$ defined over some plaintext space $\mathcal{M}$ and some ciphertext space $\mathcal{C}$. Key generation $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}(1^\lambda)$ is a randomized routine that returns a keypair consisting of a public encryption key and a secret decryption key. The encryption routine $\texttt{Enc} : (\texttt{pk}, m) \mapsto c$ encrypts the input plaintext m under the input public key $\texttt{pk}$ and produces a ciphertext c. The decryption routine $\texttt{Dec} : (\texttt{sk}, c) \mapsto m$ decrypts the input ciphertext c under the input secret key and produces the corresponding plaintext. Where the encryption routine is randomized, we denote the randomness by a coin $r \in \mathcal{R}$ where $\mathcal{R}$ is called the coin space. Decryption routines are assumed to always be deterministic.

Correctness A PKE is δ-correct if

$$E\left[\max_{m \in \mathcal{M}} P\left[\texttt{Dec}(\texttt{sk}, c) \neq m \mid c \leftarrow \texttt{Enc}(\texttt{pk}, m)\right]\right] \leq \delta$$

Where the expectation is taken with respect to the probability distribution of all possible keypairs. For many lattice-based cryptosystems, decryption failures could leak information about the secret key, although the probability of a decryption failure is low enough that classical adversaries cannot exploit decryption failure more than they can defeat the underlying lattice problems.

Security The security of PKE's is conventionally discussed using adversarial games played between a challenger and an adversary [22]. In the OW-ATK game (Figure 3), the challenger samples a random keypair and a random encryption. The adversary is given the public key, the random encryption (also called the challenge ciphertext), and access to ATK, then asked to decrypt the challenge ciphertext.

OW-ATK game

1: $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathtt{KeyGen}(1^\lambda)$
2: $m^* \leftarrow \mathcal{M}$
3: $c^* \leftarrow \mathtt{Enc}(\mathtt{pk}, m)$
4: $\hat{m} \leftarrow A^{\mathrm{ATK}}(1^\lambda, \mathtt{pk}, c^*)$
5: **return** $[\![\hat{m} = m^*]\!]$

Fig. 3: The one-wayness game: challenger samples a random keypair and a random encryption, and the adversary wins if it correctly produces the decryption

The advantage of an adversary is its probability of producing the correct decryption: $\mathrm{Adv}^{\text{OW-ATK}}_{\mathtt{PKE}}(A) = P[\hat{m} = m^*]$. A PKE is said to be OW-ATK secure if no efficient adversary can win the OW-ATK game with non-negligible probability.

IND-ATK game

1: $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathtt{KeyGen}(1^\lambda)$
2: $(m_0, m_1) \leftarrow A^{\mathrm{ATK}}(1^\lambda, \mathtt{pk})$
3: $b \leftarrow \{0, 1\}$
4: $c^* \leftarrow \mathtt{Enc}(\mathtt{pk}, m_b)$
5: $\hat{b} \leftarrow A^{\mathrm{ATK}}(1^\lambda, \mathtt{pk}, c^*)$
6: **return** $[\![\hat{b} = b]\!]$

Fig. 4: IND-ATK game: adversary is asked to distinguish the encryption of one message from another

In the IND-ATK game (Figure 4), the adversary chooses two distinct messages and receives the encryption of one of them, randomly selected by the challenger. The advantage of an adversary is its probability of correctly distinguishing the ciphertext of one message from the other beyond blind guess: $\mathrm{Adv}^{\text{IND-ATK}}_{\mathtt{PKE}}(A) = |P[\hat{b} = b] - \frac{1}{2}|$. A PKE is said to be IND-ATK secure if no efficient adversary can win the IND-ATK game with non-negligible advantage.

In public-key cryptography, all adversaries are assumed to have access to the public key (ATK = CPA). If the adversary has access to a decryption oracle $\mathcal{O}_{\mathtt{Dec}} : c \mapsto \mathtt{Dec}(\mathtt{sk}, c)$, it is said to mount chosen-ciphertext attack (ATK = CCA). If the adversary has access to a plaintext-checking oracle (PCO) $\mathcal{O}_{\mathtt{PCO}} : (m, c) \mapsto [\![m = \mathtt{Dec}(\mathtt{sk}, c)]\!]$, then it is said to mount plaintext-checking attack (ATK = PCA).

$$\text{ATK} = \begin{cases} \text{CPA} & \mathcal{O}_{\text{ATK}} = \cdot \\ \text{PCA} & \mathcal{O}_{\text{ATK}} = \mathcal{O}_{\texttt{PCO}} \\ \text{CCA} & \mathcal{O}_{\text{ATK}} = \mathcal{O}_{\texttt{Dec}} \end{cases}$$

2.2 Key encapsulation mechanism (KEM)

Syntax 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 $\mathcal{K}$. Key generation $\texttt{KeyGen} : 1^\lambda \mapsto (\texttt{pk}, \texttt{sk})$ is a randomized routine that returns a keypair. Encapsulation $\texttt{Encap} : \texttt{pk} \mapsto (c, K)$ is a randomized routine that takes a public encapsulation key and returns a pair of ciphertext c and shared secret K (also commonly referred to as session key). Decapsulation $\texttt{Decap} : (\texttt{sk}, c) \mapsto K$ is a deterministic routine that uses the secret key $\texttt{sk}$ to recover the shared secret K from the input ciphertext c. Where the KEM chooses to reject invalid ciphertext explicitly, the decapsulation routine can also output the rejection symbol $\perp$. We assume a KEM to be perfectly correct:

$$P\left[\texttt{Decap}(\texttt{sk}, c) = K \mid (\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}(1^\lambda); (c, K) \leftarrow \texttt{Encap}(\texttt{pk})\right] = 1$$

Security Similar to PKE security, the security of KEM is discussed using adversarial games. In the IND-ATK game (Figure 5), the challenger generates a random keypair and encapsulates a random secret; the adversary is given the public key and the ciphertext, then asked to distinguish the shared secret from a random bit string.

KEM IND-ATK Game

1: $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}((1^\lambda))$
2: $(c^*, K_0) \leftarrow \texttt{Encap}(\texttt{pk})$
3: $K_1 \leftarrow \mathcal{K}$
4: $b \leftarrow \{0, 1\}$
5: $\hat{b} \leftarrow A^{\texttt{ATK}}(1^\lambda, \texttt{pk}, c^*, K_b)$
6: **return** $[\![\hat{b} = b]\!]$

Fig. 5: The IND-ATK game for KEM

The advantage of an adversary is its probability of winning beyong blind guess. A KEM is said to be IND-ATK secure if no efficient adversary can win the IND-ATK game with non-negligible advantage.

$$\mathtt{Adv}^{\mathtt{IND\text{-}ATK}}(A) = \left| P\left[A^{\mathtt{ATK}}(1^\lambda, c^*, K_b) = b \mid \begin{array}{r} (\mathtt{pk}, \mathtt{sk}) \leftarrow \mathtt{KeyGen}(1^\lambda); \\ (c^*, K_0) \leftarrow \mathtt{Encap}(\mathtt{pk}); \\ K_1 \leftarrow \mathcal{K}; b \leftarrow \{0,1\} \end{array} \right] - \frac{1}{2} \right|$$

By default, all adversaries are assumed to have the public key, with which they can mount chosen plaintext attacks (ATK = CPA). If the adversary as access to a decapsulation oracle $\mathcal{O}_{\mathtt{Decap}} : c \mapsto \mathtt{Decap}(\mathtt{sk}, c)$, it is said to mount a chosen-ciphertext attack (ATK = CCA).

2.3 Message authentication code (MAC)

Syntax A message authentication code (MAC) is a collection of two routines $(\mathtt{Sign}, \mathtt{Verify})$ defined over some key space $\mathcal{K}$, some message space $\mathcal{M}$, and some tag space $\mathcal{T}$. The signing routine $\mathtt{Sign} : (k, m) \mapsto t$ authenticates the message m under the symmetric key k by producing a tag t. The verification routine $\mathtt{Verify}(k, m, t)$ outputs 1 if the message-tag pair (m, t) is authentic under the symmetric key k and 0 otherwise. Many MAC constructions are deterministic: for these constructions it is simpler to denote the signing routine by $t \leftarrow \mathtt{MAC}(k, m)$, and verification done using a simple comparison. Some MAC constructions require a distinct or randomized nonce $r \leftarrow \mathcal{R}$, and the signing routine will take this additional argument $t \leftarrow \mathtt{MAC}(k, m; r)$.

Security The standard security notion for a MAC is *existential unforgeability under chosen message attack (EUF-CMA)*. We define it using an adversarial game in which an adversary has access to a signing oracle $\mathcal{O}_{\mathtt{Sign}} : m \mapsto \mathtt{Sign}(k, m)$ and tries to produce a valid message-tag pair that has not been queried from the signing oracle (Figure 6).

MAC EUF-CMA game

1: $k^* \leftarrow \mathcal{K}$
2: $(\hat{m}, \hat{t}) \leftarrow A^{\mathtt{CMA}}()$
3: **return** $[\![\mathtt{Verify}(k^*, \hat{m}, \hat{t}) \wedge (\hat{m}, \hat{t}) \notin \mathcal{O}_{\mathtt{Sign}}]\!]$

Fig. 6: The signing oracle signs the queried message with the secret key. The adversary must produce a message-tag pair that has never been queried before

The advantage of the adversary is the probability that it successfully produces a valid message-tag pair. A MAC is said to be EUF-CMA secure if no efficient adversary has non-negligible advantage. Some MACs are *one-time existentially*

unforgeable (we call them one-time MAC), meaning that each secret key can be used to authenticate exactly one message. The corresponding security game is identical to the EUF-CMA game except for that the signing oracle will only answer up to one query.

3 The encrypt-then-MAC transformation

In this section we present the encrypt-then-MAC KEM transformation. The transformation constructs an IND-CCA secure KEM using an OW-PCA secure PKE and an existentially unforgeable MAC. Our scheme is inspired by DHIES, but differs from it in two key aspects: whereas DHIES reduces its CCA security specifically to the Gap Diffie-Hellman assumption [28], our construction's CCA security reduces generically to the PCA security of the the input PKE; in addition, we argue that if the PKE's plaintext space is large and the sampling method has sufficient entropy, then the MAC only needs to be one-time existentially unforgeable (Abdalla, Rogaway, and Bellare originally proposed to use HMAC and CBC-MAC, which are many-time secure MAC but less efficient than one-time MAC). The data flow of the encapsulation is illustrated in Figure 7

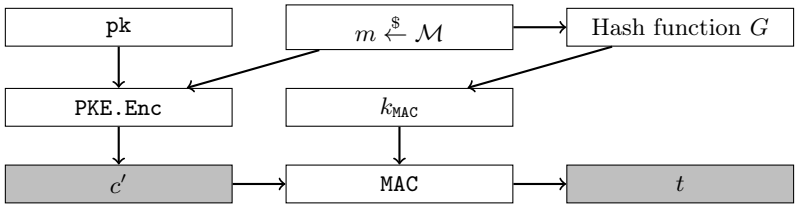

Fig. 7: Combining PKE with MAC using encrypt-then-MAC to ensure ciphertext integrity

In Section 3.1 we will describe the encrypt-then-MAC KEM routines and state the security reduction. In Section 3.2 we present the proof reducing the IND-CCA security of the KEM tightly to the OW-PCA security of the underlying PKE and non-tightly to the unforgeability of the MAC. In Section 3.3 we discuss some generic attacks on our KEM transformation.

3.1 The construction

Let $\mathcal{B}^*$ denote the set of finite bit strings. Let $\mathcal{K}_{\mathtt{KEM}}$ denote the set of all possible shared secrets. Let $(\mathtt{KeyGen}_{\mathtt{PKE}}, \mathtt{Enc}_{\mathtt{PKE}}, \mathtt{Dec}_{\mathtt{PKE}})$ be a PKE defined over message space $\mathcal{M}_{\mathtt{PKE}}$ and ciphertext space $\mathcal{C}_{\mathtt{PKE}}$. Let $\mathtt{MAC} : \mathcal{K}_{\mathtt{MAC}} \times \mathcal{B}^* \to \mathcal{T}$ be a MAC over key space $\mathcal{K}_{\mathtt{MAC}}$ and tag space $\mathcal{T}$. Let $G : \mathcal{B}^* \to \mathcal{K}_{\mathtt{MAC}}, H : \mathcal{B}^* \to \mathcal{K}_{\mathtt{KEM}}$ be hash functions. The encrypt-then-MAC transformation $\mathtt{EtM}[\mathtt{PKE}, \mathtt{MAC}, G, H]$ constructs a KEM $(\mathtt{KeyGen}_{\mathtt{EtM}}, \mathtt{Encap}_{\mathtt{EtM}}, \mathtt{Decap}_{\mathtt{EtM}})$ (Figure 8).

$\texttt{KeyGen}_{\texttt{EtM}}()$

1: $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}_{\texttt{PKE}}()$
2: $s \leftarrow \mathcal{M}_{\texttt{PKE}}$
3: $\texttt{sk} \leftarrow (\texttt{sk}, s)$
4: **return** $(\texttt{pk}, \texttt{sk})$

$\texttt{Encap}_{\texttt{EtM}}(\texttt{pk})$

1: $m \leftarrow \mathcal{M}_{\texttt{PKE}}$
2: $k \leftarrow G(m)$
3: $c' \leftarrow \texttt{Enc}_{\texttt{PKE}}(\texttt{pk}, m)$
4: $t \leftarrow \texttt{MAC}(k, c')$
5: $c \leftarrow (c', t)$
6: $K \leftarrow H(m, c)$
7: **return** (c, K)

$\texttt{Decap}_{\texttt{EtM}}(\texttt{sk}, c)$

Require: $c = (c', t)$
Require: $\texttt{sk} = (\texttt{sk}', s)$
1: $\hat{m} \leftarrow \texttt{Dec}_{\texttt{PKE}}(\texttt{sk}', c')$
2: $\hat{k} \leftarrow G(\hat{m})$
3: **if** $\texttt{MAC}(\hat{k}, c') = t$ **then**
4: $\quad K \leftarrow H(\hat{m}, c)$
5: **else**
6: $\quad K \leftarrow H(s, c)$
7: **end if**
8: **return** K

Fig. 8: The encrypt-then-MAC KEM routines

We chose to construct $\texttt{KEM}_{\texttt{EtM}}$ using implicit rejection $K \leftarrow H(s, c)$: on invalid ciphertexts, the decapsulation routine returns a fake shared secret that depends on the ciphertext and some secret values, though choosing to use explicit rejection should not impact the security of the KEM. In addition, because the underlying PKE can be randomized, the shared secret $K \leftarrow H(m, c)$ must depend on both the plaintext and the ciphertext. According to [19,23], if the input PKE is *rigid* (i.e. $m = \texttt{Dec}(\texttt{sk}, c)$ if and only if $c = \texttt{Enc}(\texttt{pk}, m)$), such as with RSA, then the shared secret may be derived from the plaintext alone $K \leftarrow H(m)$.

The CCA security of $\texttt{KEM}_{\texttt{EtM}}$ can be intuitively argued through an adversary's inability to learn additional information from the decapsulation oracle. For an adversary A to produce a valid tag for some unauthenticated ciphertext c', it must either know the correct symmetric key or produce a forgery. Under the random oracle, A cannot know the symmetric key without knowing its pre-image under the hash function G, so A must either produced c' honestly, or have broken the one-wayness of the underlying PKE. This means that the decapsulation oracle will not leak information on decryption that the adversary does not already know. We formalize the security in Theorem 1

Theorem 1. *For every IND-CCA adversary A against $\texttt{KEM}_{\texttt{EtM}}$ that makes q decapsulation queries, there exists a OW-PCA adversary B against the underlying PKE making at least q decapsulation queries, and an existential forgery adversary C against the underlying MAC such that:*

$$\mathit{Adv}^{\mathit{IND\text{-}CCA}}_{\mathit{KEM}_{\texttt{EtM}}}(A) \leq q \cdot \mathit{Adv}_{\mathit{MAC}}(C) + 2 \cdot \mathit{Adv}^{\mathit{OW\text{-}PCA}}_{\mathit{PKE}}(B)$$

3.2 Proof of Theorem 1

3.3 Cryptanalysis

4 Application to ElGamal

5 McEliece+: applying encrypt-then-MAC to the McEliece cryptosystem

References

1. ISO/IEC 9796: Information Technology — Security Techniques — Digital Signature Scheme Giving Message Recovery, Part 1: Mechanisms Using Redundancy (1999), part 1 of the ISO/IEC 9796 standard
2. Ieee standard specifications for public-key cryptography - amendment 1: Additional techniques. IEEE Std 1363a-2004 (Amendment to IEEE Std 1363-2000) pp. 1–167 (2004). https://doi.org/10.1109/IEEESTD.2004.94612
3. Abdalla, M., Bellare, M., Rogaway, P.: DHAES: an encryption scheme based on the diffie-hellman problem. IACR Cryptol. ePrint Arch. p. 7 (1999), http://eprint.iacr.org/1999/007
4. Abdalla, M., Bellare, M., Rogaway, P.: The oracle diffie-hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2020, pp. 143–158. Springer (2001). https://doi.org/10.1007/3-540-45353-9_12, https://doi.org/10.1007/3-540-45353-9_12
5. ANSI, X.: 63: Public key cryptography for the financial services industry, key agreement and key transport using elliptic curve cryptography. American National Standards Institute (1998)
6. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000). https://doi.org/10.1007/3-540-44448-3_41, https://doi.org/10.1007/3-540-44448-3_41
7. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Santis, A.D. (ed.) Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings. Lecture Notes in Computer Science, vol. 950, pp. 92–111. Springer (1994). https://doi.org/10.1007/BFB0053428, https://doi.org/10.1007/BFb0053428
8. Bernstein, D.J.: FO derandomization sometimes damages security. Cryptology ePrint Archive, Paper 2021/912 (2021), https://eprint.iacr.org/2021/912
9. Bernstein, D.J.: On the looseness of FO derandomization. IACR Cryptol. ePrint Arch. p. 912 (2021), https://eprint.iacr.org/2021/912
10. Bernstein, D.J., Chou, T., Schwabe, P.: Mcbits: Fast constant-time code-based cryptography. In: Bertoni, G., Coron, J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8086, pp. 250–272. Springer (2013). https://doi.org/10.1007/978-3-642-40349-1_15, https://doi.org/10.1007/978-3-642-40349-1_15

11. Bernstein, D.J., Persichetti, E.: Towards KEM unification. IACR Cryptol. ePrint Arch. p. 526 (2018), https://eprint.iacr.org/2018/526
12. Bleichenbacher, D.: Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 1–12. Springer (1998). https://doi.org/10.1007/BFB0055716, https://doi.org/10.1007/BFb0055716
13. Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 1006–1018. ACM (2016). https://doi.org/10.1145/2976749.2978425, https://doi.org/10.1145/2976749.2978425
14. Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018. pp. 353–367. IEEE (2018). https://doi.org/10.1109/EUROSP.2018.00032, https://doi.org/10.1109/EuroSP.2018.00032
15. Coppersmith, D., Halevi, S., Jutla, C.: Iso 9796-1 and the new forgery strategy. rump session of Crypto **99** (1999)
16. Coron, J., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., Tymen, C.: GEM: A generic chosen-ciphertext secure encryption method. In: Preneel, B. (ed.) Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2271, pp. 263–276. Springer (2002). https://doi.org/10.1007/3-540-45760-7_18, https://doi.org/10.1007/3-540-45760-7_18
17. Coron, J., Naccache, D., Stern, J.P.: On the security of RSA padding. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 1–18. Springer (1999). https://doi.org/10.1007/3-540-48405-1_1, https://doi.org/10.1007/3-540-48405-1_1
18. D'Anvers, J., Karmakar, A., Roy, S.S., Vercauteren, F.: Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In: Joux, A., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10831, pp. 282–305. Springer (2018). https://doi.org/10.1007/978-3-319-89339-6_16, https://doi.org/10.1007/978-3-319-89339-6_16
19. Dent, A.W.: A designer's guide to kems. In: Paterson, K.G. (ed.) Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2898, pp. 133–151. Springer (2003). https://doi.org/10.1007/978-3-540-40974-8_12, https://doi.org/10.1007/978-3-540-40974-8_12
20. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, Au-

gust 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer (1999). https://doi.org/10.1007/3-540-48405-1_34, https://doi.org/10.1007/3-540-48405-1_34
21. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 260–274. Springer (2001). https://doi.org/10.1007/3-540-44647-8_16, https://doi.org/10.1007/3-540-44647-8_16
22. Goldwasser, S., Micali, S.: Probabilistic encryption and how to play mental poker keeping secret all partial information. In: Lewis, H.R., Simons, B.B., Burkhard, W.A., Landweber, L.H. (eds.) Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA. pp. 365–377. ACM (1982). https://doi.org/10.1145/800070.802212, https://doi.org/10.1145/800070.802212
23. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer (2017). https://doi.org/10.1007/978-3-319-70500-2_12, https://doi.org/10.1007/978-3-319-70500-2_12
24. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the fujisaki-okamoto transform. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 414–443. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_15, https://doi.org/10.1007/978-3-031-22972-5_15
25. Huguenin-Dumittan, L., Vaudenay, S.: Classical misuse attacks on NIST round 2 PQC - the power of rank-based schemes. In: Conti, M., Zhou, J., Casalicchio, E., Spognardi, A. (eds.) Applied Cryptography and Network Security - 18th International Conference, ACNS 2020, Rome, Italy, October 19-22, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12146, pp. 208–227. Springer (2020). https://doi.org/10.1007/978-3-030-57808-4_11, https://doi.org/10.1007/978-3-030-57808-4_11
26. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer (2018). https://doi.org/10.1007/978-3-319-96878-0_4, https://doi.org/10.1007/978-3-319-96878-0_4
27. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is ssl?). In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 310–331. Springer (2001). https://doi.org/10.1007/3-540-44647-8_19, https://doi.org/10.1007/3-540-44647-8_19
28. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: Kim, K. (ed.) Public Key Cryptography,

4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings. Lecture Notes in Computer Science, vol. 1992, pp. 104–118. Springer (2001). https://doi.org/10.1007/3-540-44586-2_8, https://doi.org/10.1007/3-540-44586-2_8

29. Okamoto, T., Pointcheval, D.: REACT: rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2020, pp. 159–175. Springer (2001). https://doi.org/10.1007/3-540-45353-9_13, https://doi.org/10.1007/3-540-45353-9_13
30. RSA Data Security, I.: PKCS 1: RSA Encryption Standard Version 1.5. Request for Comments: 2313 (Mar 1998), https://www.rfc-editor.org/rfc/rfc2313
31. Shoup, V.: Using hash functions as a hedge against chosen ciphertext attack. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 275–288. Springer (2000). https://doi.org/10.1007/3-540-45539-6_19, https://doi.org/10.1007/3-540-45539-6_19
32. Shoup, V.: OAEP reconsidered. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 239–259. Springer (2001). https://doi.org/10.1007/3-540-44647-8_15, https://doi.org/10.1007/3-540-44647-8_15
33. Shoup, V.: A proposal for an ISO standard for public key encryption. IACR Cryptol. ePrint Arch. p. 112 (2001), http://eprint.iacr.org/2001/112
34. Tanaka, Y., Ueno, R., Xagawa, K., Ito, A., Takahashi, J., Homma, N.: Multiple-valued plaintext-checking side-channel attacks on post-quantum kems. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2023**(3), 473–503 (2023). https://doi.org/10.46586/TCHES.V2023.I3.473-503, https://doi.org/10.46586/tches.v2023.i3.473-503
35. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/em analysis on post-quantum kems. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(1), 296–322 (2022). https://doi.org/10.46586/TCHES.V2022.I1.296-322, https://doi.org/10.46586/tches.v2022.i1.296-322
36. Xagawa, K., Yamakawa, T.: (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers. Lecture Notes in Computer Science, vol. 11505, pp. 249–268. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_14, https://doi.org/10.1007/978-3-030-25510-7_14