A note on IND-qCCA security in the ROM and its applications

Loïs Huguenin-Dumittan, Serge Vaudenay

LASEC, EPFL, Switzerland
{lois.huguenin-dumittan,serge.vaudenay}@epfl.ch

Abstract. We show in this note that bounded KEM IND-CCA security (IND-qCCA) is easily obtained from any passively secure PKE in the (Q)ROM. That is, simply adding a confirmation hash or computing the key as the hash of the plaintext and ciphertext holds an IND-qCCA KEM. In particular, there is no need for derandomization or re-encryption as in the Fujisaki-Okamoto transform [8]. Such KEMs could be used in the recently proposed KEMTLS protocol [17] that requires IND-1CCA ephemeral key-exchange mechanisms. We also highlight and briefly discuss several use cases of IND-1CCA KEMs in TLS and ratcheting primitives.

1 Introduction

We show how to build an efficient IND-qCCA KEM (i.e. the adversary can only make q decapsulation queries) from any OW-CPA PKE in the ROM. The bound has a loose factor of 2^q, making it insecure or impractical for more than a few queries. However, such a construction is sufficient to build an efficient IND-1CCA KEM from any OW-CPA public-key encryption scheme. The transform simply sends a confirmation hash along the ciphertext encrypting the seed. In addition, we prove the security of such a construction in the QROM as well.

Such a construction might be useful in the KEMTLS protocol designed by Schwabe et al. [17], which requires IND-1CCA KEMs for ephemeral key-exchange. However, in the current version of the paper [18], these KEMs are implemented with IND-CCA KEMs, which are usually obtained by applying the Fujisaki-Okamoto (FO) transform. The FO construction re-encrypts the decrypted plaintext during decapsulation, making it an expensive operation.

We also note that plain IND-CPA PQ schemes are often not IND-1CCA. In particular, it is stated in Section 4.3 of the KEMTLS paper [18]:

> *"We leave as an open question to what extent non-FO-protected post-quantum KEMs may be secure against a single decapsulation query, but at this point IND-CCA is the safe choice."*

It obviously depends on how the IND-CPA PKE is used as a KEM, but if it used in the trivial way (i.e. $m \leftarrow_{\$} \mathcal{M}$, $K := H(m), \mathsf{ct} := \mathsf{enc}(\mathsf{pk}, m)$) most schemes can be broken with 1 query. The adversary receives $K^*, \mathsf{ct}^* := \mathsf{enc}(\mathsf{pk}, m^*)$, queries

$\mathsf{ct}^* + \delta$ and gets back $H(m^*)$ with high probability, if δ is "small", for most PQ PKEs. Then, it can just compare whether $H(m^*) = K^*$ or not and break IND-1CCA security.

The reaction attacks (e.g. [7]) requiring thousands of queries mentioned in the same paper [18] are key-recovery attacks, not distinguishing attacks.

However, we also show that deriving the key as $K := H(m, \mathsf{ct})$ holds an IND-qCCA KEM in the ROM. The bound is worse than for the first transform, having a $\approx q_H^{2q}$ factor, where q_H is the number of queries an adversary can make to the random oracle H.

Finally, in Section 4, we discuss possible use cases of IND-qCCA in the context of communication protocols and ratcheting primitives. In particular, we note that IND-1CCA security is sufficient in many recent applications as the trend is to move to forward secure schemes, which discard key pairs after one use.

Related work. The notion of bounded IND-CCA (i.e. IND-qCCA) has been studied in several works. Cramer et al. [5] defined IND-qCCA and showed that one can build an IND-qCCA PKE from any CPA-secure PKE in a black-box manner in the standard model, using one-time signatures. Peirera et al. [15] built a more efficient IND-qCCA PKE based on the CDH assumption and Yamakawa et al. [21] proposed other constructions based on the factoring and bilinear CDH assumptions. As far as we know, we are the first to note that a IND-qCCA KEM can be obtained from any CPA-secure PKE through a very simple and efficient transform in the ROM.

2 PKC and KEM

2.1 Public-Key Encryption scheme

A Public-Key Encryption (PKE) scheme is defined as follows.

Definition 1 (Public-Key Encryption). *A Public-Key Encryption scheme is composed of four algorithms* $\mathsf{setup}, \mathsf{gen}, \mathsf{enc}, \mathsf{dec}$*:*

- $\mathsf{pp} \leftarrow_{\$} \mathsf{setup}(1^\lambda)$*: The setup algorithm randomly generates the public parameters* pp *according to a security parameter* λ*.*
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{gen}(\mathsf{pp})$*: The key generation algorithm takes the public parameters as inputs and outputs the public key* pk *and the secret key* sk*.*
- $\mathsf{ct} \leftarrow_{\$} \mathsf{enc}(\mathsf{pp}, \mathsf{pk}, \mathsf{pt})$*: The encryption algorithm takes as inputs the public parameters* pp*, the public key* pk *and a plaintext* $\mathsf{pt} \in \mathcal{M}$ *and it outputs a ciphertext* ct*.*
- $\mathsf{pt}' \leftarrow \mathsf{dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$*: The decryption procedure takes as inputs the public parameters* pp*, the secret key* sk *and the ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and it outputs a plaintext* $\mathsf{pt}' \in \mathcal{M} \cup \{\perp\}$*.*

The setup*,* gen *and* enc *are probabilistic algorithms that can be made deterministic by adding random coins as inputs. The decryption procedure is deterministic. Finally, for the sake of simplicity, we omit the public parameters in the inputs from now on.*

$\underline{\mathrm{CORR}_{\mathsf{PKE}}(\mathcal{A})}$

$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(\mathsf{pp})$
$\mathsf{pt} \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
$\mathsf{ct} \leftarrow\!\!\$\ \mathsf{enc}(\mathsf{pk}, \mathsf{pt})$
return $1_{\mathsf{dec}(\mathsf{sk},\mathsf{ct}) \neq \mathsf{pt}}$

Fig. 1: Correctness game.

Table 1: Oracles for OW games.

ATK	CPA	PCA
$\mathcal{O}^{\mathsf{ATK}}$	$\perp$	$\mathcal{O}^{\mathsf{PCO}}$

Correctness. We say a PKE scheme is δ correct if for any ppt adversary $\mathcal{A}$ playing the game CORR defined in Figure 1, we have

$$\Pr[\mathrm{CORR}_{\mathsf{PKE}}(\mathcal{A}) \Rightarrow 1] \leq \delta(\lambda)$$

where λ is the security parameter, we omit it from now on for the sake of simplicity.

Plaintext Checking. We recall the notions of One-Wayness under Chosen Plaintext Attacks (OW-CPA) and Plaintext-Checking Attacks (OW-PCA).

Definition 2 (One-Wayness and Plaintext Checking). *Let $\mathcal{M}$ be the message space,* PKE *a PKE scheme and we consider the games defined in Figure 2 with the different oracles as defined in Table 1. Then,* PKE *is* OW-ATK*, for* $\mathsf{ATK} \in \{\mathrm{CPA}, \mathrm{PCA}\}$*, if for any ppt adversary $\mathcal{A}$ we have*

$$\mathsf{Adv}^{\text{ow-atk}}_{\mathsf{PKE}}(\mathcal{A}) = \Pr\left[\text{OW-}\mathsf{ATK}_{\mathsf{PKE}}(\mathcal{A}) \Rightarrow 1\right] = \mathsf{negl}(\lambda)$$

where $\Pr\left[\text{OW-}\mathsf{ATK}_{\mathsf{PKE}}(\mathcal{A}) \Rightarrow 1\right]$ *is the probability that the adversary wins the* OW-ATK *game.*

2.2 Key Encapsulation Mechanism (KEM)

A Key Encapsulation Mechanism is defined as follows.

Definition 3 (Key Encapsulation Mechanism). *A KEM is a tuple of three algorithms* setup, gen, encaps, decaps*:*

- $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{setup}(1^\lambda)$*: The setup algorithm takes the security parameter λ as input and outputs the public parameters* pp*.*

$\underline{\text{OW-ATK}_{\mathsf{PKE}}(\mathcal{A})}$

$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(\mathsf{pp})$
$\mathsf{pt}^* \leftarrow\!\!\$\ \mathcal{M}$
$\mathsf{ct}^* \leftarrow \mathsf{enc}(\mathsf{pk}, \mathsf{pt}^*)$
$\mathsf{pt}' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{ATK}}}(\mathsf{pk}, \mathsf{ct}^*)$
return $1_{\mathsf{pt}'=\mathsf{pt}^*}$

$\underline{\textbf{Oracle } \mathcal{O}^{\mathsf{PCO}}(\mathsf{pt}, \mathsf{ct})}$

1 : $\mathsf{pt}' \leftarrow \mathsf{dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$
2 : **return** $1_{\mathsf{pt}'=\mathsf{pt}}$

Fig. 2: One-Wayness games.

$\underline{\text{IND-(q)CCA}_{\mathsf{KEM}}(\mathcal{A})}$

$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(\mathsf{pp})$
$b \leftarrow\!\!\$\ \{0, 1\}$
$\mathsf{ct}^*, K_0 \leftarrow\!\!\$\ \mathsf{encaps}(\mathsf{pk})$
$K_1 \leftarrow\!\!\$\ \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}}(\mathsf{pk}, \mathsf{ct}^*, K_b)$
return $1_{b'=b}$

$\underline{\textbf{Oracle } \mathcal{O}^{\mathsf{Dec}}(\mathsf{ct})}$

1 : **if** $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
2 : **if** more than q queries : **return** $\perp$ // If IND-qCCA
3 : $K' \leftarrow \mathsf{decaps}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$
4 : **return** K'

Fig. 3: Indistinguishability games.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(\mathsf{pp})$*: The key generation algorithm takes as inputs the public parameters and it outputs the public key* pk *and the secret key* sk.
- $\mathsf{ct}, K \leftarrow\!\!\$\ \mathsf{encaps}(\mathsf{pp}, \mathsf{pk})$*: The encapsulation algorithm takes as inputs the public parameters* pp*, the public key* pk *and it outputs a ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and a key* $K \in \mathcal{K}$.
- $K' \leftarrow \mathsf{decaps}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$*: The decapsulation procedure takes as inputs the public parameters* pp*, the secret key* sk *and the ciphertext* $\mathsf{ct} \in \mathcal{C}$ *and it outputs a key* K*. If the KEM allows explicit rejection, the output is a key* $K \in \mathcal{K}$ *or the rejection symbol* $\perp$*. If the rejection is implicit, the output is always a key* $K \in \mathcal{K}$.

The setup*,* gen *and* encaps *are probabilistic algorithms. The randomness can be made explicit by adding random coins as inputs. The decapsulation function is deterministic. We omit the public parameters in the inputs from now on.*

Indistinguishability security. KEM indistinguishability against bounded and unbounded adversaries are defined as follows.

Definition 4. *We consider the games defined in Figure 3. Let* $\mathcal{K}$ *be the key space. A KEM scheme* $\mathsf{KEM} = (\mathsf{setup}, \mathsf{gen}, \mathsf{encaps}, \mathsf{decaps})$ *is* IND-CCA *(resp.* IND-qCCA*) if for any ppt adversary* $\mathcal{A}$ *(resp. any ppt* $\mathcal{A}$ *limited to* q *decapsula-*

$\mathsf{gen}()$	$\mathsf{encaps}(\mathsf{pk})$	$\mathsf{decaps}(\mathsf{sk},\mathsf{ct})$
$(\mathsf{pk},\mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}^{\mathsf{p}}()$	$\sigma \leftarrow\!\!\$\ \mathcal{M}$	$(\mathsf{ct}'_0,\mathsf{tag}') \leftarrow \mathsf{ct}$
return $(\mathsf{pk},\mathsf{sk})$	$\mathsf{ct}_0 \leftarrow\!\!\$\ \mathsf{enc}^{\mathsf{p}}(\mathsf{pk},\sigma)$	$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk},\mathsf{ct}'_0)$
	$\mathsf{tag} \leftarrow H'(\sigma,\mathsf{ct}_0)$	**if** $H'(\sigma',\mathsf{ct}'_0) \neq \mathsf{tag}'$:
	$K \leftarrow H(\sigma)$	**return** $\perp$
	return $K,(\mathsf{ct}_0,\mathsf{tag})$	**return** $H(\sigma')$

Fig. 4: T_{CH} transform.

tion queries) we have

$$\mathsf{Adv}^{\text{ind-(q)cca}}_{KEM}(\mathcal{A}) = \left|\Pr\left[\text{IND}-\text{(q)CCA}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$$

where $\Pr\left[\text{IND}-\text{(q)CCA}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1\right]$ *is the probability that* $\mathcal{A}$ *wins the* IND-*(q)CCA*$_{\mathsf{KEM}}(\mathcal{A})$ *game defined in Figure 3.*

3 OW-CPA to IND-qCCA transform

We first prove the following simple lemma.

Lemma 1. *Let* PKE *be a PKE. Then, for any ppt OW-PCA adversary* $\mathcal{A}$ *making at most* q *queries to the PCO oracle, there exists a OW-CPA adversary* $\mathcal{B}$ *s.t.*

$$\mathsf{Adv}^{\text{ow}-\text{pca}}_{\mathsf{PKE}}(\mathcal{A}) \leq 2^q \cdot \mathsf{Adv}^{\text{ow}-\text{cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Proof. We can simply see that the PCO oracle returns 1 bit of information, thus PKE loses at most q bits of security when a PCO oracle is available. More formally, given $\mathcal{A}$, one can build $\mathcal{B}$ as follows. It passes its input to $\mathcal{A}$ and simulates the PCO oracle by sampling a response at random in $\{0,1\}$. Then, it returns the response of $\mathcal{A}$. Its probability of success is $\mathsf{Adv}^{\text{ow}-\text{cpa}}_{\mathsf{PKE}}(\mathcal{B}) \geq \frac{1}{2^q}\mathsf{Adv}^{\text{ow}-\text{pca}}_{\mathsf{PKE}}(\mathcal{A})$, as the probability the q responses are correct is $\frac{1}{2^q}$. □

We consider the transform T_{CH} given in Figure 4 (the setup algorithm is ommited). This construction takes a PKE $\mathsf{PKE} = (\mathsf{setup}^{\mathsf{p}},\mathsf{gen}^{\mathsf{p}},\mathsf{enc}^{\mathsf{p}},\mathsf{dec}^{\mathsf{p}})$ and outputs a KEM $(\mathsf{setup},\mathsf{gen},\mathsf{encaps},\mathsf{decaps})$. Note that T_{CH} is basically the REACT transform [13] without the asymmetric part (to get a KEM instead of a PKE).

We now show that the resulting KEM is IND-qCCA assuming the underlying PKE is OW-PCA.

Theorem 1. *We consider two random oracles* $H, H' : \{0,1\}^* \mapsto \{0,1\}^n$. *Let* KEM *be the KEM resulting from applying the* T_{CH} *transform to a* δ*-correct* PKE.

Then, for any IND-qCCA adversary $\mathcal{A}$ that makes at most q_H (resp. $q_{H'}$) queries to H (resp. H'), there exists a OW-PCA adversary $\mathcal{B}$ s.t.

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \frac{(q+q_{H'})^2}{2^n} + \delta + \frac{q}{2^n} + (q_H + q_{H'} + q) \cdot \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B})$$

where $\mathcal{B}$ makes at most q queries to its plaintext-checking oracle. In addition, if PKE *is a deterministic encryption scheme, the bound becomes*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \frac{(q+q_{H'})^2}{2^n} + \delta + \frac{q}{2^n} + \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Proof. We proceed by game hopping, the sequence of games is presented in Figure 5. Let $\mathcal{L}_H$ (resp. $\mathcal{L}_{H'}$) be the list of queries (x, h) made to the RO H (resp. H') s.t. $H(x) = h$ (resp. $H'(x) = h$). In addition, let the challenge ciphertext be $\mathsf{ct}^* = (\mathsf{ct}_0^*, h^*)$, and σ^* be s.t. $\mathsf{enc}^{\mathsf{p}}(\mathsf{pk}, \sigma^*) = \mathsf{ct}^*$. We start with game Γ^0 which is the IND-qCCA game, except we abort if the adversary finds a collision on H' (i.e. $H'(x) = H'(x')$ for $x \neq x'$ and $(x, h), (x', h) \in \mathcal{L}_{H'}$). This happens with prob. at most $\frac{(q+q_{H'})^2}{2^n}$ and we have

$$\left|\Pr\left[\mathrm{IND-qCCA}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1\right] - \Pr[\Gamma^0(\mathcal{A}) \Rightarrow 1]\right| \leq \frac{(q+q_{H'})^2}{2^n} \ .$$

$\underline{\Gamma^1:}$ The decapsulation oracle is modified s.t. it returns $\perp$ whenever ct_0^* or h^* is queried (note that both cannot be submitted at the same time). This game is the same as Γ^0 except if the oracle in Γ^0 does not return $\perp$ on such queries. Let bad be this event. We split this into two cases:

- $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct}_0^*, h \neq h^*) \neq \perp$. This happens only if

  $$H'(\mathsf{dec}(\mathsf{sk}, \mathsf{ct}_0^*), \mathsf{ct}_0^*) = h \neq h^* = H'(\sigma^*, \mathsf{ct}_0^*) \ .$$

  In turn, this implies that $\mathsf{dec}(\mathsf{sk}, \mathsf{ct}_0^*) \neq \sigma^*$ and thus it is a correctness error. Such an error happens at most with probability δ.
- $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct}_0 \neq \mathsf{ct}_0^*, h^*) \neq \perp$. It means that $h^* = H'(\sigma^*, \mathsf{ct}_0^*) = H'(\sigma', \mathsf{ct}_0)$, with $\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}_0)$, which is not possible since $\mathsf{ct}_0 \neq \mathsf{ct}_0^*$ and we assume no collision occurs.

Therefore, overall $\Pr[\mathsf{bad}] \leq \delta$ and

$$|\Pr[\Gamma^0 \Rightarrow 1] - \Pr[\Gamma^1 \Rightarrow 1]| \leq \Pr[\mathsf{bad}] \leq \delta \ .$$

$\underline{\Gamma^2:}$ We modify the decapsulation oracle into another oracle $\mathcal{O}^{\mathsf{Dec}^2}$ as follows. On a decapsulation query (ct, h) (with $\sigma' \leftarrow \mathsf{dec}(\mathsf{sk}, \mathsf{ct})$):

1. If there is no $((*, \mathsf{ct}), h)$ in $\mathcal{L}_{H'}$: return $\perp$. This differs from the previous game only if $h = H'(\sigma', \mathsf{ct})$ but (σ', ct) was never queried to H'. As the RO values are unif. distributed, this happens at most with probability $\frac{1}{2^n}$.

2. If $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_{H'}$ for some σ: If $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}) := 1_{\mathsf{dec}(\mathsf{sk},\mathsf{ct})=\sigma} = 1$, return $H(\sigma)$. Otherwise, return $\perp$. This perfectly simulates the previous oracle as $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}) = 1$ iff $\sigma = \sigma'$ and we know $h = H(\sigma = \sigma', \mathsf{ct})$.
Note that there is at most one σ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_{H'}$ as we assume no collision occurs. In particular, it means that $\mathcal{O}^{\mathsf{PCO}}$ is called at most once every decapsulation query.

Therefore, by a union bound we get

$$|\Pr[\Gamma^1 \Rightarrow 1] - \Pr[\Gamma^2 \Rightarrow 1]| \leq \frac{q}{2^n} \ .$$

$\underline{\Gamma^3}$: Finally, we abort whenever $\mathcal{A}$ queries σ^* to H or $(\sigma^*, \cdot)$ to H'. Let this event be query. Then, we can build a OW-PCA adversary $\mathcal{B}$ (shown in Figure 6) that perfectly simulates $\mathcal{A}$'s view as long as query does not happen. More precisely, $\mathcal{B}$ can simulate the decapsulation oracle using its PCO oracle. Then, on input $(\mathsf{pk}, \mathsf{ct}^*)$, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{pk}, (\mathsf{ct}^*, h^*), K^*)$, where h^* and K^* are picked at random. Unless query occurs, $\mathcal{A}$ cannot distinguish between these random h^*, K^* and the real ones. Finally, if query occurs, $\mathcal{B}$ can recover σ^* with probability $\frac{1}{q_H+q_{H'}+q}$ by sampling a random σ from $S = \{\sigma : (\sigma, *) \in \mathcal{L}_H \vee ((\sigma, *), *) \in \mathcal{L}_{H'}\}$. Thus,

$$|\Pr[\Gamma^2 \Rightarrow 1] - \Pr[\Gamma^3 \Rightarrow 1]| \leq \Pr[\mathsf{query}] \leq (q_H + q_{H'} + q) \cdot \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B})$$

where $\mathcal{B}$ makes q query to the PCO oracle. Note that if PKE is deterministic, $\mathcal{B}$ can check whether $\mathsf{enc}(\mathsf{pk}, \sigma) = \mathsf{ct}^*$ for all $\sigma \in S$ to find σ^*. In this case, we obtain

$$|\Pr[\Gamma^2 \Rightarrow 1] - \Pr[\Gamma^3 \Rightarrow 1]| \leq \Pr[\mathsf{query}] \leq \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Finally, since $\mathcal{A}$ cannot query σ^* to H anymore, it cannot distinguish between a random key and $H(\sigma^*)$. Hence, $\Pr[\Gamma^3 \Rightarrow 1] = \frac{1}{2}$. Collecting the probabilities holds the result. □

Corollary 1. *We consider two random oracles $H, H' : \{0,1\}^* \mapsto \{0,1\}^n$. Let KEM be the KEM resulting from applying the T_{CH} transform to a δ-correct PKE. Then, for any IND-qCCA adversary $\mathcal{A}$ that makes at most q_H (resp. $q_{H'}$) queries to H (resp. H'), there exists a OW-CPA adversary $\mathcal{B}$ s.t.*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \frac{(q+q_{H'})^2}{2^n} + \delta + \frac{q}{2^n} + (q_H + q_{H'} + q)2^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

If PKE is deterministic, we get

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \frac{(q+q_{H'})^2}{2^n} + \delta + \frac{q}{2^n} + 2^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

In particular, in the case of IND-1CCA (i.e. $q = 1$), if the underlying PKE is OW-CPA the KEM obtained from the T_{CH} transform is IND-1CCA with a security

$\Gamma^{0-3}(\mathcal{A})$

$\mathsf{pp} \leftarrow\$ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{gen}(\mathsf{pp})$
$b \leftarrow\$ \{0,1\}$
$\sigma^* \leftarrow\$ \{0,1\}^n$
$\mathsf{ct}_0^* \leftarrow\$ \mathsf{enc}^{\mathsf{p}}(\mathsf{pk}, \sigma^*)$
$K_0 \leftarrow H(\sigma^*); h^* \leftarrow H'(\sigma^*, \mathsf{ct}_0^*)$
$K_1 \leftarrow\$ \mathcal{K}$
$\mathsf{ct}^* \leftarrow (\mathsf{ct}_0^*, h^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Γ^0-Γ^1
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Dec2}}}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Γ^2-Γ^3
if query : **abort** // Γ^3
return $1_{b'=b}$

$H(\sigma)$

if $\exists h$ s.t. $(\sigma, h) \in \mathcal{L}_H$:
  return h
if $\sigma = \sigma^*$: query $\leftarrow$ **true** // Γ^3
$h \leftarrow\$ \{0,1\}^n$
$\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{(\sigma, h)\}$
return h

Oracle $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$ or $h = h^*$: // Γ^1-Γ^3
  return $\perp$ // Γ^1-Γ^3
$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}_0')$
if $H'(\sigma', \mathsf{ct}_0) \neq h$:
  return $\perp$
return $H(\sigma')$

$H'(\sigma, \mathsf{ct})$

if $\exists x, x', h$ s.t. $x \neq x'$
  $\wedge\ (x, h) \in \mathcal{L}_{H'}$
  $\wedge\ (x', h) \in \mathcal{L}_{H'}$:
  abort
if $\exists h$ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_{H'}$:
  return h
if $\sigma = \sigma^*$: query $\leftarrow$ **true** // Γ^3
$h \leftarrow\$ \{0,1\}^n$
$\mathcal{L}_{H'} \leftarrow \mathcal{L}_{H'} \cup \{((\sigma, \mathsf{ct}), h)\}$
return h

Oracle $\mathcal{O}^{\mathsf{Dec2}}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$ or $h = h^*$:
  return $\perp$
if $\exists \sigma$ s.t. $((\sigma, \mathsf{ct}_0), h) \in \mathcal{L}_{H'}$:
  if $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}_0)$:
    return $H(\sigma)$
return $\perp$

Fig. 5: Sequence of games.

loss of ≈ 1 bit compared to the OW-CPA advantage. Finally, we note that as q is a constant that does not depend on the security parameter (e.g. n) of the PKE, if the OW-CPA advantage of the PKE is negligible, so is the KEM IND-qCCA one. However, in practice, we would need to take n very large to guarantee security for more than a few queries.

Security in the QROM.

We also show that the T_{CH} transform is secure in the Quantum Random Oracle Model (QROM) by proving that Thm 1 holds in the QROM. First, we recall a variant of the well-known one-way to hiding lemma (OW2H) [20] as stated by Hofheinz et al. [11].

Lemma 2 (AOW2H [11]). *Let $\mathcal{A}$ be a quantum adversary making at most q_H queries to the QRO $|H\rangle : \{0,1\}^n \mapsto \{0,1\}^m$ and outputting 0 or 1. Let* $\mathsf{Ext}^{|H\rangle}_{q_H}(\mathcal{A})$

$\mathcal{B}^{\mathcal{O}^{\mathsf{PCO}}}(\mathsf{pk}, \mathsf{ct}^*)$

init $\mathcal{L}_H, \mathcal{L}_{H'} \leftarrow \emptyset$
$h^* \leftarrow\$ \{0,1\}^n$
$K^* \leftarrow\$ \{0,1\}^n$
simulate H, H' for $\mathcal{A}$ with lazy sampling:
run $\mathcal{A}^{H,H',\mathcal{O}^{\mathsf{Dec2}}}(\mathsf{pk}, (\mathsf{ct}^*, h^*), K^*)$
$\sigma' \leftarrow\$ \{\sigma : \sigma \in \mathcal{L}_H \vee (\sigma, *) \in \mathcal{L}_{H'}\}$
return σ'

Oracle $\mathcal{O}^{\mathsf{Dec2}}(\mathsf{ct})$

if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$ or $h = h^*$:
  return $\perp$
if $\exists \sigma$ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_{H'}$:
  if $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}_0)$: **return** $H(\sigma)$
return $\perp$

Fig. 6: $\mathcal{B}$ adversary for the proof of Thm 1.

$\mathsf{Ext}^{\mathcal{A},|H\rangle}(inp)$

$i \leftarrow\$ \{1, \ldots, q_H\}$
run $\mathcal{A}^{|H\rangle}(inp)$ until i-th query $|\mathsf{QUERY}_i\rangle$
$x' \leftarrow$ measure input register of $|\mathsf{QUERY}_i\rangle$
if $\mathcal{A}$ did not make i queries : **return** $\perp$
return x'

Fig. 7: Extractor Ext for the AOW2H lemma.

be the algorithm in Figure 7. Then, for any algorithm F *that does not use* $|H\rangle$

$$\begin{aligned}&\big|\Pr[\mathcal{A}^{|H\rangle}(inp) \Rightarrow 1 | \sigma^* \leftarrow\$ \{0,1\}^n; inp \leftarrow \mathsf{F}(\sigma^*, H(\sigma^*))] \\ &- \Pr[\mathcal{A}^{|H\rangle}(inp) \Rightarrow 1 | (\sigma^*, K) \leftarrow\$ \{0,1\}^{n+m}; inp \leftarrow \mathsf{F}(\sigma^*, K)]\big| \\ &\leq 2q_H \sqrt{\Pr[\sigma^* \leftarrow \mathsf{Ext}^{\mathcal{A},|H\rangle}(inp) | (\sigma^*, K) \leftarrow\$ \{0,1\}^{n+m}; inp \leftarrow \mathsf{F}(\sigma^*, K)}} \quad .\end{aligned}$$

We now prove that T_{CH} holds a secure IND-qCCA KEM in the QROM if the underlying PKE is OW-PCA.

Theorem 2. *We consider two quantum random oracles* $H, H' : \{0,1\}^* \mapsto \{0,1\}^n$*. Let* KEM *be the KEM resulting from applying the* T_{CH} *transform to a* PKE*. Then, for any IND-qCCA adversary* $\mathcal{A}$ *that makes at most* q_H *(resp.* $q_{H'}$*) quantum queries to* H *(resp.* H'*), there exists a OW-PCA adversary* $\mathcal{B}$ *s.t.*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \delta + 2(2q_{H'} + q_H + q) \cdot \sqrt{(2(2q_{H'} + q))^q \cdot \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B})}$$

where $\mathcal{B}$ *makes at most* q *queries to its plaintext-checking oracle.*

Proof. The proof is somewhat similar to the security proof of the $\mathsf{QU}^{\perp}_m$ transform of Hofheinz et al. [11]. However, we explicitly take care of some details seemingly not addressed in the original proof. In particular, the fact that the decapsulation

oracle also makes queries to the random oracles has to be taken into account when applying the OW2H lemma.

The beginning of the proof follows the same strategy as the classical one. The sequence of games is shown in Figure 8.

$\underline{\Gamma^0}$: This is the original IND-CCA game.

$\underline{\Gamma^1}$: The decapsulation oracle is modified s.t. it returns $\perp$ whenever $(\mathsf{ct}_0^*, *)$ is queried (note that (ct_0^*, h^*) cannot be submitted). This game is the same as Γ^0 except if the oracle in Γ^0 does not return $\perp$ on such queries. Now, let's assume $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct}_0^*, h \neq h^*) \neq \perp$. This happens only if

$$H'(\mathsf{dec}(\mathsf{sk}, \mathsf{ct}_0^*), \mathsf{ct}_0^*) = h \neq h^* = H'(\sigma^*, \mathsf{ct}_0^*) \ .$$

In turn, this implies that $\mathsf{dec}(\mathsf{sk}, \mathsf{ct}_0^*) \neq \sigma^*$ and thus it is a correctness error. Such an error happens at most with probability δ. Therefore, overall

$$|\Pr[\Gamma^0 \Rightarrow 1] - \Pr[\Gamma^1 \Rightarrow 1]| \leq \delta \ .$$

$\underline{\Gamma^2}$: We replace the challenge key K^* and tag h^* by random values. As the key is now always random we have

$$\Pr[\Gamma^2 \Rightarrow 1] = \frac{1}{2} \ .$$

We can consider H' as the combination of two ROs $H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}$ s.t.

$$H'(\sigma, \mathsf{ct}) := \begin{cases} H'_{\mathsf{ct}_0^*}(\sigma), & \text{if } \mathsf{ct} = \mathsf{ct}_0^* \\ H'_{\neq \mathsf{ct}_0^*}(\sigma, \mathsf{ct}), & \text{if } \mathsf{ct} \neq \mathsf{ct}_0^* \end{cases}$$

since all values of H' are uniformly and independently distributed. Moreover, one can simulate any quantum query to H' by 2 calls to the quantum random oracle $|H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}\rangle$. Thus, from now on, we assume the adversary can make $2q_{H'}$ queries to this quantum RO.

Then, by the OW2H lemma (Lemma 2) applied on $|H, H'_{\mathsf{ct}_0^*}\rangle$ with F as in Figure 9, we have

$$\Pr[\Gamma^1 \Rightarrow 1] - \Pr[\Gamma^2 \Rightarrow 1] \leq 2(2q_{H'} + q_H + q) \cdot \sqrt{\Pr[\Upsilon \Rightarrow 1]}$$

where Υ is the same game as Γ^2, except that we measure the input register of a random quantum query made to $|H, H'_{\mathsf{ct}_0^*}\rangle$ (by the adversary or the decapsulation oracle) and outputs 1 iff this is equal to the challenge seed σ^*. Note that the number of queries made to $H'_{\mathsf{ct}_0^*}$ throughout the game is at most $2q_{H'}$, as the decapsulation oracle never queries $H'(*, \mathsf{ct}_0^*)$ for tag verification (a decapsulation query $(\mathsf{ct}_0^*, *)$ immediately returns $\perp$). However, there can be

at most $q + q_H$ queries to H, as the decapsulation oracle might query H on a successful query.

$\underline{\Upsilon'}$: We modify Υ as follows. We replace $H'_{\neq \mathsf{ct}_0^*}$ by a random polynomial of degree $2(q+2q_{H'})$ over the field $\mathbb{F}_{2^n}$ (i.e. on a query m, we evalute $H'_{\neq \mathsf{ct}_0^*}(m)$). By Zhandry et al. [22], this is indistinguishable from a random oracle for adversaries making at most $q+2q_{H'}$ quantum queries to $H'_{\neq \mathsf{ct}_0^*}$, which is the case here. Then, we replace the decryption check in the decapsulation oracle by verifying whether a correct ciphertext is in the roots of the polynomial. More precisely, on a query $\mathcal{O}^{\mathsf{Dec2}}((\mathsf{ct}_0, h))$, we compute the list of roots of $H'_{\neq \mathsf{ct}_0^*}(X) - h$ and check whether there is a σ s.t. (ct_0, σ) is in the list and $\mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}_0) = \sigma$. If that is the case, we output $H(\sigma)$, otherwise we output $\perp$. We show that both oracles are equivalent:

- On a query (ct_0, h) s.t. $\mathsf{ct}_0 = \mathsf{ct}_0^*$ both oracles output $\perp$.
- $\mathcal{O}^{\mathsf{Dec2}}((\mathsf{ct}_0, h))$ outputs $H(\sigma)$: That means that $H'_{\neq \mathsf{ct}_0^*}(\sigma, \mathsf{ct}_0) = h$ and $\mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}_0) = \sigma$. Thus, the oracle $\mathcal{O}^{\mathsf{Dec}}$ would also output $H(\sigma)$.
- $\mathcal{O}^{\mathsf{Dec2}}((\mathsf{ct}_0, h))$ outputs $\perp$: Let $\sigma := \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}_0)$. The oracle returning $\perp$ means that (σ, ct_0) is not in the roots of $H'_{\neq \mathsf{ct}_0^*}(X) - h$, thus $H'_{\neq \mathsf{ct}_0^*}(\sigma, \mathsf{ct}_0) \neq h$ or $\sigma = \perp$. Hence, the original oracle $\mathcal{O}^{\mathsf{Dec}}$ would also return $\perp$.

Thus, $\mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}}$ and $\mathcal{A}^{\mathcal{O}^{\mathsf{Dec2}}}$ have identical behaviour. In particular, the queries made to $|H, H'_{\mathsf{ct}_0^*}\rangle$ are the same (for fixed randomness). Hence, we have

$$\Pr[\Upsilon \Rightarrow 1] = \Pr[\Upsilon' \Rightarrow 1] \ .$$

We note that $\mathcal{O}^{\mathsf{Dec2}}$ makes no query to $H'_{\neq \mathsf{ct}_0^*}$, contrary to the original decapsulation oracle $\mathcal{O}^{\mathsf{Dec}}$, which makes "classical" queries to $H'_{\neq \mathsf{ct}_0^*}$ to verify the tag. However, this has no impact on $\Pr[\Upsilon' \Rightarrow 1]$ as the extractor measures only a random query made to $|H, H'_{\mathsf{ct}_0^*}\rangle$ (and not to $|H'_{\neq \mathsf{ct}_0^*}\rangle$).

Finally, one can build an OW-PCA adversary $\mathcal{B}$ s.t. $\frac{1}{(2(2q_{H'}+q))^q} \Pr[\Upsilon' \Rightarrow 1] \leq \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B})$. The adversary $\mathcal{B}$ is given in Figure 10. In particular, $\mathcal{B}$ can perfectly simulate $\mathsf{Ext}^{\mathcal{A}^{\mathcal{O}^{\mathsf{Dec2}}}}$ as the modified decapsulation oracle only requires a plaintext-checking oracle. However, in order to limit the number of queries to $\mathcal{O}^{\mathsf{PCO}}$, $\mathcal{B}$ makes some guessing. I.e. after computing the list of roots, it picks one (σ, ct_0) at random and checks whether $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}_0)$ holds or not. The probability to pick the correct root (if it exists) is $\frac{1}{2(2q_{H'}+q)}$ as $H'_{\neq \mathsf{ct}_0^*}$ is a polynomial of degree $2(2q_{H'}+q)$. Hence, overall the simulation is perfect with probability $\frac{1}{(2(2q_{H'}+q))^q}$ and $\mathcal{B}$ recovers σ^* with probability $\Pr[\Upsilon' \Rightarrow 1]$, thus

$$\Pr[\Upsilon' \Rightarrow 1] \leq (2(2q_{H'} + q))^q \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

In addition, $\mathcal{B}$ makes at most q queries to the plaintext-checking oracle. Collecting the probabilities holds the result. □

$\Gamma^{0-2}, \Upsilon, \Upsilon'(\mathcal{A})$

$H'_{\neq \mathsf{ct}_0^*} \leftarrow\$ \{\text{polynomials of deg. } 2(q+2q_{H'})\}$ // Υ'
$\mathsf{pp} \leftarrow\$ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{gen}(\mathsf{pp})$
$b \leftarrow\$ \{0,1\}$
$\sigma^* \leftarrow\$ \{0,1\}^n$
$K_0 \leftarrow H(\sigma^*); \mathsf{ct}_0^* \leftarrow\$ \mathsf{enc}^{\mathsf{p}}(\mathsf{pk}, \sigma^*)$
$h^* \leftarrow H'(\sigma^*, \mathsf{ct}_0^*)$
$K_0 \leftarrow\$ \{0,1\}^n; h^* \leftarrow\$ \{0,1\}^n$ // Γ^2,Υ,Υ'
$K_1 \leftarrow\$ \{0,1\}^n$
$\mathsf{ct}^* \leftarrow (\mathsf{ct}_0^*, h^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}, |H\rangle, |H'\rangle}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Γ^0-Γ^1
$b' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}, |H\rangle, |H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}\rangle}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Γ^2
return $1_{b'=b}$ // Γ^0-Γ^2
$\sigma' \leftarrow\$ \mathsf{Ext}^{\mathcal{A}^{\mathcal{O}^{\mathsf{Dec}}, |H\rangle, |H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}\rangle}}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Υ
$\sigma' \leftarrow\$ \mathsf{Ext}^{\mathcal{A}^{\mathcal{O}^{\mathsf{Dec2}}, |H\rangle, |H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}\rangle}}(\mathsf{pk}, \mathsf{ct}^*, K_b)$ // Υ'
return $1_{\sigma'=\sigma^*}$ // Υ-Υ'

$H'(\sigma, \mathsf{ct})$

use a standard QRO to reply // Γ^0-Γ^1
use two QROs $H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}$: // $\Gamma^2, \Upsilon, \Upsilon'$
if $\mathsf{ct} = \mathsf{ct}^*$: **return** $H'_{\mathsf{ct}_0^*}(\sigma)$ // $\Gamma^2, \Upsilon, \Upsilon'$
return $H'_{\neq \mathsf{ct}_0^*}(\sigma, \mathsf{ct})$ // $\Gamma^2, \Upsilon, \Upsilon'$

Oracle $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$: // Γ^1-Γ^2,Υ
  return $\perp$ // Γ^1-Γ^2,Υ
$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct}'_0)$
if $H'(\sigma', \mathsf{ct}_0) \neq h$:
  return $\perp$
return $H(\sigma')$

Oracle $\mathcal{O}^{\mathsf{Dec2}}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$:
  return $\perp$
if $\exists(\sigma, \mathsf{ct}_0) \in \mathsf{Roots}(H'_{\neq \mathsf{ct}_0^*}(X) - h)$
  s.t. $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}_0)$:
  return $H(\sigma)$
return $\perp$

Fig. 8: Sequence of games for Thm 2.

Corollary 2. *We consider two quantum random oracles $H, H' : \{0,1\}^* \mapsto \{0,1\}^n$. Let* KEM *be the KEM resulting from applying the* T_{CH} *transform to a δ-correct* PKE*. Then, for any IND-qCCA adversary $\mathcal{A}$ that makes at most q_H (resp. $q_{H'}$) queries to H (resp. H'), there exists a OW-CPA adversary $\mathcal{B}$ s.t.*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \delta + 2(2q_{H'} + q_H + q) \cdot 2^q \sqrt{((2q_{H'}+q))^q \cdot \mathsf{Adv}^{\mathrm{ow-pca}}_{\mathsf{PKE}}(\mathcal{B})} \ .$$

Hashing the plaintext and ciphertext

One can also wonder what is the leakage of the decapsulation oracle in the ROM, when the key is simply the hash of the seed and the plaintext. That is, we consider the simple PKE to KEM transform given in Figure 11, which we call T_{H}. Note that this is the same transform as the one called $\mathsf{U}^{\perp}$ in [11]. We now

$\mathsf{F}(\sigma^*, (K^*, h^*))$

$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(\mathsf{pp})$
$\mathsf{ct}_0^* \leftarrow\!\!\$\ \mathsf{enc}^{\mathsf{P}}(\mathsf{pk}, \sigma^*)$
$\mathsf{ct}^* \leftarrow (\mathsf{ct}_0^*, h^*)$
return $(\mathsf{pk}, \mathsf{ct}^*, K^*)$

Fig. 9: F function for applying the AOW2H lemma in the proof of Thm 2.

$\mathcal{B}^{\mathcal{O}^{\mathsf{PCO}}}(\mathsf{pk}, \mathsf{ct}_0^*)$

$h^* \leftarrow\!\!\$\ \{0,1\}^n$
$K^* \leftarrow\!\!\$\ \{0,1\}^n$
simulate $H, H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}$
for $\mathcal{A}$ with random polynomials:
$\sigma' \leftarrow\!\!\$\ \mathsf{Ext}^{\mathcal{A}^{\mathcal{O}^{\mathsf{Dec2}}, |H\rangle, |H'_{\mathsf{ct}_0^*}, H'_{\neq \mathsf{ct}_0^*}\rangle}}(\mathsf{pk}, (\mathsf{ct}_0^*, h^*), K^*)$
return σ'

Oracle $\mathcal{O}^{\mathsf{Dec2}}(\mathsf{ct})$

if more than q queries :
  return $\perp$
$(\mathsf{ct}_0, h) \leftarrow \mathsf{ct}$
if $\mathsf{ct}_0 = \mathsf{ct}_0^*$:
  return $\perp$
$\mathcal{L} \leftarrow \{(\sigma, \mathsf{ct}') : (\sigma, \mathsf{ct}') \in \mathsf{Roots}(H'_{\neq \mathsf{ct}_0^*} - h) \wedge \mathsf{ct}' = \mathsf{ct}_0\}$
$(\sigma, \mathsf{ct}_0) \leftarrow\!\!\$\ \mathcal{L}$
if $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct}_0)$:
  return $H(\sigma)$
return $\perp$

Fig. 10: $\mathcal{B}$ adversary for the proof of Thm 2.

show that if q is small (logarithmic in the security parameter), the T_{H} holds a secure (and practical) IND-qCCA scheme in the ROM, given that the underlying PKE is OW-CPA.

Theorem 3. *We consider a random oracle $H : \{0,1\}^* \mapsto \{0,1\}^n$. Let* KEM *be the KEM resulting from applying the* T_{H} *transform to a PKE* PKE *(which never queries H). Then, for any IND-qCCA adversary $\mathcal{A}$ that makes at most q_H queries to H, there exists a OW-CPA adversary $\mathcal{B}$ s.t.*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq q_H \cdot ((q_H+1)(q_H+2))^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

If PKE *is deterministic, we get*

$$\mathsf{Adv}^{\mathrm{ind-qcca}}_{\mathsf{KEM}}(\mathcal{A}) \leq \delta + ((q_H+1)(q_H+2))^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Proof. We start by defining an oracle $\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$ (see Figure 12). This oracle returns the index i s.t. $((\sigma_i, \mathsf{ct}_i), K) \in \mathcal{L}_H$ (we first sort $\mathcal{L}_H$ according to some fixed order) and $\mathsf{ct}_i = \mathsf{ct}$ and $\mathsf{dec}^{\mathsf{P}}(\mathsf{sk}, \mathsf{ct}_i) = \sigma_i$. If such a i does not exist and $\mathsf{dec}^{\mathsf{P}}(\mathsf{sk}, \mathsf{ct}_i) = \perp$ it returns $\perp_d$, otherwise it returns $\perp$.

$\mathsf{gen}()$	$\mathsf{encaps}(\mathsf{pk})$	$\mathsf{decaps}(\mathsf{sk}, \mathsf{ct})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{gen}^{\mathsf{p}}()$	$\sigma \leftarrow_\$ \mathcal{M}$	$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct})$
return $(\mathsf{pk}, \mathsf{sk})$	$\mathsf{ct} \leftarrow_\$ \mathsf{enc}^{\mathsf{p}}(\mathsf{pk}, \sigma)$	**if** $\sigma' = \perp$: **return** $\perp$
	$K \leftarrow H(\sigma, \mathsf{ct})$	**return** $H(\sigma', \mathsf{ct})$
	return K, ct	

Fig. 11: $\mathsf{T_H}$ transform.

$\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$

sort $\mathcal{L}_H$ according to query order :
$\mathcal{L}_H = ((\sigma_i, \mathsf{ct}_i), K_i)_{i \in \{1,\dots,|\mathcal{L}_H|\}}$
$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct})$
if $\sigma' = \perp$: **return** $\perp_d$
for $i \in \{1, \dots, |\mathcal{L}_H|\}$:
  if $\mathsf{ct}_i = \mathsf{ct}$ **and** $\sigma' = \sigma_i$:
    return i
return $\perp$

Fig. 12: $\mathcal{O}^{\mathsf{i}}$ oracle for the proof of Thm 3.

Now we show how to simulate the IND-qCCA decapsulation oracle in the ROM, using $\mathcal{O}^i$ and $\mathcal{O}^{\mathsf{PCO}}$ only. The original (resp. modified) oracles $\mathcal{O}^{\mathsf{Dec}}$ and H (resp. $\mathcal{O}^{\mathsf{Dec}'}$ and H') are on the left (resp. right) in Figure 13. We now prove that any IND-qCCA adversary cannot distinguish between the real and modified oracles.

First, we show that the outputs of the ROs H and H' on any query (σ, ct) have the same distribution, given the adversary's view. We break this into four subcases:

- (σ, ct) was queried before to H (resp. H'): In this case, both H and H' return the value h returned on the previous similar query. Thus, we assume from now on that every RO query made by the adversary is unique.
- ct was never queried to the decapsulation oracle before: In this case, both H and H' return a random value h and store the query/response in $\mathcal{L}_H$.
- ct was queried to the decapsulation oracle before: In both cases (original and modified oracles) one can see that if the decryption of ct either fails or $\sigma' = \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct})$ is different from σ, then the output of the decapsulation oracle is independent of $H(\sigma, \mathsf{ct})$ (and $H'(\sigma, \mathsf{ct})$). In both cases, the ROs sample a fresh value (H' will do so because $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct})$ will output 0 in this case, as $\sigma \neq \sigma'$ or the ciphertext is not valid). Now, if ct decrypts to σ, the original decapsulation oracle outputs $H(\sigma, \mathsf{ct})$. In the modified game, the decapsulation oracle outputs a random K. Indeed, as we assume (σ, ct)

was never queried to H, $\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$ outputs $\perp$. Then, the modified RO will output the same K, as $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct})$ will verify. In both cases, the ROs output the same value as the decapsulation oracle.

We now show that the decapsulation oracles $\mathcal{O}^{\mathsf{Dec}}$ and $\mathcal{O}^{\mathsf{Dec}'}$ are indistinguishable. Let ct be the queried ciphertext and $\sigma = \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct})$.

- $\mathsf{ct} = \mathsf{ct}^*$: both oracles return $\perp$.
- $\sigma = \perp$: Both oracles return $\perp$, as $\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$ returns $\perp_d$.
- $H(\sigma, \mathsf{ct})$ (resp. $H'(\sigma, \mathsf{ct})$) was never queried. Both oracles return a random value if ct was never queried, or a consistent value if it was. It is straightforward to see this is the case in the original oracle. In the modified oracle, as $H'(\sigma, \mathsf{ct})$ was never queried, we have $\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$ that returns $\perp$. Thus, the decapsulation oracle returns a random K if ct was not queried or a consistent K if it was.
- $H(\sigma, \mathsf{ct})$ (resp. $H'(\sigma, \mathsf{ct})$) was queried and it output K. Both oracles return K. In the modified decapsulation oracle, $\mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$ will output a valid i s.t. $H'(\sigma_i, \mathsf{ct}) = h_i$ and h_i is returned. Thus, the answer is consistent with the RO.

Now we can prove the theorem by game hopping as before. We define Γ^0 as the original IND-qCCA game.

$\underline{\Gamma^1:}$ We modify the original IND-qCCA game into another game Γ^1 where the random/decapsulation oracles are the modified ones (i.e. H' and $\mathcal{O}^{\mathsf{Dec}'}$) described above. As shown, both games are indistinguishable and thus

$$|\Pr[\Gamma^0 \Rightarrow 1] - \Pr[\Gamma^1 \Rightarrow 1]| = 0 \ .$$

$\underline{\Gamma^2:}$ We replace the challenge key by a random one, as in the previous proof. Then, similarly, the real key is indistinguishable from a random one unless $H(\sigma^*, \mathsf{ct}^*)$ is queried. We define this event as query and

$$|\Pr[\Gamma^1 \Rightarrow 1] - \Pr[\Gamma^2 \Rightarrow 1]| \leq \Pr[\mathsf{query}] \ .$$

We can upper bound this probability by the advantage of a OW-CPA adversary $\mathcal{B}$ againt PKE. That is, given a IND-qCCA adversary playing game Γ^2, we build an adversary $\mathcal{B}$ as shown in Figure 14. One can see that if $\mathcal{B}$ was simulating $\mathcal{A}$ with the H' and $\mathcal{O}^{\mathsf{Dec}'}$ oracles (instead of its own oracles H'' and $\mathcal{O}^{\mathsf{Dec}''}$), the simulation would be perfect as long as query did not occur. Then, whenever query would happen, $\mathcal{B}$ would recover σ^* with prob. $\frac{1}{q_H}$. Now $\mathcal{B}$ does not simulate the modified oracles perfectly but instead makes some guessing in its own oracles H'' and $\mathcal{O}^{\mathsf{Dec}''}$:

- $\mathcal{O}^{\mathsf{Dec}''}$: In line 5, i is picked at random instead of being the returned value of the $\mathcal{O}^{\mathsf{i}}$ oracle. On each query the simulation is perfect with prob. $1/(q_H + 2)$ and overall with probability $\frac{1}{(q_H+2)^q}$, as there are at most q queries to this oracle. In line 11, we associate a random index to each ct s.t. $(\mathsf{ct}, *) \in \mathcal{L}_K$.

- H'': In line 5, when $(\mathsf{ct}, *) \in \mathcal{L}_K$, instead of querying the plaintext-checking oracle we check whether the corresponding sampled index $\mathcal{L}_q[\mathsf{ct}]$ is equal to the query number. If it is, we reply with K s.t. $(\mathsf{ct}, K) \in \mathcal{L}_K$ otherwise we proceed as before (i.e. as in H'). Let's assume w.l.o.g that each query to H'' is unique. For each ct s.t. $(\mathsf{ct}, *) \in \mathcal{L}_K$, there can be at most one query (σ, ct) s.t. $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct})$ returns 1 (it is when σ is the decryption of ct). Here, $\mathcal{B}$ guesses beforehand which query it is (or if no such query will be made) and gets the correct answer with prob. $\frac{1}{q_H+1}$. Overall, the probability H'' simulates correctly H' is $\frac{1}{(q_H+1)^q}$.

From this we can deduce that $\mathcal{B}$ correctly simulates Γ^2 with probability $\frac{1}{((q_H+1)(q_H+2))^q}$ and wins the OW-CPA game with prob. at least $\frac{1}{q_H} \cdot \Pr[\mathsf{query}]$. Hence,

$$|\Pr[\Gamma^1 \Rightarrow 1] - \Pr[\Gamma^2 \Rightarrow 1]| \leq \Pr[\mathsf{query}] \leq q_H \cdot ((q_H+1)(q_H+2))^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Note that when PKE is deterministic, in order to recover σ^*, $\mathcal{B}$ can check which σ' queried is s.t. $\mathsf{enc}(\mathsf{pk}^*, \sigma') = \mathsf{ct}^*$. This works as long as the challenge ciphertext is correct, which happens with prob. $1 - \delta$. Therefore, for deterministic PKEs, the last inequality becomes

$$|\Pr[\Gamma^1 \Rightarrow 1] - \Pr[\Gamma^2 \Rightarrow 1]| \leq \Pr[\mathsf{query}] \leq \delta + ((q_H+1)(q_H+2))^q \cdot \mathsf{Adv}^{\mathrm{ow-cpa}}_{\mathsf{PKE}}(\mathcal{B}) \ .$$

Finally, in game Γ^2, the challenge key is always random and thus $\Pr[\Gamma^2 \Rightarrow 1] = \frac{1}{2}$. Collecting the probabilities holds the result. □

4 Impact

The simple constructions considered above produce IND-qCCA KEMs without any derandomization and re-encryption steps. Thus, using IND-1CCA ephemeral KEMs obtained through these transforms would speed up the decapsulation process in several systems that we list here.

KEMTLS. As discussed in the introduction, improving the KEMTLS protocol [17] was the main motivation of this work. In particular, a more efficient decapsulation in the ephemeral KEM would imply less overall latency and less computation on the client side. In particular, this could be of interest for less powerful clients like IoT devices, which would not need to perform re-encryption. Overall, the efficiency gain in practice would obviously depend on the ephemeral KEM used, as encryption is expensive in some schemes while it is not in others. For instance, using KEMTLS with a modified version of SIKE (i.e. obtained through our transform instead of the FO one) would probably reduce significantly the latency.

The same remarks apply to the very recent variants of KEMTLS with pre-distributed keys proposed by Günther et al. [10] and Schwabe et al. [19].

Oracle $\mathcal{O}^{\mathsf{Dec}}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries :
  return $\perp$
$\sigma' \leftarrow \mathsf{dec}^{\mathsf{p}}(\mathsf{sk}, \mathsf{ct})$
if $\sigma' = \perp$: **return** $\perp$
return $H(\sigma', \mathsf{ct})$

$H(\sigma, \mathsf{ct})$

if $\exists h$ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_H$:
  return h
$h \leftarrow\!\!\$\ \{0,1\}^n$
$\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{((\sigma, \mathsf{ct}), h)\}$
return h

Oracle $\mathcal{O}^{\mathsf{Dec}'}(\mathsf{ct})$

if $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
if more than q queries : **return** $\perp$
if $\exists K$ s.t. $(\mathsf{ct}, K) \in \mathcal{L}_K$:
  return K
$i \leftarrow \mathcal{O}^{\mathsf{i}}(\mathcal{L}_H, \mathsf{ct})$
if $i = \perp_d$: **return** $\perp$
if $i \neq \perp$:
  $(\mathsf{ct}_i, K_i) \leftarrow \mathcal{L}_H[i]$
  return K_i // return i-th valued returned by H'
$K \leftarrow\!\!\$\ \{0,1\}$
$\mathcal{L}_K \leftarrow \mathcal{L}_K \cup \{(\mathsf{ct}, K)\}$
return K

$H'(\sigma, \mathsf{ct})$

if $\exists h$ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_H$:
  return h
if $\exists K$ s.t. $(\mathsf{ct}, K) \in \mathcal{L}_K$:
  if $\mathcal{O}^{\mathsf{PCO}}(\sigma, \mathsf{ct})$:
    $\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{((\sigma, \mathsf{ct}), K)\}$
    return K
$h \leftarrow\!\!\$\ \{0,1\}^n$
$\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{((\sigma, \mathsf{ct}), h)\}$
return h

Fig. 13: Original and modified oracles for the proof of Thm 3.

TLS 1.3. TLS 1.3 only supports ephemeral DH as a key-exchange. In turn, in the security proof (e.g. [6]), the PRF-ODH assumption is sufficient for the key-exchange security. The PRF-ODH assumption can be seen as a variant of the hashed Diffie-Hellman assumption with a 1-time "decapsulation" oracle. More precisely, an adversary is given (g, g^u, g^v) and either $y_0 := \mathsf{PRF}(g^{uv}, \mathsf{ad}^*)$ or a random y_1, where ad^* is some auxiliary data chosen by the adversary. Then, the adversary must distinguish between y_0 and y_1 with the help of *one* query to an oracle $\mathcal{O}((x, \mathsf{ad}) \neq (g^v, \mathsf{ad}^*)) := \mathsf{PRF}(x^v, \mathsf{ad})$.

One can see that this is very close to IND-1CCA security transposed to DH key-exchange. Thus, if we consider TLS 1.3 modified s.t. it supports KEM (as in the PQ openssl fork [1,14]), we conjecture that the proof should hold if the KEM is IND-1CCA secure. For instance, in the Multi-Stage security proof of TLS 1.3 1-RTT handshake of Dowling et al. [6], the only step of the proof impacted should be the transition **Game B.1** $\rightarrow$ **Game B.2.** that uses the PRF-ODH assumption. Replacing this step using the IND-1CCA property of the KEM should suffice for the proof to hold. Hence, if this is the case, PQ TLS

$\mathcal{B}(\mathsf{pk}, \mathsf{ct}^*)$

init $\mathcal{L}_H, \mathcal{L}_K \leftarrow \emptyset$
init $\mathcal{L}_q \leftarrow []$
$K^* \leftarrow\!\!\$\, \mathcal{K}$
run $\mathcal{A}^{H'', \mathcal{O}^{\mathsf{Dec}''}}(\mathsf{pk}, \mathsf{ct}^*, K^*)$
sample random query (σ', ct') made to H''
return σ'

$H''(\sigma, \mathsf{ct})$

1 : $i_q \leftarrow$ query number
2 : **if** $\exists h$ s.t. $((\sigma, \mathsf{ct}), h) \in \mathcal{L}_H$:
3 : **return** h
4 : **if** $\exists K$ s.t. $(\mathsf{ct}, K) \in \mathcal{L}_K$:
5 : **if** $\mathcal{L}_q[\mathsf{ct}] = i_q$:
6 : $\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{((\sigma, \mathsf{ct}), K)\}$
7 : **return** K
8 : $h \leftarrow\!\!\$\, \{0,1\}^n$
9 : $\mathcal{L}_H \leftarrow \mathcal{L}_H \cup \{((\sigma, \mathsf{ct}), h)\}$
10 : **return** h

Oracle $\mathcal{O}^{\mathsf{Dec}''}(\mathsf{ct})$

1 : **if** $\mathsf{ct} = \mathsf{ct}^*$: **return** $\perp$
2 : **if** more than q queries : **return** $\perp$
3 : **if** $\exists K$ s.t. $(\mathsf{ct}, K) \in \mathcal{L}_K$:
4 : **return** K
5 : $i \leftarrow\!\!\$\, \{1, \ldots, q_H, \perp, \perp_d\}$
6 : **if** $i = \perp_d$: **return** $\perp$
7 : **if** $i \neq \perp$:
8 : $(\mathsf{ct}_i, K_i) \leftarrow \mathcal{L}_H[i]$
9 : **return** K_i // return i-th valued returned by H''
10 : $K \leftarrow\!\!\$\, \{0,1\}$
11 : $\mathcal{L}_K \leftarrow \mathcal{L}_K \cup \{(\mathsf{ct}, K)\}$
12 : $\mathcal{L}_q[\mathsf{ct}] \leftarrow\!\!\$\, \{0, \ldots, q_H\}$
13 : **return** K

Fig. 14: $\mathcal{B}$ adversary for the proof of Thm 3.

1.3 could also be boosted by using PQ IND-1CCA KEMs instead of the slower IND-CCA KEMs derived with the FO transform.

Ratcheting. IND-1CCA security is also a property used (often implicitly) in several works on ratcheting. For instance, Jost et al. [12] build a *healable and key-updating public-key encryption* scheme based on a *one time* IND-CCA2 PKE (with authenticated data). The latter primitive can easily be made out of an IND-1CCA KEM using KEM/DEM techniques. Another paper by Poettering et al. [16] introduces a construction of *unidirectional ratcheted key exchange* (URKE) that is based (implicitly) on IND-1CCA KEMs, as noticed by Balli et al. [2].

In another recent paper, Brendel et al. [4] propose an alternative to the Signal handshake based on KEMs and designated verifier signature schemes. They first define a core protocol that uses two KEMs in the same vein as KEMTLS: one with long-term keys for implicit authentication of one of the parties and another one with ephemeral keys to guarantee forward security. Again, the latter one requires only IND-1CCA security for the handshake to be secure. Similarly, in the full Signal-like handshake built upon the core protocol (called SPQR), three KEMs are used and one requires only IND-1CCA security.

Concerns over key-reuse. The main security risk of using an IND-1CCA KEM instead of its IND-CCA counterpart is the vulnerability to key-reuse/misuse attacks. Indeed, if a system/protocol is misimplemented s.t. the

IND-1CCA KEM is used with a "static" public key instead of an ephemeral one, an adversary might be able to recover the secret key after several decryption queries. In KEMTLS, this risk is mitigated by the use of an IND-CCA KEM in addition to the ephemeral one (which can be IND-1CCA). In particular, the final shared key is derived from shares of both KEMs. Thus, even if the public-key meant to be ephemeral is reused, the final shared key should remain secure.

In other systems (e.g. TLS 1.3), this risk could be mitigated by using hybrid cryptography. For instance, a very efficient IND-CCA KEM could be combined with an IND-1CCA one. That would improve the overall security and resistance against key-reuse attacks at a small cost (see e.g. Giacon et al. [9] or Bindel et al. [3] for KEM combiners). Finally, note that if ephemeral keys were misimplemented as static ones in these systems, the forward security property would be lost.

Conclusion. Ratcheting and several recent protocols (e.g. TLS 1.3) are aiming at forward security, which often implies generating a new pair of public/secret keys for each message exchanged. Informally, in many settings this means that an adversary requesting a decryption will be able to do so only once for a given key pair. Thus, IND-1CCA security of the underlying encryption/encapsulation primitive might be sufficient to guarantee the security of such systems.

Acknowledgments. We thank Daniel Collins for pointing out possible use-cases of IND-1CCA KEMs in ratcheting. Loïs Huguenin-Dumittan is supported by a grant (project № 192364) of the Swiss National Science Foundation (SNSF).

References

1. OQS OpenSSL, https://github.com/open-quantum-safe/openssl, June 2021
2. Balli, F., Rösler, P., Vaudenay, S.: Determining the core primitive for optimally secure ratcheting. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 621–650. Springer (2020)
3. Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., Stebila, D.: Hybrid key encapsulation mechanisms and authenticated key exchange. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 206–226. Springer International Publishing, Cham (2019)
4. Brendel, J., Fiedler, R., Gnther, F., Janson, C., Stebila, D.: Post-quantum asynchronous deniable key exchange and the signal handshake. Cryptology ePrint Archive, Report 2021/769 (2021), https://eprint.iacr.org/2021/769
5. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 502–518. Springer (2007)
6. Dowling, B., Fischlin, M., Günther, F., Stebila, D.: A cryptographic analysis of the TLS 1.3 handshake protocol. Journal of Cryptology (2020)
7. Fluhrer, S.: Cryptanalysis of ring-lwe based key exchange with key share reuse. Cryptology ePrint Archive, Report 2016/085 (2016), https://eprint.iacr.org/2016/085

8. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology **26**(1), 80–101 (2013), https://doi.org/10.1007/s00145-011-9114-1
9. Giacon, F., Heuer, F., Poettering, B.: KEM Combiners. Cryptology ePrint Archive, Report 2018/024 (2018), https://eprint.iacr.org/2018/024
10. Gnther, F., Towa, P.: KEMTLS with delayed forward identity protection in (almost) a single round trip. Cryptology ePrint Archive, Report 2021/725 (2021), https://eprint.iacr.org/2021/725
11. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Theory of Cryptography Conference. pp. 341–371. Springer (2017)
12. Jost, D., Maurer, U., Mularczyk, M.: Efficient ratcheting: Almost-optimal guarantees for secure messaging. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 159–188. Springer (2019)
13. Okamoto, T., Pointcheval, D.: REACT: Rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) Topics in Cryptology — CT-RSA 2001. pp. 159–174. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
14. Paquin, C., Stebila, D., Tamvada, G.: Benchmarking post-quantum cryptography in tls. Cryptology ePrint Archive, Report 2019/1447 (2019), https://eprint.iacr.org/2019/1447
15. Pereira, M., Dowsley, R., Hanaoka, G., Nascimento, A.C.: Public key encryption schemes with bounded CCA security and optimal ciphertext length based on the CDH assumption. In: International Conference on Information Security. pp. 299–306. Springer (2010)
16. Poettering, B., Rösler, P.: Towards bidirectional ratcheted key exchange. In: Annual International Cryptology Conference. pp. 3–32. Springer (2018)
17. Schwabe, P., Stebila, D., Wiggers, T.: Post-quantum TLS without handshake signatures. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 1461–1480 (2020)
18. Schwabe, P., Stebila, D., Wiggers, T.: Post-quantum TLS without handshake signatures. Cryptology ePrint Archive, Report 2020/534 (2020), https://eprint.iacr.org/2020/534
19. Schwabe, P., Stebila, D., Wiggers, T.: More efficient post-quantum KEMTLS with pre-distributed public keys. Cryptology ePrint Archive, Report 2021/779 (2021), https://eprint.iacr.org/2021/779
20. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM (JACM) **62**(6), 1–76 (2015)
21. Yamakawa, T., Yamada, S., Matsuda, T., Hanaoka, G., Kunihiro, N.: Reducing public key sizes in bounded CCA-secure KEMs with optimal ciphertext length. In: Information Security, pp. 100–109. Springer (2015)
22. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Annual Cryptology Conference. pp. 758–775. Springer (2012)