CCA-Secure (Puncturable) KEMs from Encryption With Non-Negligible Decryption Errors

Valerio Cini, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks

AIT Austrian Institute of Technology, Vienna, Austria
{firstname.lastname}@ait.ac.at

Abstract. Public-key encryption (PKE) schemes or key-encapsulation mechanisms (KEMs) are fundamental cryptographic building blocks to realize secure communication protocols. There are several known transformations that generically turn weakly secure schemes into strongly (i.e., IND-CCA) secure ones. While most of these transformations require the weakly secure scheme to provide perfect correctness, Hofheinz, Hövelmanns, and Kiltz (HHK) (TCC 2017) have recently shown that variants of the Fujisaki-Okamoto (FO) transform can work with schemes that have negligible correctness error in the (quantum) random oracle model (QROM). Many recent schemes in the NIST post-quantum competition (PQC) use variants of these transformations. Some of their CPA-secure versions even have a non-negligible correctness error and so the techniques of HHK cannot be applied.

In this work, we study the setting of generically transforming PKE schemes with potentially large, i.e., non-negligible, correctness error to ones having negligible correctness error. While there have been previous treatments in an asymptotic setting by Dwork et al. (EUROCRYPT 2004), our goal is to come up with practically efficient compilers in a concrete setting and apply them in two different contexts: firstly, we show how to generically transform weakly secure deterministic or randomized PKEs into CCA-secure KEMs in the (Q)ROM using variants of HHK. This applies to essentially all candidates to the NIST PQC based on lattices and codes with non-negligible error, for which we provide an extensive analysis. We thereby show that it improves some of the code-based candidates. Secondly, we study puncturable KEMs in terms of the Bloom Filter KEM (BFKEM) proposed by Derler et al. (EUROCRYPT 2018) which inherently have a non-negligible correctness error. BFKEMs are a building block to construct fully forward-secret zero round-trip time (0-RTT) key-exchange protocols. In particular, we show how to achieve the first post-quantum secure BFKEM generically from lattices and codes by applying our techniques to identity-based encryption (IBE) schemes with (non-)negligible correctness error.

Keywords: CPA-to-CCA transformations, Fujisaki-Okamoto transform, non-negligible correctness error, puncturable encryption

1 Introduction

Public-key encryption (PKE) schemes or key-encapsulation mechanisms (KEM) are fundamental cryptographic building blocks to realize secure communication protocols. The security property considered standard nowadays is security against chosen-ciphertext attacks (IND-CCA security). This is important to avoid pitfalls and attacks in the practical deployments of such schemes, e.g., padding oracle attacks as demonstrated by Bleichenbacher [12] and still showing up very frequently [46, 5, 14, 57]. Also, for key exchange protocols that achieve the desirable forward secrecy property, formal analysis shows that security against active attacks is required (cf. [45, 50, 22, 56]). This equally holds for recent proposals for fully forward-secret zero round-trip time (0-RTT) key-exchange protocols from puncturable KEMs [34, 21, 20] and even for ephemeral KEM keys for a post-quantum secure TLS handshake without signatures [61].

In the literature, various different ways of obtaining CCA security generically from weaker encryption schemes providing only chosen-plaintext (IND-CPA) or one-way (OW-CPA) security are known. These can be in the standard model using the double-encryption paradigm due to Naor and Yung [54], the compiler from selectively secure identity-based encryption (IBE) due to Canetti, Halevi and Katz [18], or the more recent works due to Koppula and Waters [49] based on so called hinting pseudo-random generators and Hohenberger, Koppula, and Waters [42] from injective trapdoor functions. In the random oracle model (ROM), CCA security can be generically obtained via the well-known and widely-used Fujisaki-Okamoto (FO) transform [27, 28] yielding particularly practical efficiency.

Perfect correctness and (non-)negligible correctness error. A property common to many compilers is the requirement for the underlying encryption schemes to provide perfect correctness, i.e., there are no valid ciphertexts where the decryption algorithm fails when used with honestly generated keys. Recently, Hofheinz, Hövelmanns and Kiltz (HHK) [40] investigated different variants of the FO transform also in a setting where the underlying encryption scheme has non-perfect correctness and in particular decryption errors may occur with a negligible probability in the security parameter. This is interesting since many PKE or KEM schemes based on conjectured quantum safe assumptions and in particular assumptions on lattices and codes do not provide perfect correctness. Even worse, some of the candidates submitted to the NIST post-quantum competition (PQC) suffer from a *non-negligible* correctness error and so the FO transforms of HHK cannot be applied. Ad-hoc approaches to overcome this problem that are usually chosen by existing constructions in practice — if the problem is considered at all — is to increase the parameters to obtain a suitably small decryption error, applying an error correcting code on top or implementing more complex decoders. In practice, these ad-hoc methods come with drawbacks. Notably, LAC which is a Learning With Errors (LWE) based IND-CCA secure KEM in the 2nd round of the NIST PQC that applies an error correcting code is susceptible to a key recovery attack recently proposed by Guo et al. [37]. Also, code-based schemes have a history of attacks [36, 59, 26] due to decoding errors. Recently, Bindel and Schanck [10] proposed a failure boosting attack for lattice-based schemes with a non-zero correctness error. For some code-based schemes, the analysis of the decoding error is a non-trivial task as it specifically depends on the decoder. For instance, the analysis of BIKE's decoder, another 2nd round NIST PQC candidate, has recently been updated [62].

Consequently, it would be interesting to have rigorous and simple approaches to remove decryption errors (to a certain degree) from PKE and KEM schemes.

Immunizing encryption schemes. The study of "immunizing" encryption schemes from decryption errors is not new. Goldreich, Goldwasser, and Halevi [32] studied the reduction or removal of decryption errors in the Ajtai-Dwork encryption scheme as well as Howgrave-Graham et al. [44] in context of NTRU. The first comprehensive and formal treatment has been given by Dwork, Naor, and Reingold [25] who study different amplification techniques in the standard and random oracle model to achieve non-malleable ($\mathsf{IND\text{-}CCA}$ secure) schemes. One very intuitive compiler is the direct product compiler $\mathsf{Enc}^{\otimes \ell}$ which encrypts a message M under a PKE $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with a certain decryption error δ under ℓ independent public-keys from KGen, i.e,. $\mathsf{pk}' := (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ as $\mathsf{Enc}'(\mathsf{pk}', M) := (\mathsf{Enc}(\mathsf{pk}_1, M), \ldots, \mathsf{Enc}(\mathsf{pk}_\ell, M))$. Dec', given $C' = (C_1, \ldots, C_\ell)$ tries to decrypt C_i, $1 \leq i \leq \ell$, and returns the result of a majority vote among all decrypted messages, yielding an encryption scheme with some error $\delta' \leq \delta$. Their asymptotic analysis, however, and limitation to PKEs with a binary message space does not make it immediate what this would mean in a concrete setting and in particular how to choose ℓ for practically interesting values of δ and δ'. For turning a so-obtained amplified scheme with negligible correctness error into a CCA-secure one in the ROM, they provide a transform using similar ideas, but more involved than the FO transform. Bitansky and Vaikuntanathan [11] go a step further and turn encryption schemes with a correctness error into perfectly correct ones, whereas they even consider getting completely rid of bad keys (if they exist) and, thus, completely immunize encryption schemes. They build upon the direct product compiler of Dwork et al. and then apply reverse randomization [53] and Nisan-Wigderson style derandomization [55]. Thereby, they partition the randomness space into good and bad randomness, and ensure that only good randomness is used for encryption and key generation.

Our goals. In this work, we are specifically interested in transformations that lift weaker schemes with non-negligible correctness error into CCA-secure ones with negligible error. Thereby, our focus is on modular ways of achieving this and can be seen as a concrete treatment of ideas that have also be discussed by Dwork et al. [25], who, however, treat their approaches in an asymptotic setting only. We show that the direct product compiler can be used with variants of the standard FO transform considered by HHK [40] (in the ROM) as well as Bindel et al. [9] and Jiang et al. [48] (in the quantum ROM (QROM) [15]). They are used by many candidates of the NIST PQC, when starting from PKE schemes having non-negligible correctness error generically. As we are particularly interested in *practical compilers* in a *concrete setting* to obtain CCA security for KEMs in the (Q)ROM, we analyze the concrete overhead of this compiler and its use with widely used variants of the transforms from HHK. Moreover, we provide a rigorous treatment of non-black-box applications of these ideas and show that they yield better concrete results than the direct application of the direct product compiler. Importantly, it gives a generic way to deal with the error from weaker schemes (e.g., $\mathsf{IND\text{-}CPA}$ secure ones with non-negligible error) which are easier to design. An interesting question that we will study is how does increasing from one to ℓ ciphertexts compare to increasing the parameters at comparable resulting decryption errors for existing round-two submissions in the NIST PQC. As it turns out, our approach performs well in context of code-based schemes but gives less advantage for lattice-based schemes.

We also study our approach beyond conventional PKEs and KEMs. In particular, a class of KEMs that have recently found interest especially in context of full forward-secrecy for zero round-trip time (0-RTT) key-exchange (KE) protocols are so-called *puncturable KEMs* [33, 34, 21, 63] and, in particular, Bloom Filter KEMs (BFKEMs) [21, 20]. BFKEMs schemes are CCA-secure KEMs that inherently have non-negligible correctness error. Interestingly, however, the non-negligible correctness error comes from the Bloom filter layer and the underlying IBE scheme (specifically, the Boneh-Franklin [16] instantiation in [21]) is required to provide perfect correctness. Thus, as all post-quantum IBEs have at least negligible correctness error, there are no known post-quantum BFKEMs.

1.1 Contribution

Our contributions on a more technical level can be summarized as follows:

Generic transform. We revisit the ideas of the direct product compiler of Dwork et al. [25] (dubbed $\mathsf{C}_{\mathsf{p,r}}$ and $\mathsf{C}_{\mathsf{p,d}}$ for randomized and deterministic PKEs respectively) in the context of the modular framework of HHK [40]. In particular, we present a generic transform dubbed $\mathsf{T}^\star$ that, given any randomized PKE scheme with non-negligible correctness error, produces a derandomized PKE scheme with negligible correctness error. We analyze the transform both in the ROM and QROM and give a tight reduction in the ROM and compare it to a generic application of the direct product compiler. The transform naturally fits into the modular framework of HHK [40], and, thus, by applying the $\mathsf{U}^{\not\perp}$ transform, gives rise to an $\mathsf{IND\text{-}CCA}$-secure KEM. For the analysis in the QROM, we follow the work of Bindel et al. [9]. We show that the $\mathsf{T}^\star$ transform also fits into their framework. Hence, given the additional injectivity assumption, we also obtain a tight proof for $\mathsf{U}^{\not\perp}$. But even if this assumption does not hold, the non-tight proofs of Jiang et al. [48] and Hövelmanns et al. [43] still apply. Compared to the analysis of the T transform that is used in the modular frameworks, our reductions lose a factor of ℓ, i.e., the number of parallel ciphertexts required to reach a negligible correctness error, in the ROM and a factor of ℓ^2 in the QROM. For concrete schemes this number is small (e.g., ≤ 5) and thus does not impose a significant loss. An overview of the transformations and how our transform

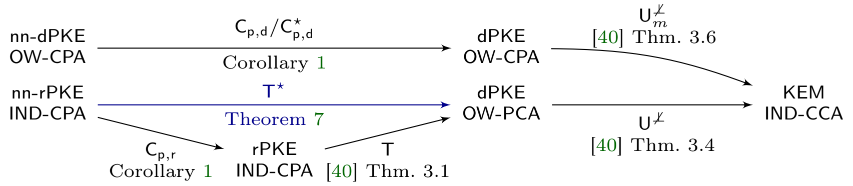

Fig. 1. Overview of the transformations in the ROM with the results related to $\mathsf{T}^\star$ highlighted in blue. rPKE denotes a randomized PKE. dPKE denotes a deterministic PKE. The prefix nn indicates encryption schemes with non-negligible correctness error.

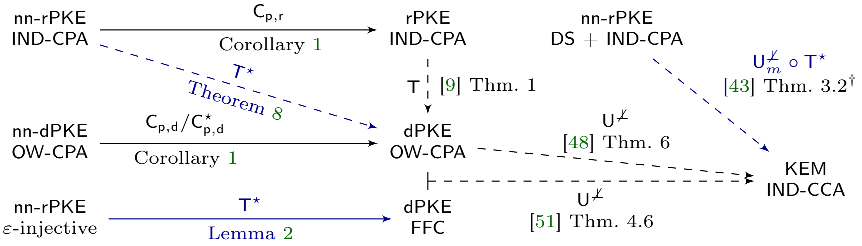

Fig. 2. Overview of the transformations in the QROM using the notation from Figure 1. A dashed arrow denotes a non-tight reduction. DS denotes disjoint simulatability.
†: Obtained by applying the modifications from Theorems 7 and 8 to [43, Thm 3.2].

fits into the modular frameworks is given in Figure 1 (ROM) and Figure 2 (QROM). Furthermore, using ideas similar to $\mathsf{T}^\star$, we discuss a modified version of the deterministic direct product compiler $\mathsf{C}_{\mathsf{p,d}}$ which we denote by $\mathsf{C}^\star_{\mathsf{p,d}}$, that compared to the original one allows to reduce the number of parallel repetitions needed to achieve negligible correctness error.

Evaluation. We evaluate $\mathsf{T}^\star$ based on its application to code- and lattice-based second-round candidates in the NIST PQC. In particular, we focus on schemes that offer IND-CPA secure versions with non-negligible correctness error such as ROLLO [4], BIKE [3] and Round5 [30]. We compare their IND-CCA variants with our transform applied to the IND-CPA schemes. In particular, for the code-based schemes such as ROLLO we can observe improvements in the combined size of public keys and ciphertexts, a metric important when used in protocols such as TLS, as well as its runtime efficiency. We also argue the ease of implementing our so-obtained schemes which can rely on simpler decoders. For lattice-based constructions, we find that the use of the transform results in an increase in the sum of ciphertext and public-key size of 30% even in the best case scenario, i.e., for an IND-CPA version of KEM Round5 [30]. Nevertheless, it offers easier constant-time implementations and the opportunity of decreasing the correctness error without changing the underlying parameter set and, thus, the possibility to focus on analyzing and implementing one parameter set for both, IND-CPA and IND-CCA security.

Bloom Filter KEMs. Finally, we revisit puncturable KEMs from Bloom filter KEMs (BFKEMs) [21, 20], a recent primitive to realize 0-RTT key exchange protocols with full forward-secrecy [34]. Currently, it is unclear how to instantiate BFKEMs generically from IBE and, in particular, from conjectured post-quantum assumptions due to the correctness error of the respective IBE schemes. We show that one can construct BFKEMs generically from any IBE and even base it upon IBEs with a (non-)negligible correctness error. Consequently, our results allow BFKEMs to be instantiated from lattice- and code-based IBEs and, thereby, we obtain the first post-quantum CCA-secure BFKEM.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$, and let $\lambda \in \mathbb{N}$ be the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow_{\$} \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For an algorithm A, let $y \leftarrow A(\lambda, x)$ be the process of running A on input (λ, x) with access to uniformly random coins and assigning the result to y (we may assume that all algorithms take λ as input). To make the random coins r explicit, we write $A(x; r)$. We say an algorithm A is probabilistic polynomial time (PPT) if the running time of A is polynomial in λ. A function f is negligible if its absolute value is smaller than the inverse of any polynomial, i.e., if $\forall c\ \exists k_0$ s.t. $\forall \lambda \geq k_0 : |f(\lambda)| < 1/\lambda^c$.

2.1 Public-Key Encryption and Key-Encapsulation Mechanisms

Public-key encryption. A public-key encryption (PKE) scheme Π with message space $\mathcal{M}$ consists of the three PPT algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{KGen}(\lambda)$, on input security parameter λ, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$. $\mathsf{Enc}(\mathsf{pk}, M)$, on input pk and message $M \in \mathcal{M}$, outputs a ciphertext C. $\mathsf{Dec}(\mathsf{sk}, C)$, on input sk and C, outputs $M \in \mathcal{M} \cup \{\perp\}$. We may assume that pk is implicitly available in Dec.

Correctness. We recall the definition of δ-correctness of [40]. A PKE Π is δ-correct if

$$E\left[\max_{M \in \mathcal{M}} \Pr\left[c \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, C) \neq M\right]\right] \leq \delta,$$

where the expected value is taken over all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$.

PKE-IND-CPA, PKE-OW-CPA, and PKE-OW-PCA security. We say a PKE Π is PKE-IND-CPA-secure if and only if any PPT adversary A has only negligible advantage in the following security experiment. First, A gets an honestly generated public key pk. A outputs equal-length messages (M_0, M_1) and, in return, gets $C_b^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M_b)$, for $b \leftarrow_{\$} \{0, 1\}$. Eventually, A outputs a guess b'. If $b = b'$, then the experiment outputs 1. For PKE-OW-CPA security, A does not receive a ciphertext for A-chosen messages, but only a ciphertext $C^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$ for $M \leftarrow_{\$} \mathcal{M}$ and outputs M'; if $M = M'$, then the experiment

```
Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda)$
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $(M_0, M_1) \leftarrow A(\mathsf{pk})$
  $b \leftarrow_{\$} \{0,1\}$
  $C^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M_b)$
  $b' \leftarrow A(C^*)$
  if $b = b'$ then return 1
  else return 0
```

```
Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,A}(\lambda)$
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $M \leftarrow_{\$} \mathcal{M}$
  $C^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$
  $M' \leftarrow A(\mathsf{pk}, C^*)$
  if $M = M'$ then return
  1 else return 0
```

```
Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi,A}(\lambda)$
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $M \leftarrow_{\$} \mathcal{M}$
  $C^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$
  $M' \leftarrow A^{\textsc{Pco}(\cdot,\cdot)}(\mathsf{pk}, C^*)$
  if $M = M'$ then return 1
  else return 0
```

Fig. 3. PKE-x-y security with $\mathsf{x} \in \{\mathsf{OW}, \mathsf{IND}\}$, $\mathsf{y} \in \{\mathsf{CPA}, \mathsf{PCA}\}$ for Π.

outputs 1. For PKE-OW-PCA security, A additionally has access to a plaintext checking oracle $\textsc{Pco}(M, C)$ returning 1 if $M = \mathsf{Dec}(\mathsf{sk}, C)$ and 0 otherwise.

Definition 1. *For any PPT adversary A the advantage function*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda) = 1\right] - \frac{1}{2}\right|,$$

is negligible in λ, where the experiment $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda)$ is given in Figure 3 and Π is a PKE as above.

Definition 2. *For any PPT adversary A, and $\mathsf{y} \in \{\mathsf{CPA}, \mathsf{PCA}\}$ the advantage function*

$$\mathsf{Exp}^{\mathsf{pke\text{-}OW\text{-}y}}_{\Pi,A}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{pke\text{-}OW\text{-}y}}_{\Pi,A}(\lambda) = 1\right],$$

is negligible in λ, where the experiments $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,A}(\lambda)$ and $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi,A}(\lambda)$ are given in Figure 3 and Π is a PKE as above.

We recall a well known lemma below:

Lemma 1. *For any adversary B there exists an adversary A with the same running time as that of B such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,B}(\lambda) \leq \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda) + \frac{1}{|\mathcal{M}|}.$$

We note that Lemma 1 equivalently holds for the ℓ-IND-CPA notion below.

Multi-challenge setting. We recall some basic observations from [8] regarding the multi-challenge security of PKE schemes. In particular, for our construction we need the relation between OW-CPA/IND-CPA security in the conventional single-challenge and single-user setting and n-OW-CPA/n-IND-CPA respectively, which represents the multi-challenge and multi-user setting. In particular, latter means that the adversary is allowed to obtain multiple challenges under multiple different public keys.

Theorem 1 (Th. 4.1 [8]). *Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme that provides x-CPA security with $\mathsf{x} \in \{\mathsf{OW}, \mathsf{IND}\}$. Then, it holds that:*

$$\mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi,A}(\lambda) \geq \frac{1}{q \cdot n} \cdot \mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}x\text{-}cpa}}_{\Pi,A}(\lambda),$$

```
Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,A}(\lambda)$
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $L \leftarrow A(\mathsf{pk})$
  if exists $C \in L$ with $M \in \mathcal{M}$ such that $\mathsf{Enc}(\mathsf{pk},M) = C$ and $\mathsf{Dec}(\mathsf{sk},C) \neq M$
  then return $1$ else return $0$
```

Fig. 4. Finding-failing-ciphertext experiment for Π.

where n is the number of public keys and A makes at most q queries to any of its n challenge oracles.

Although the loss imposed by the reduction in Theorem 1 can be significant when used in a general multi-challenge and multi-user setting, in our application we only have cases where $n = 1$ and small q ($q = 5$ at most), or vice versa (i.e., $q = 1$ and $n = 5$ at most) thus tightness in a concrete setting is preserved.

Finding failing ciphertexts and injectivity. For the QROM security proof we will need the following two definitions from [9].

Definition 3 (ε-injectivity). *A PKE Π is called ε-injective if*

- *Π is deterministic and*

$$\Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda) : M \mapsto \mathsf{Enc}(\mathsf{pk},M) \text{ is not injective}] \leq \varepsilon.$$

- *Π is non-deterministic with randomness space $\mathcal{R}$ and*

$$\Pr\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda), \\ M, M' \leftarrow_{\$} \mathcal{M}, r, r' \leftarrow_{\$} \mathcal{R}\end{matrix} : \mathsf{Enc}(\mathsf{pk},M;r) = \mathsf{Enc}(\mathsf{pk},M';r')\right] \leq \varepsilon.$$

Definition 4 (Finding failing ciphertexts). *For a deterministic* PKE*, the* FFC*-advantage of an adversary A is defined as*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\Pi,A}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,A}(\lambda) = 1\right],$$

where the experiment $\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,A}$ is given in Figure 4.

Key-encapsulation mechanism. A key-encapsulation mechanism (KEM) scheme KEM with key space $\mathcal{K}$ consists of the three PPT algorithms $(\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$: $\mathsf{KGen}(\lambda)$, on input security parameter λ, outputs public and secret keys $(\mathsf{pk},\mathsf{sk})$. $\mathsf{Encaps}(\mathsf{pk})$, on input pk, outputs a ciphertext C and key k. $\mathsf{Decaps}(\mathsf{sk}, C)$, on input sk and C, outputs k or $\{\perp\}$.

Correctness of KEM. We call a KEM δ-correct if for all $\lambda \in \mathbb{N}$, for all $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$, for all $(C,\mathsf{k}) \leftarrow \mathsf{Enc}(\mathsf{pk})$, we have that

$$\Pr[\mathsf{Dec}(\mathsf{sk},C) \neq \mathsf{k}] \leq \delta.$$

KEM-IND-CCA security. We say a KEM KEM is KEM-IND-CCA-secure if and only if any PPT adversary A has only negligible advantage in the following

```
**Exp.** $\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},A}(\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $(C^*, \mathsf{k}_0) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \mathsf{k}_1 \leftarrow_\$ \mathcal{K}$
  $b \leftarrow_\$ \{0, 1\}$
  $b' \leftarrow A^{\mathsf{Decaps}(\mathsf{sk},\cdot)}(\mathsf{pk}, C^*, \mathsf{k}_b)$
  **if** $b = b'$ **then return** $1$ **else return** $0$
```

Fig. 5. KEM-IND-CCA security experiment for KEM.

security experiment. First, A gets an honestly generated public key pk as well as a ciphertext-key pair (C^*, k_b), for $(C^*, \mathsf{k}_0) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, for $\mathsf{k}_1 \leftarrow_\$ \mathcal{K}$, and for $b \leftarrow_\$ \{0, 1\}$. A has access to a decapsulation oracle $\mathsf{Dec}(\mathsf{sk}, \cdot)$ and we require that A never queries $\mathsf{Decaps}(\mathsf{sk}, C^*)$. Eventually, A outputs a guess b'. Finally, if $b = b'$, then the experiment outputs 1.

Definition 5. *For any PPT adversary A, the advantage functions*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},A}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},A}(\lambda) = 1 \right] - \frac{1}{2} \right|,$$

is negligible in λ, where the experiment $\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},A}(\lambda)$ is given in Figure 5 and KEM *is a KEM as above.*

2.2 Identity-Based Encryption

An identity-based encryption (IBE) scheme IBE with identity space $\mathcal{ID}$ and message space $\mathcal{M}$ consists of the five PPT algorithms $(\mathsf{KGen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{KGen}(\lambda)$ on input security parameter λ, outputs master public and secret keys (mpk, msk). $\mathsf{Ext}(msk, id)$ on input identity $id \in \mathcal{ID}$, outputs a user secret key usk_{id}. $\mathsf{Enc}(mpk, id, M)$ on input mpk, $id \in \mathcal{ID}$, and message $M \in \mathcal{M}$, outputs a ciphertext C. $\mathsf{Dec}(usk_{id}, C)$ on input usk_{id} and C, outputs $M \in \mathcal{M} \cup \{\bot\}$.

Correctness of IBE. Analogous to [40] we define δ-correctness of an IBE IBE for any $id \in \mathcal{ID}$ as

$$E\left[\max_{M \in \mathcal{M}} \Pr\left[C \leftarrow \mathsf{Enc}(mpk, id, M) : \mathsf{Dec}(usk_{id}, C) \neq M \right] \right] \leq \delta(\lambda),$$

where the expected value is taken over all $(mpk, msk) \leftarrow \mathsf{KGen}(\lambda)$ and $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$.

We recall the formal definitions of IBE-sIND-CPA security in the full version.

3 CCA Security from Non-Negligible Correctness Errors

In this section, we present our approaches to generically achieve CCA secure KEMs in the (Q)ROM with negligible correctness error when starting from an $\mathsf{OW\text{-}CPA}$ or $\mathsf{IND\text{-}CPA}$ secure PKE with non-negligible correctness error. We start by discussing the definitions of correctness errors of PKE and KEMs. Then, we present a generic transform based on the direct product compiler of Dwork et al. [25] and revisit certain FO transformation variants from [40] (in particular the T and U transformations), their considerations in the QROM [9] and their application with the direct product compiler. As a better alternative, we analyze the non-black-box use of the previous technique yielding transformation $\mathsf{T}^{\star}$, that combines the direct product compiler with the T transformation. Finally, we provide a comprehensive comparison of the two approaches.

3.1 On the Correctness Error

In this work, we use the δ-correctness for PKEs given by HHK in [40]. With this definition, particularly bad keys in terms of correctness error only contribute a fraction to the overall correctness error as it averages the error probability over all key pairs: if there are negligible many keys with a higher correctness error, then those keys do not really contribute to the overall correctness error. At the same time this definition is tailored, via maxing over all possible messages, to the security proofs of the FO-transforms where an adversary could actively search for the worst possible message, in order to trigger decryption failure. As also done by Dwork et al. [25], we explicitly write the correctness error as a function in the security parameter:

Definition 6. *A* PKE Π *is* $\delta(\cdot)$*-correct if*

$$E\left[\max_{M \in \mathcal{M}} \Pr\left[C \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, C) \neq M\right]\right] \leq \delta(\lambda),$$

where the expected value is taken over all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$.

It will be important for our transform to make explicit that the correctness error depends on the security level, as this allows us to chose a function $\ell(\cdot)$ such that $\delta(\lambda)^{\ell(\lambda)} \leq 2^{-\lambda}$. We will often just write $\delta = \delta(\lambda)$ and $\ell = \ell(\lambda)$ for simplicity.

An alternative but equivalent definition, as used in [40], can be given in the following form: a PKE Π is called $\delta(\cdot)$-correct if we have for all (possibly unbounded) adversaries A that

$$\mathsf{Adv}^{\mathsf{cor}}_{\Pi,A}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{cor}}_{\Pi,A}(\lambda) = 1\right] \leq \delta(\lambda),$$

where the experiment is given in Figure 6. If Π is defined relative to a random oracle H, then the adversary is given access to the random oracle and δ is additionally a function in the number of queries q_{H}, i.e., the bound is given by $\leq \delta(\lambda, q_{\mathsf{H}})$. We note that in [10] an alternative definition of correctness was proposed, where the adversary does not get access to sk and the adversary's runtime is bounded. With this change, it can be run as part of the $\mathsf{IND\text{-}CCA}$ experiment which does not change the power of the $\mathsf{IND\text{-}CCA}$ adversary and additionaly removes a factor q_{H} from the correctness error and advantage analysis. In particular, one can obtain an upper bound for $\mathsf{IND\text{-}CCA}$ security of a scheme via the correctness error.

```
Exp. $\mathsf{Exp}^{\mathsf{cor}}_{\Pi,A}(\lambda)$
  $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
  $M \leftarrow A(\mathsf{pk},\mathsf{sk})$
  if $M \neq \mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},M))$ then return 1 else return 0
```

Fig. 6. Correctness experiment for PKE.

We recall, for completeness, the definition of correctness error, here denoted as DNR-δ-correctness (from Dwork-Naor-Reingold), used by Dwork et al.:

Definition 7 (Def. 2, Def. 3 [25]). *A* PKE Π *is*

- *DNR-$\delta(\cdot)$-correct if we have that*

$$\Pr[\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},M)) \neq M] \leq \delta(\lambda),$$

where the probability is taken over the choice of key pairs $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$*,* $M \in \mathcal{M}$ *and over the random coins of* Enc *and* Dec*.*

- *DNR-(almost-)all-keys $\delta(\cdot)$-correct if for all (but negligible many) keys* $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$*, we have that*

$$\Pr[\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},M)) \neq M] \leq \delta(\lambda),$$

where the probability is taken over the choice of $M \in \mathcal{M}$ *and over the random coins of* Enc *and* Dec*.*

Correctness error in this sense still allows bad key pairs that potentially have an even worse error but it is not suited for our security proofs as the probability is also taken over $M \leftarrow_{\$} \mathcal{M}$. Recently Drucker et al. [23] introduced the notion of message agnostic PKE and showed that all the versions of BIKE, a 2nd round candidate in the NIST PQC, are message-agnostic: in such a PKE, the probability that, given $(\mathsf{sk},\mathsf{pk})$, the encryption of a message $M \in \mathcal{M}$ correctly decrypts is independent of the message $M \in \mathcal{M}$ itself. For such PKEs the definitions of δ-correctness and DNR-δ-correctness coincide (Cor. 1 [23]).

3.2 Compiler for Immunizing Decryption Errors

Now we present two variants of a compiler $\mathsf{C_p}$ denoted $\mathsf{C_{p,d}}$ (for deterministic schemes) and $\mathsf{C_{p,r}}$ (for randomized schemes) which is based on the direct product compiler by Dwork et al. [25]. We recall that the idea is to take a PKE scheme $\Pi = (\mathsf{KGen},\mathsf{Enc},\mathsf{Dec})$ with non-negligible correctness error δ (and randomness space $\mathcal{R}$ in case of randomized schemes) and output a PKE scheme $\Pi' = (\mathsf{KGen}',\mathsf{Enc}',\mathsf{Dec}')$ with negligible correctness error δ' (and randomness space $\mathcal{R}' := \mathcal{R}^{\ell}$, for some $\ell \in \mathbb{N}$, in case of a randomized schemes). We present a precise description of the compilers in Figure 7. Note that in Dec', the message that is returned most often by Dec is returned. If two or more messages are tied, one of them is returned arbitrarily and we denote this operation as $\mathsf{maj}(M')$.

```
$\mathit{\Pi}'.\mathsf{KGen}'(\lambda, \ell)$
  // if $\mathsf{C}_{\mathsf{p,r}}$
  return $\mathit{\Pi}.\mathsf{KGen}(\lambda)$
  // if $\mathsf{C}_{\mathsf{p,d}}$
  for $i \in [\ell]$
    $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathit{\Pi}.\mathsf{KGen}(\lambda)$
  $\mathsf{pk} := (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$
  $\mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$
  return $(\mathsf{pk}, \mathsf{sk})$
```

```
$\mathit{\Pi}'.\mathsf{Enc}'(\mathsf{pk}, M)$
  for $i \in [\ell]$
    // if $\mathsf{C}_{\mathsf{p,r}}$
    $r_i \leftarrow_\$ \mathit{\Pi}.\mathcal{R}$
    $C_i \leftarrow \mathit{\Pi}.\mathsf{Enc}(\mathsf{pk}, M; r_i)$
    // if $\mathsf{C}_{\mathsf{p,d}}$
    $C_i \leftarrow \mathit{\Pi}.\mathsf{Enc}(\mathsf{pk}_i, M)$
  $C := (C_1. \ldots, C_\ell)$
  return $C$
```

```
$\mathit{\Pi}'.\mathsf{Dec}'(\mathsf{sk}, C)$
  $C := (C_1. \ldots, C_\ell)$
  for $i \in [\ell]$
    // if $\mathsf{C}_{\mathsf{p,r}}$
    $M_i' := \mathit{\Pi}.\mathsf{Dec}(\mathsf{sk}, C_i)$
    // if $\mathsf{C}_{\mathsf{p,d}}$
    $M_i' := \mathit{\Pi}.\mathsf{Dec}(\mathsf{sk}_i, C_i)$
  return $\mathsf{maj}(M_1', \ldots, M_\ell')$
```

Fig. 7. Compilers $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$.

Analyzing correctness. Dwork et al. in [25] explicitly discuss the amplification of the correctness for encryption schemes with a binary message space $\mathcal{M} = \{0, 1\}$ and obtain that to achieve DNR-δ'-correctness $\ell > {}^{c}/_{(1-\delta)^2} \cdot \log {}^{1}/_{\delta'}$ when starting from a scheme with DNR-δ-correctness. As c is some constant that is never made explicit, the formula is more of theoretical interest and for concrete instances it is hard to estimate the number of required ciphertexts. We can however analyze the probabilities that the majority vote in Dec' returns the correct result. As far as the correctness notion used in this work is concerned, in order to prove an acceptable good lower bound for the δ-correctness of the direct product compiler, it suffices to find an event, in which the decryption procedure fails, that happens with a large enough probability. The following reasoning applies to both its deterministic and randomized versions, $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$ respectively. One such case is the following: only 1 ciphertext correctly decrypts and all other $\ell - 1$ ciphertexts decrypt to $\ell - 1$ distinct wrong messages. During the maj operation, one of the "wrong" messages is then returned. The probability of this event is

$$\frac{\ell-1}{\ell}\binom{\ell}{\ell-1}\delta^{\ell-1}(1-\delta)\frac{M-1}{M-1}\frac{M-2}{M-1}\cdots\frac{M-(\ell-1)}{M-1}.$$

Looking ahead to our compiler T^* presented in Section 3.4, if the message space is sufficiently large, this probability is bigger than $\delta^{\ell-1}(1-\delta)$, which gives that at least one more ciphertext is needed to achieve the same decryption error as with our compiler T^*. The results are shown in Table 1. One can compute the exact probability of decryption error by listing all cases in which the decryption fails and summing up all these probabilities to obtain the overall decryption failure of the direct product compiler. This computation is not going to give a significantly different result from the lower bound that we have just computed.

We note that using 2 parallel ciphertexts does not improve the correctness error, so the direct product compiler only becomes interesting for $\ell \geq 3$: indeed for $\ell = 2$, we have 3 possible outcomes in which the decryption algorithm can fail: 1) the first ciphertext decrypts and the second does not, 2) vice versa, 3) both fail to decrypt. In 1), 2), half the time the wrong plaintext is returned. Summing these probabilities gives exactly δ.

Table 1. Estimation of the correctness error for the direct product compilers. $\delta'(\ell)$ denotes the correctness error for ℓ ciphertexts.

δ	$\delta'(2)$	$\delta'(3)$	$\delta'(4)$
2^{-32}	$\approx 2^{-32}$	$\approx 2^{-63}$	$\approx 2^{-94}$
2^{-64}	$\approx 2^{-64}$	$\approx 2^{-127}$	$\approx 2^{-190}$
2^{-96}	$\approx 2^{-96}$	$\approx 2^{-191}$	$\approx 2^{-284}$

Remark 1. As far as the deterministic direct product compiler $\mathsf{C}_{\mathsf{p,d}}$ is concerned, the correctness error can be improved by modifying the decryption: instead of relying on the maj operation, we can re-encrypt the plaintexts obtained during decryption with the respective keys and compare them to the original ciphertexts. Only if this check passes, the plaintext is returned. If this is done, then decryption fails iff no ciphertext decrypts correctly, i.e., with probability δ^ℓ, and thereby the number of parallel repetition necessary to achieve negligible correctness-error is reduced at the cost of a computational overhead in the decryption. We denote this version of the deterministic direct product compiler by $\mathsf{C}^\star_{\mathsf{p,d}}$.

Their security follows by applying Theorem 1 with $q = 1$ and $n = \ell$ in the deterministic case, for both $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}^\star_{\mathsf{p,d}}$, or vice versa with $q = \ell$ and $n = 1$ in the randomized case:

Corollary 1. *For any* x-CPA *adversary B against Π' obtained via applying* $\mathsf{C}_{\mathsf{p,y}}$ *to Π, there exists an* x-CPA *adversary A such that:*

$$\mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi',B}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi,A}(\lambda),$$

where $\mathsf{y} = \mathsf{d}$ *if* $\mathsf{x} = \mathsf{OW}$ *and* $\mathsf{y} = \mathsf{r}$ *if* $\mathsf{x} = \mathsf{IND}$.

As the analysis above suggests, ℓ will be a small constant, so the loss in ℓ does not pose a problem regarding tightness.

3.3 Transformations T and $\mathsf{U}^{\not\perp}$

Subsequently, we discuss basic transformations from [41] to first transform an IND-CPA secure PKE into an OW-PCA secure PKE (transformation T in [41]) and then to convert an OW-PCA secure PKE into an IND-CCA secure KEM with implicit rejection (transformation $\mathsf{U}^{\not\perp}$ in [41]) and we discuss alternative transformations later. We stress that these transformations either work for perfectly correct schemes or schemes with a negligible correctness error.

T: IND-CPA $\Longrightarrow$ OW-PCA **(ROM)**/OW-CPA **(QROM).** The transform T is a simple de-randomization of a PKE by deriving the randomness r used by the algorithm Enc via evaluating a random oracle (RO) on the message to be encrypted. More precisely, let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE with message space $\mathcal{M}$ and randomness space $\mathcal{R}$ and $\mathsf{G}\colon \mathcal{M} \to \mathcal{R}$ be a RO. We denote the PKE

```
$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$
    $C := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M))$
    return $C$

$\Pi'.\mathsf{Dec}(\mathsf{sk}, C)$
    $M' := \Pi.\mathsf{Dec}(\mathsf{sk}, C)$
    if $M' = \bot$ or $C \neq \Pi.\mathsf{Enc}(\mathsf{pk}, M'; \mathsf{G}(M'))$
        return $\bot$
    else return $M'$
```

Fig. 8. OW-PCA-secure scheme $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$ with deterministic encryption.

```
$\mathsf{KEM.KGen}(\lambda)$
    $(\mathsf{pk}', \mathsf{sk}') \leftarrow \Pi'.\mathsf{KGen}(\lambda)$
    $s \leftarrow_\$ \mathcal{M}$
    $\mathsf{sk} := (\mathsf{sk}', s)$
    return $(\mathsf{pk}', \mathsf{sk})$

$\mathsf{KEM.Encaps}(\mathsf{pk})$
    $M \leftarrow_\$ \mathcal{M}$
    $C \leftarrow \Pi'.\mathsf{Enc}(\mathsf{pk}, M)$
    $K := \mathsf{H}(M, C)$
    return $(K, C)$

$\mathsf{KEM.Decaps}\ (\mathsf{sk}, C)$
    Parse $\mathsf{sk} = (\mathsf{sk}', s)$
    $M' := \Pi'.\mathsf{Dec}(\mathsf{sk}', C)$
    if $M' \neq \bot$
        return $K := \mathsf{H}(M', C)$
    else return $K := \mathsf{H}(s, C)$
```

Fig. 9. IND-CCA-secure KEM scheme $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\Pi', \mathsf{H}]$.

Π' obtained by applying transformation T depicted in Figure 8 as $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$, where $\Pi'.\mathsf{KGen} = \Pi.\mathsf{KGen}$ and is thus omitted.

For the ROM, we recall the following theorem:

Theorem 2 (Thm. 3.2 [41] (Π IND-CPA $\Longrightarrow$ Π' OW-PCA)). *Assume Π to be δ-correct. Then, Π' is $\delta_1(q_\mathsf{G}) = q_\mathsf{G} \cdot \delta$ correct and for any* OW-PCA *adversary B that issues at most q_G queries to the RO* G *and q_P queries to a plaintext checking oracle* Pco*, there exists an* IND-CPA *adversary A running in about the same time as B such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',B}(\lambda) \leq q_\mathsf{G} \cdot \delta + \frac{2q_\mathsf{G} + 1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda).$$

And for the QROM, we recall the following theorem:

Theorem 3 (Thm. 1 [9] (Π IND-CPA $\Longrightarrow$ Π' OW-CPA)). *If A is an* OW-CPA*-adversary against $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$ issuing at most q_G queries to the quantum-accessible RO* G *of at most depth d, then there exists an* IND-CPA *adversary B against Π running in about the same time as A such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',A}(\lambda) \leq (d+1)\left(\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) + \frac{8(q_\mathsf{G}+1)}{|\mathcal{M}|}\right).$$

<u>$\mathsf{U}^{\not\perp}$: OW-PCA $\Longrightarrow$ IND-CCA.</u> The transformation $\mathsf{U}^{\not\perp}$ transforms any OW-PCA secure PKE Π' into an IND-CCA secure KEM in the (Q)ROM. The basic idea is that one encrypts a random message M from the message space $\mathcal{M}$ of Π' and the encapsulated key is the RO evaluated on the message M and the corresponding ciphertext C under Π'. This transformation uses implicit rejection and on decryption failure does not return $\bot$, but an evaluation of the RO on the ciphertext and a random message $s \in \mathcal{M}$, being part of sk of the resulting KEM, as a "wrong" encapsulation key. It is depicted in Figure 9.

In the ROM, we have the following result:

Theorem 4 (Thm. 3.4 [41] ($\mathit{\Pi}'$ $\mathsf{OW\text{-}PCA} \implies \mathsf{KEM}$ $\mathsf{IND\text{-}CCA}$)). *If $\mathit{\Pi}'$ is δ_1-correct, then KEM is δ_1-correct in the random oracle model. For any $\mathsf{IND\text{-}CCA}$ adversary B against KEM, issuing at most q_{H} queries to the random oracle H, there exists an $\mathsf{OW\text{-}PCA}$ adversary A against $\mathit{\Pi}'$ running in about the same time as B that makes at most q_{H} queries to the* Pco *oracle such that*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},B}(\lambda) \leq \frac{q_{\mathsf{H}}}{|\mathcal{M}|} + \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\mathit{\Pi}',A}(\lambda).$$

For the QROM, we have the following non-tight result:

Theorem 5 (Thm. 6 [48] ($\mathit{\Pi}'$ $\mathsf{OW\text{-}PCA} \implies \mathsf{KEM}$ $\mathsf{IND\text{-}CCA}$)). *Let $\mathit{\Pi}'$ be a deterministic PKE scheme which is independent of H. Let B be an $\mathsf{IND\text{-}CCA}$ adversary against the KEM $\mathsf{U}^{\not\perp}[\mathit{\Pi}', \mathsf{H}]$, and suppose that A makes at most q_d (classical) decryption queries and q_{H} queries to quantum-accessible random oracle H of depth at most d, then there exists and adversary B against $\mathit{\Pi}'$ such that*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{U}^{\not\perp}[\mathit{\Pi}',\mathsf{H}],A}(\lambda) \leq \frac{2 \cdot q_{\mathsf{H}}}{\sqrt{|\mathcal{M}|}} + 2 \cdot \sqrt{(q_{\mathsf{H}}+1)(2 \cdot \delta + \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\mathit{\Pi}',B}(\lambda))}.$$

If we assume ε-injectivity and FFC, respectively, we have tighter bounds:

Theorem 6 (Thm. 4.6 [51] ($\mathit{\Pi}'$ $\mathsf{OW\text{-}CPA} + \mathsf{FFC} \implies \mathsf{KEM}$ $\mathsf{IND\text{-}CCA}$)). *Let $\mathit{\Pi}'$ be an ε-injective deterministic PKE scheme which is independent of H. Suppose that A is an $\mathsf{IND\text{-}CCA}$ adversary against the KEM $\mathsf{U}^{\not\perp}[\mathit{\Pi}', \mathsf{H}]$, and suppose that A makes at most q_d (classical) decryption queries and q_{H} queries to quantum-accessible random oracle H of depth at most d, then there exist two adversaries running in about the same time as A:*

- *an $\mathsf{OW\text{-}CPA}$-adversary B_1 against $\mathit{\Pi}'$ and*
- *a FFC-adversary B_2 against $\mathit{\Pi}'$ returning a list of at most q_d ciphertexts,*

such that

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{U}^{\not\perp}[\mathit{\Pi}',\mathsf{H}],A}(\lambda) \leq 4d \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\mathit{\Pi}',B_1}(\lambda) + 6\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\mathit{\Pi}',B_2}(\lambda) + (4d+6) \cdot \varepsilon.$$

$\mathsf{FO}^{\not\perp}[\mathit{\Pi}, \mathsf{G}, \mathsf{H}]$**.** By combining transformation T with $\mathsf{U}^{\not\perp}$ one consequently obtains an $\mathsf{IND\text{-}CCA}$ secure KEM KEM from an $\mathsf{IND\text{-}CPA}$ secure PKE $\mathit{\Pi}$. Note that the security reduction of the $\mathsf{FO}^{\not\perp} := \mathsf{U}^{\not\perp} \circ \mathsf{T}$ variant of the FO is tight in the random oracle model and works even if $\mathit{\Pi}$ has negligible correctness error instead of perfect correctness.

$\mathsf{FO}^{\not\perp}[\mathit{\Pi}, \mathsf{G}, \mathsf{H}]$ **in the QROM.** Hofheinz et al. in [41] also provide variants of the FO transform that are secure in the QROM, but they are (highly) non-tight. Bindel et al. [9] presented a tighter proof for $\mathsf{U}^{\not\perp}$ under an additional assumption of ε-injectivity. This result was recently improved by Kuchta et al. [51]. Additionally, Jiang et al. [48] provided tighter proofs for the general case.

$\mathsf{U}^{\perp}$, $\mathsf{U}^{\perp}_{m}$, $\mathsf{U}^{\not\perp}_{m}$ and other approaches. Besides the transform with implicit rejection, $\mathsf{U}^{\not\perp}$, one can also consider explicit rejection, $\mathsf{U}^{\perp}$ and versions of both where the derived session key depends on the ciphertext, $\mathsf{U}^{\not\perp}_{m}$ and $\mathsf{U}^{\perp}_{m}$, respectively. Bindel et al. [9] show that security of implicit rejection implies security with explicit rejection. The opposite direction also holds if the scheme with explicit rejection also employs key confirmation. Moreover, they show that the security is independent of including the ciphertext in the session key derivation.

A different approach was proposed by Saito et al. [58], where they start from a deterministic disjoint simulatable PKE and apply $\mathsf{U}^{\not\perp}_{m}$ with an additional re-encryption step in the decryption algorithm. While the original construction relied on a perfectly correct PKE, Jiang et al. gave non-tight reductions for schemes with negligible correctness error in [47]. Hövelmanns et al. [43] improve over this approach by giving a different modularization of Saito et al.'s TPunc.

Black-box use of the compiler $\mathsf{C}_{\mathsf{p,d}}/\mathsf{C}^{\star}_{\mathsf{p,d}}/\mathsf{C}_{\mathsf{p,r}}$. Using $\mathsf{C}_{\mathsf{p,d}}$, $\mathsf{C}^{\star}_{\mathsf{p,d}}$ or $\mathsf{C}_{\mathsf{p,r}}$ from Section 3.2, we can transform any deterministic or randomized PKE with non-negligible correctness error into one with negligible correctness error. Consequently, Theorem 1 as a result yields a scheme that is compatible with all the results on the T and variants of the U transformations in this section. Note that in particular this gives us a general way to apply these variants of the FO transform to PKE schemes with non-negligible correctness error.

3.4 Non Black-Box Use: the Transformation $\mathsf{T}^{\star}$

Since the direct product compiler is rather complicated to analyze, we alternatively investigate to start from an $\mathsf{IND\text{-}CPA}$ secure PKE Π with non-negligible correctness error δ and introduce a variant of the transform T to de-randomize a PKE, denoted $\mathsf{T}^{\star}$. The idea is that we compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a RO (see Figure 10 for a compact description). The resulting de-randomized PKE Π' has then correctness error $\delta' := \delta^{\ell}$, where ℓ is chosen in a way that δ^{ℓ} is negligible. To the resulting PKE Π' we can then directly apply the transformation $\mathsf{U}^{\not\perp}$ to obtain an $\mathsf{IND\text{-}CCA}$ secure KEM KEM with negligible correctness error in the (Q)ROM.

Note that as we directly integrate the product compiler into the T transform, the correctness of the message can be checked via the de-randomization. Hence, we can get rid of the majority vote in the direct product compiler. With this change the analysis of the concrete choice of ℓ becomes simpler and, more importantly, allows us to choose smaller ℓ than in the black-box use of the compiler.

Remark 2. Note that in Figure 10 we explicitly consider the case where Dec of the PKE scheme Π may return something arbitrary on failed decryption. For the simpler case where we have a PKE scheme Π which always returns $\perp$ on failed decryption, we can easily adapt the approach in Figure 10. Namely, we would decrypt all ℓ ciphertexts C_i, $i \in [\ell]$. Let $h \in [\ell]$ be the minimum index such that $\mathtt{res}[h] \neq \perp$. Then for every element $j \in [\ell]$ run

```
$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$
  for $i = 1, \ldots, \ell$ do
    $C_i := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$
  $C := (C_1, \ldots, C_\ell)$
  return $C$
```

```
$\Pi'.\mathsf{Dec}(\mathsf{sk}, C)$
  $\mathtt{res} \leftarrow \perp$, $\mathtt{check} \leftarrow \perp$
  for $i = 1, \ldots, \ell$ do
    $\mathtt{res}[i] := \Pi.\mathsf{Dec}(\mathsf{sk}, C_i)$
  for $i \in [\ell]$ s.t. $\mathtt{res}[i] \neq \perp$ do
    if $\forall j \in [\ell] : C_j = \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[i], \mathsf{G}(\mathtt{res}[i], j))$
      $\mathtt{check} \leftarrow i$
  if $\mathtt{check} \neq \perp$
    return $\mathtt{res}[\mathtt{check}]$
  return $\perp$
```

Fig. 10. $\mathsf{OW\text{-}PCA}$-secure scheme $\Pi' = \mathsf{T}^\star[\Pi, \mathsf{G}]$ with deterministic encryption and correctness error δ^ℓ from $\mathsf{IND\text{-}CPA}$ secure scheme Π with correctness error δ.

$C'_j := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[h]; \mathsf{G}(\mathtt{res}[h], j)$. If for all $j \in [\ell]$ we have $C'_j = C_j$ we return $\mathtt{res}[h]$. If this is not the case we return $\perp$. Note that all ℓ C'_j have to be re-encrypted and checked against C_j, as otherwise IND-CCA-security is not achieved. The difference is, that only ℓ encryptions instead of ℓ^2 are required.

We now show the following theorem.

Theorem 7 (Π $\mathsf{IND\text{-}CPA} \implies \Pi'$ $\mathsf{OW\text{-}PCA}$)**.** *Assume Π to be δ-correct. Then, Π' is $\delta_1(q_\mathsf{G}, \ell) \leq \frac{q_\mathsf{G}}{\ell} \cdot \delta^\ell$ correct and for any* OW-PCA *adversary B that issues at most q_G queries to the random oracle* G *and q_P queries to a plaintext checking oracle* Pco*, there exists an* IND-CPA *adversary A running in about the same time as B such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',B}(\lambda) \leq \frac{q_\mathsf{G}}{\ell} \cdot \delta^\ell + \frac{2q_\mathsf{G} + 1}{|\mathcal{M}|} + 3\ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda).$$

We provide the proof which closely follows the proof of [41, Thm 3.2] in the full version. Note that we lose an additional factor of ℓ. Additionally, when using the bounded δ-correctness notion from Bindel. et al. [10], the factor of q_G disappears.

We now have an OW-PCA secure PKE Π' with negligible correctness error and can thus directly use $\mathsf{U}^{\not\perp}$ and by invoking Theorem 4 obtain an IND-CCA secure KEM KEM. Note that all steps in the reduction are tight. For the security in the QROM, we can directly conclude from Corollary 1 that the generic framework of Bindel et al. [9] can be applied to $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$ with the additional constraint of ε-injectivity and FFC, respectively. Without these additional constraints, the results of Jiang et al. [48] or Hövelmanns et al. [43][1] apply without the tighter reductions that the Bindel et al.'s and Kuchta et al.'s results offer.

The security of the $\mathsf{T}^\star$ transform in the QROM follows in a similar vein. To highlight how ℓ influences the advantages, we follow the proof strategy of Bindel et al. [9]. Therefore, we first show that a randomized IND-CPA-secure

[1] Without restating [43, Thm 3.2], note that we can adopt it the same way we highlight in Theorems 7 and 8. So, we start with their Punc to obtain disjoint simutability and then apply $\mathsf{T}^\star$ and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$.

PKE scheme with a non-negligible correctness error is transformed to $\mathsf{OW\text{-}CPA}$-secure deterministic PKE scheme with negligible correctness error. Second, we prove that if the $\mathsf{T}^\star$-transformed version is also ε-injective, then it provides FFC. With these two results in place, we can apply Theorem 6 to obtain an $\mathsf{IND\text{-}CCA}$-secure KEM.

In the following theorem, we prove $\mathsf{OW\text{-}CPA}$ security of the $\mathsf{T}^\star$ transform in the QROM (see the full version). We follow the strategy of the proof of [9, Thm. 1] and adapt it to our transform. Compared to the T transform, we lose a factor of ℓ^2. Once the loss is incurred by Theorem 1 and once by the semi-classical one-way to hiding Theorem [2].

Theorem 8 ($\mathit{\Pi}$ $\mathsf{IND\text{-}CPA}$ $\implies$ $\mathit{\Pi}'$ $\mathsf{OW\text{-}CPA}$). *Let $\mathit{\Pi}$ be a non-deterministic PKE with randomness space $\mathcal{R}$ and decryption error δ. Let $\ell \in \mathbb{N}$ such that δ^ℓ is negligible in the security parameter λ. Let $\mathsf{G}\colon \mathcal{M} \times [\ell] \to \mathcal{R}$ be a quantum-accessible random oracle and let q_G the number queries with depth at most d. If A is an $\mathsf{OW\text{-}CPA}$-adversary against $\mathsf{T}^\star[\mathit{\Pi}, \mathsf{G}, \ell]$, then there exists an $\mathsf{IND\text{-}CPA}$ adversary B against $\mathit{\Pi}$, running in about same time as A, such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\mathsf{T}^\star[\mathit{\Pi},\mathsf{G},\ell],A}(\lambda) \leq (d + \ell + 1)\left(\ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\mathit{\Pi},B}(\lambda) + \frac{8(q_\mathsf{G} + 1)}{|\mathcal{M}|}\right).$$

We refer to the full version for the proof. Next, we show that the transform provides the FFC property (cf. [9, Lemma 6]).

Lemma 2. *If $\mathit{\Pi}$ is a δ-correct non-deterministic PKE with randomness space $\mathcal{R}$, $\ell \in \mathbb{N}$ such that δ^ℓ is negligible in the security parameter λ, $\mathsf{G}\colon \mathcal{M} \times [\ell] \to \mathcal{R}$ is a random oracle so that $\mathit{\Pi}' = \mathsf{T}^\star[\mathit{\Pi}, \mathsf{G}, \ell]$ is ε-injective, then the advantage for any FFC-adversary A against $\mathit{\Pi}'$ which makes at most q_G queries at depth d to G and which returns a list of at most q_L ciphertexts is bounded by*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\mathit{\Pi}',A}(\lambda) \leq \left((4d + 1)\delta^\ell + \sqrt{3\varepsilon}\right)(q_\mathsf{G} + q_L) + \varepsilon.$$

For the proof we refer to the full version.

3.5 Comparison of the Two Approaches

The major difference between the generic approach using the direct product compiler $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$, and $\mathsf{T}^\star$ (or the modified deterministic direct product compiler $\mathsf{C}^\star_{\mathsf{p},\mathsf{d}}$) is the number of ciphertexts required to reach a negligible correctness error. As observed in Section 3.2, the analysis of the overall decryption error is rather complicated and $\mathsf{C}_{\mathsf{p},\mathsf{y}}$ requires at least $\ell \geq 3$. With $\mathsf{T}^\star/\mathsf{C}^\star_{\mathsf{p},\mathsf{d}}$ however, the situation is simpler. As soon as one ciphertext decrypts correctly, the overall correctness of the decryption can be guaranteed. Also, for the cases analysed in Table 1, $\mathsf{C}_{\mathsf{p},\mathsf{y}}$ requires at least one ciphertext more than $\mathsf{T}^\star$ and $\mathsf{C}^\star_{\mathsf{p},\mathsf{d}}$. For the correctness error, we have a loss in the number of random oracle queries in both cases. For the comparison of the runtime and bandwidth overheads, we refer to Table 2. Note that if the Dec of the underlying PKE $\mathit{\Pi}$ reports decryption failures with $\perp$, then the overhead of $\mathsf{T}^\star$ for Dec is only a factor ℓ (cf. Remark 2).

Table 2. Comparison of the runtime and bandwidth overheads of $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$, with ℓ ciphertexts and $\mathsf{T}^\star$ and $\mathsf{C}^\star_{\mathsf{p},\mathsf{d}}$ with ℓ' ciphertexts such that $\ell \geq \ell' + 1$.

	$\lvert\mathsf{pk}\rvert$	$\lvert C\rvert$	KGen	Enc	Dec
$\mathsf{C}_{\mathsf{p},\mathsf{y}}$	1 (r) / ℓ (d)	ℓ	1 (r) / ℓ (d)	ℓ	ℓ
$\mathsf{C}^\star_{\mathsf{p},\mathsf{d}}$	ℓ'	ℓ'	ℓ'	ℓ'	ℓ'
$\mathsf{T}^\star$	1	ℓ'	1	ℓ'	ℓ'^2 / ℓ' ($\perp$)

4 Our Transform in Practice

The most obvious use-case for IND-CCA secure KEMs in practice is when considering static long-term keys. Systems supporting such a setting are for example RSA-based key exchange for SSH [39] or similarly in TLS up to version 1.2. But since the use of long-term keys precludes forward-secrecy guarantees, using static keys is not desirable. For ephemeral keys such as used in the ephemeral Diffie-Hellman key exchange, an IND-CPA secure KEM might seem sufficient. Yet, in the post-quantum setting accidental re-use of an ephemeral key leads to a wide range of attacks [7]. But also from a theoretical viewpoint it is unclear whether CPA security actually would be enough. Security analysis of the TLS handshake protocol suggests that in the case of version 1.2 an only passively secure version is insufficient [45, 50] (cf. also [56]). Also, security analysis of the version 1.3 handshake requires IND-CCA security [22]. Thus, even in the case of ephemeral key exchanges, using a IND-CCA secure KEM is actually desirable and often even necessary as highlighted by Schwabe et al. [61].

For comparing KEMs in this context, the interesting metric is hence not the ciphertext size alone, but the combined public key and ciphertext size. Both parts influence the communication cost of the protocols. Additionally, the combined runtime of the key generation, encapsulation and decapsulation is also an interesting metric. All three operations are performed in a typical ephemeral key exchange and hence give a lower bound for the overall runtime of the protocol.

In the following comparison, we assume that the underlying PKE never returns $\perp$ on failure, but an incorrect message instead. Thereby we obtain an upper bound for the runtime of the Decaps algorithm. For specific cases where Decaps explicitly returns $\perp$ on failure, the runtime figures would get better since the overhead to check the ciphertexts is reduced to a factor of ℓ (cf. Remark 2).

4.1 Code-Based KEMs

KEMs based on error correcting codes can be parametrized such that the decoding failure rate (DFR) is non-negligible, negligible, or 0. Interestingly, the DFR rate is also influenced by the actual decoder. Even for the same choice of code and the exact same instance of the code, a decoder might have a non-negligible DFR, whereas another (usually more complex) decoder obtains a negligible DFR. For the submissions in the NIST PQC we can observe all three choices. The candidates providing IND-CPA-secure variants with non-negligible DFR include:

Table 3. Sizes (in bytes) and runtimes (in ms and millions of cycles for BIKE), where O denotes the transformed scheme. The LEDAcrypt instances with postfix NN refer to those with non-negligible DFR. Runtimes are taken from the respective submission documents and are only intra-scheme comparable.

KEM	δ	pk	C	$\sum$	KGen	Encaps	Decaps
O[ROLLO-I-L1,5]	2^{-150}	**465**	2325	**2790**	**0.10**	**0.02**/0.10	**0.26**/1.30
ROLLO-II-L1	2^{-128}	1546	1674	3220	0.69	0.08	0.53
O[ROLLO-I-L3,4]	2^{-128}	**590**	2360	**2950**	**0.13**	**0.02**/**0.08**	**0.42**/1.68
ROLLO-II-L3	2^{-128}	2020	2148	4168	0.83	0.09	0.69
O[ROLLO-I-L5,4]	2^{-168}	**947**	7576	8523	**0.20**	**0.03**/0.12	**0.78**/3.12
ROLLO-II-L5	2^{-128}	2493	2621	5114	0.79	0.10	0.84
O[BIKE-2-L1,3]	2^{-147}	**10163**	30489	40652	**4.79**	**0.14**/0.42	**3.29**/9.88
BIKE-2-CCA-L1	2^{-128}	11779	12035	23814	6.32	0.20	4.12
O[LEDAcrypt-L5-NN,2]	2^{-128}	22272	22272	44544	5.04	**0.14**/**0.29**	**1.55**/3.11
LEDAcrypt-L5	2^{-128}	19040	19040	38080	4.25	0.84	2.28

BIKE [3], ROLLO [4], and LEDAcrypt [6]. We discuss the application of our transform to those schemes below. For the comparison in Table 3, we consider the DFR as upper bound for correctness error.

In Table 3, we present an overview of the comparison (see the full version for the full comparison). First we consider ROLLO, and in particular ROLLO-I, where we obtain the best results: public key and ciphertext size combined is always smaller than for ROLLO-II and the parallel implementation is faster even in case of a ℓ^2 overhead. For both BIKE (using $\mathsf{T}^{\star}$) and LEDAcrypt (using $\mathsf{C}^{\star}_{\mathsf{p,d}}$ since it starts from a deterministic PKE), we observe a trade-off between bandwidth and runtime.

4.2 Lattice-Based KEMs

For lattice-based primitives the decryption error depends both on the modulus q and the error distribution used. As discussed in [60], an important decision that designers have to make is whether to allow decryption failures or choose parameters that not only have a negligible, but a zero chance of failure. Having a perfectly correct encryption makes transforms to obtain $\mathsf{IND\text{-}CCA}$ security and security proofs easier, but with the disadvantage that this means either decreasing security against attacks targeting the underlying lattice problem or decreasing performance. The only NIST PQC submissions based on lattices which provide parameter sets achieving both negligible and non-negligible decryption failure are ThreeBears [38] and Round5 [30]. The $\mathsf{IND\text{-}CCA}$-secure version of ThreeBears is obtained by tweaking the error distribution, hence, our approach does not yield any improvements. For Round5 we achieve a trade-off between bandwidth and runtime. We also considered FrodoKEM [52], comparing its version [17] precedent to the NIST PQC, which only achieved non-negligible failure probability, to the ones in the second round of the above competition, but we do not observe any improvements for this scheme. For the full comparison we refer to the full version. It would be interesting to understand the reasons why the compiler does not perform well on lattice-based scheme compared to the code-based ones and whether this is due to the particular schemes analysed or due to some intrinsic difference between code- and lattice-based constructions.

4.3 Implementation Aspects

One of the strengths of $\mathsf{T}^\star$ compared to the black-box use of $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$ (and $\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$), is that besides the initial generation of the encapsulated key, all the random oracle calls can be evaluated independently. Therefore, the encryptions of the underlying PKE do not depend on each other. Thus, the encapsulation algorithms are easily parallelizable – both in software and hardware. The same applies to the decapsulation algorithm. While in this case only one successful run of the algorithm is required, doing all of them in parallel helps to obtain a constant-time implementation. Then, after all ciphertexts have been processed, the first valid one can be used to re-compute the ciphertexts, which can be done again in parallel. For software implementations on multi-core CPUs as seen on today's desktops, servers, and smartphones with 4 or more cores, the overhead compared to the IND-CPA secure version is thus insignificant as long as the error is below 2^{-32}. If not implemented in a parallel fashion, providing a constant-time implementation of the decapsulation algorithms is more costly. In that case, all of the ciphertexts have to be dealt with to not leak the index of invalid ciphertexts. Note that a constant-time implementation of the transform is important to avoid key-recovery attacks [35].

The $\mathsf{T}^\star$ transform also avoids new attack vectors such as [37] that are introduced via different techniques to decrease the correctness error, e.g., by applying an error-correcting code on top. Furthermore, since the same parameter sets are used for the IND-CPA and IND-CCA secure version when applying our transforms, the implementations of proposals with different parameter sets can be simplified. Thus, more focus can be put on analysing one of the parameter sets and also on optimizing the implementation of one of them.

5 Application to Bloom Filter KEMs

A Bloom Filter Key Encapsulation Mechanism (BFKEM) [21, 20] is a specific type of a puncturable encryption scheme [33, 34, 21, 63] where one associates a Bloom Filter (BF) [13] to its public-secret key pair. The initial (i.e., non-punctured) secret key is associated to an empty BF where all bits are set to 0. Encapsulation, depending on an element s in the universe of the BF, takes the public key and returns a ciphertext and an encapsulation key k corresponding to the evaluation of $BF(s)$, i.e., k hash evaluations on s yielding indexes in the size m of the BF. Puncturing, on input a ciphertext C (associated to s) and a secret key sk', punctures sk' on C and returns the resulting secret key. Decapsulation, on input a ciphertext C (with an associated tag s) and secret key sk' is able to decapsulate the ciphertext to k if sk' was not punctured on C. We want to mention, as in [20], we solely focus on KEMs since a Bloom Filter Encryption (BFE) scheme (which encrypts a message from some message space) can be generically derived from a BFKEM (cf. [27]).

The basic instantiation of a BFKEM in [21, 20] is non-black box and based on the pairing-based Boneh-Franklin IBE (BF-IBE) scheme [16], where sk contains an IBE secret key for every identity $i \in [m]$ of the BF bits and puncturing amounts to inserting s in the BF and deleting the IBE secret keys for the corresponding bits. Although the BFKEM is defined with respect to a non-negligible correctness error, the underlying BF-IBE has perfect correctness. So the non-negligible error in the BFKEM is only introduced on an abstraction (at the level of the BF) above the FO transform applied to the k BF-IBE ciphertexts (so the application of the FO can be done as usual for perfectly correct encryption schemes).

However, if one targets instantiations of BFE where the underlying IBE does not have perfect correctness (e.g., lattice- or code-based IBEs), it is not obvious whether the security proof using the BF-IBE as presented in [21, 20] can easily be adapted to this setting.[2]

We first recall necessary definitions and then show a generic construction of BFKEM from any IBE scheme with (non-)negligible correctness error.

Due to space constraints, we present the definition of Bloom filters with its formal properties in the full version.

Bloom Filter key encapsulation mechanism. We recap the Bloom Filter Key Encapsulation Mechanism (BFKEM) and its formal properties from [20] that tolerates a non-negligible correctness error and the key generation takes parameters m and k as input which specify this correctness error. A BFKEM BFKEM with key space $\mathcal{K}$ consists of the PPT algorithms $(\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$.

$\mathsf{KGen}(\lambda, m, k)$: Key generation, on input security parameter λ and BF parameters m, k, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk}_0)$.

$\mathsf{Encaps}(\mathsf{pk})$: Encapsulation, on input pk, outputs a ciphertext C and key k.

$\mathsf{Punc}(\mathsf{sk}, C)$: Secret-key puncturing, on input sk and C, outputs an updated secret key sk'.

$\mathsf{Decaps}(\mathsf{sk}, C)$: Decapsulation, on input sk and C, outputs k or $\{\bot\}$.

Definition 8 (Correctness). *For all $\lambda, m, k, n \in \mathbb{N}$ and any $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$, we require the following. For any (arbitrary interleaved) sequence of invocations of $\mathsf{sk}_{j+1} \leftarrow \mathsf{Punc}(\mathsf{sk}_j, C_j)$, where $j \in \{0, \ldots, n\}$, and $(C_j, \mathsf{k}_j) \leftarrow$*

[2] Note that we want the size of the BFKEM public key to be independent of the BF parameters for practical reasons (besides the descriptions of the hash functions). Right now, we only can guarantee this with IBE schemes as such schemes allow for exponentially many secret keys with a short master public key and, hence, we consider IBE schemes as a main building block of our BFKEM constructions.

$\mathsf{Encaps}(\mathsf{pk})$*, it holds that*

$$\Pr\left[\mathsf{Decaps}(\mathsf{sk}_{n+1}, C^*) \neq \mathsf{k}^*\right] \leq \left(1 - e^{-\frac{(n+1/2)k}{m-1}}\right)^k + \varepsilon(\lambda),$$

where $(C^*, \mathsf{k}^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ *and* $\varepsilon(\cdot)$ *is a negligible function in* λ*.*

Definition 9 (Extended Correctness). *For all* $\lambda, m, k, n \in \mathbb{N}$ *and any* $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$*, we require the following. For any (arbitrary interleaved) sequence of invocations of* $\mathsf{sk}_{j+1} \leftarrow \mathsf{Punc}(\mathsf{sk}_j, C_j)$ *where* $j \in \{0, \ldots, n\}$ *and* $(C_j, \mathsf{k}_j) \leftarrow \mathsf{Encaps}(\mathsf{pk})$*, it holds that:*

1. *Impossibility of false-negatives:* $\mathsf{Decaps}(\mathsf{sk}_{n+1}, C_j) = \bot$ *for all* $j \leq n$*.*
2. *Perfect correctness of the initial secret key:* $\mathsf{Decaps}(\mathsf{sk}, C) = \mathsf{k}$ *for all* $(C, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$*.*
3. *Semi-correctness of punctured secret keys: If* $\mathsf{Decaps}(\mathsf{sk}_{j+1}, C) \neq \bot$ *then* $\mathsf{Decaps}(\mathsf{sk}_{j+1}, C) = \mathsf{Decaps}(\mathsf{sk}_0, C)$*.*

All probabilities are taken over the random coins of KGen, Punc, and Encaps. We recall additional properties (i.e., separable randomness, publicly-checkable puncturing, and γ-spreadness) and formal definitions of BFKEM-IND-CPA and BFKEM-IND-CCA security in the full version.

5.1 IBE with Negligible from Non-Negligible Correctness Error

We follow the approach for randomized PKE schemes in Section 3.2 adapted for the IBE case (cf. Figure 11).[3] Let $\mathsf{IBE} = (\mathsf{KGen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme with identity, message spaces, and randomness spaces $\mathcal{ID}$, $\mathcal{M}$, and $\mathcal{R}$, respectively, with *non-negligible correctness error* $\delta(\lambda)$, we construct an IBE scheme $\mathsf{IBE}' = (\mathsf{KGen}', \mathsf{Ext}', \mathsf{Enc}', \mathsf{Dec}')$ with identity and message spaces $\mathcal{ID}' := \mathcal{ID}$ and $\mathcal{M}' := \mathcal{M}$, respectively, with *negligible correctness error* $\delta'(\lambda)$. The construction is as follows. Set $\mathsf{KGen}' := \mathsf{KGen}$ and $\mathsf{Ext}' := \mathsf{Ext}$ while Enc' and Dec' are given in Figure 11. See that $\ell = \ell(\lambda)$ can be chosen appropriately to accommodate a negligible correctness error $\delta'(\lambda)$.

As for randomized PKE schemes, by an analogue of Theorem 1 for IBEs with $q = \ell$ and $n = 1$, the security claim follows.

Corollary 2. *For any IBE-sIND-CPA adversary* B *against* IBE' *obtained via applying the above transformation to* IBE*, there exists an IBE-sIND-CPA adversary* A *such that:*

$$\mathsf{Adv}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE}',B}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},A}(\lambda).$$

The correctness error analysis is again equivalent to the one in the PKE scenario. We refer to Section 3.2 for a more in depth discussion.

[3] We explicitly mention that we are only concerned with randomized IBEs. Adopting $\mathsf{C}_{\mathsf{p,d}}$ for deterministic IBEs will work as well. Though in the latter case, one can further optimize the compiler depending on whether the IBE has deterministic or randomized key extraction Ext.

```
$\mathsf{Enc}'(mpk, id, M)$
  for $i \in [\ell]$
    $r_i \leftarrow_\$ \mathcal{R}$
    $C_i \leftarrow \mathsf{Enc}(mpk, id, M; r_i)$
  return $(C_1, \ldots, C_\ell)$
```

```
$\mathsf{Dec}'(usk_{id}, C)$
  $C =: (C_1, \ldots, C_\ell)$
  for $i \in [\ell]$
    $M'_i := \mathsf{Dec}(usk_{id}, C_i)$
  return $\mathsf{maj}(M'_1, \ldots, M'_\ell)$
```

Fig. 11. Compiler for Enc' and Dec' for constructing IBE with negligible correctness error from IBE with non-negligible correctness error.

5.2 BFKEM from IBE with Negligible Correctness Error

The intuition for our generic construction from any IBE with negligible correctness error is as follows. We associate "user-secret keys" of IBE with the indexes $i \in [m]$ of the Bloom filter BF and annotate sk'_0 as a special key for "fixed identity" 0. We consider the encapsulation key as $\mathsf{k}_0 \oplus \mathsf{k}_1$ where one share is encrypted under "identity" 0 (yielding C_0) while the other share is encrypted under the "identities" $(i_j)_j$ of indexes of the BF that are determined by C_0. Put differently, C_0 acts as a tag of the overall ciphertext while the other IBE-ciphertexts $(C_{i_j})_j$ are utilized for correct decryption. The secret key is punctured on "tag" C_0. Note that the secret key sk'_0 is not affected by the puncturing mechanism and one can always at least decrypt C_0. However, one additionally needs the encapsulation-key share from the other ciphertexts $(C_{i_j})_j$; those ciphertexts can only be decrypted if at least one secret key sk'_{i*} is available which can be checked with $\mathsf{BFCheck}$.

Let $\mathsf{IBE} = (\mathsf{IBE.KGen}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ be an IBE-sIND-CPA-secure IBE scheme with identity and message spaces $\mathcal{ID} = [m] \cup \{0\}$ and $\mathcal{M} = \{0,1\}^\lambda$, respectively, with negligible correctness error $\delta = \delta(\lambda)$, and $\mathsf{BF} = (\mathsf{BFGen}, \mathsf{BFUpdate}, \mathsf{BFCheck})$ a BF with universe $\mathcal{U}$, we construct a BFKEM-IND-CPA-secure BFKEM scheme $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$ with key space $\mathcal{K} := \mathcal{M} = \{0,1\}^\lambda$ as a stepping stone towards a BFKEM-IND-CCA-secure BFKEM as follows.

$\mathsf{KGen}(\lambda, m, k)$: on input security parameter λ and BF parameters m, k, compute $(mpk, msk) \leftarrow \mathsf{IBE.KGen}(\lambda)$, $\mathsf{sk}'_{id} \leftarrow \mathsf{IBE.Ext}(msk, id)$, for all $id \in [m] \cup \{0\}$, and $(H, T_0) \leftarrow \mathsf{BFGen}(m, k)$. Return $\mathsf{pk} := (mpk, H)$ and $\mathsf{sk} := (T_0, (\mathsf{sk}'_{id})_{id})$ (we assume that pk is available to Punc and Decaps implicitly).

$\mathsf{Encaps}(\mathsf{pk})$: on input $(mpk, H) := \mathsf{pk}$, sample $\mathsf{k}_0, \mathsf{k}_1 \leftarrow_\$ \mathcal{K}$ and compute $C_0 \leftarrow \mathsf{Enc}(mpk, 0, \mathsf{k}_0)$. For $id_j := H_j(C_0)$ with $(H_j)_j := H$ and all $j \in [k]$, compute $C_{id_j} \leftarrow \mathsf{Enc}(mpk, id_j, \mathsf{k}_1)$ and output

$$((C_0, (C_{id_j})_j), \mathsf{k}_0 \oplus \mathsf{k}_1).$$

$\mathsf{Punc}(\mathsf{sk}, C)$: on input $(T, \mathsf{sk}'_0, (\mathsf{sk}'_{id})_{id \in [m]}) := \mathsf{sk}$ and $(C_0, \ldots) := C$, compute $T' := \mathsf{BFUpdate}(H, T, C_0)$ and set

$$\mathsf{sk}''_{id} := \begin{cases} \mathsf{sk}'_{id} & \text{if } T'[id] = 0, \\ \bot & \text{if } T'[id] = 1, \end{cases}$$

for $T'[id]$ the id-th bit of T'. Return $(T', \mathsf{sk}'_0, (\mathsf{sk}''_{id})_{id \in [m]})$.

$\mathsf{Decaps}(\mathsf{sk}, C)$: on input $(T, (\mathsf{sk}'_{id})_{id \in [m] \cup \{0\}}) := \mathsf{sk}$ and $(C_0, (C_{id_j})_{j \in [k]}) := C$, output $\perp$ if $\mathsf{BFCheck}(H, T, C_0) = 1$. Otherwise, there exists a smallest $id^* \in [m]$ such that $\mathsf{sk}'_{id^*} \neq \perp$, compute $\mathsf{k}_0 := \mathsf{Dec}(\mathsf{sk}'_0, C_0)$ and $\mathsf{k}_1 := \mathsf{Dec}(\mathsf{sk}'_{id^*}, C_{id^*})$, and output $\mathsf{k}_0 \oplus \mathsf{k}_1$.

We prove the correctness (Definition 8), extended correctness (Definition 9), separable randomness, publicly-checkable puncturing, and γ-spreadness properties of BFKEM in the full version.

BFKEM-IND-CPA security of BFKEM. We start by showing the BFKEM-IND-CPA security of $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$.

Theorem 9. *If IBE is IBE-sIND-CPA-secure, then BFKEM is BFKEM-IND-CPA-secure. Concretely, for any PPT adversary A there is a distinguisher D for the IBE-sIND-CPA security experiment such that*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},A}(\lambda, m, k) \leq k \cdot m \cdot \mathsf{Adv}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},D}(\lambda). \tag{1}$$

Proof. We show the BFKEM-IND-CPA-security of BFKEM for any valid PPT adversary A in series of games where:

Game 0. Game 0 is the BFKEM-IND-CPA-experiment.

Game i. Game i is defined as Game $i-1$ except that the i-th challenge-ciphertext element C_{id_i} in C^* is independent of the challenge bit, for $i \in [k]$.

Game $k+1$. Game $k+1$ is defined as Game k except that the encapsulation key in the challenge ciphertext is independent of b'.

We denote the event of the adversary winning Game i as S_i. In Game $k+1$, A has no advantage (i.e., success probability of $\Pr[S_{k+1}] = 1/2$) in the sense of BFKEM-IND-CPA. We argue in hybrids that the Games $i \in [k+1]$ are computationally indistinguishable from Game 0.

Hybrids between Games 0 and $k+1$. Each hybrid between Games $i-1$ and i, $i \in [k]$, is constructed as follows: on input m and k, D samples $(H, T_0) \leftarrow \mathsf{BFGen}(m, k)$, for $H =: (H_j)_{j \in [k]}$ and sets $T_0 = 0^m$. Next, D samples $id^* \leftarrow_{\$} [m]$ and sends id^* to its IBE-sIND-CPA-challenger. D retrieves mpk in return and sets $\mathsf{pk} := (mpk, H)$.

Furthermore, for all $id \in ([m] \cup \{0\}) \setminus \{id^*\}$, D retrieves $\mathsf{sk}_0 := (usk_{id})_{id}$ from its Ext-oracle. (Note that D does not have a secret key for id^* and A has to query the challenge ciphertext to the Punc'-oracle in order to receive secret keys via the Cor-oracle, which results in "deleting" the secret key for id^* if there were any. Particularly, all Cor-queries can be answered correctly.)

Furthermore, D sends $\mathsf{k}_1^{(0)}, \mathsf{k}_1^{(1)} \leftarrow_{\$} \mathcal{M} = \{0,1\}^\lambda$ to its IBE-sIND-CPA-challenger and retrieves $C^*_{id^*} \leftarrow \mathsf{Enc}(mpk, id^*, \mathsf{k}^{(b)})$, for some $b \leftarrow_{\$} \{0,1\}$.

D samples $b' \leftarrow_{\$} \{0,1\}$, computes $C_0 \leftarrow \mathsf{Enc}(mpk, 0, \mathsf{k}_0)$, for $\mathsf{k}_0 \leftarrow \mathcal{M}$, and sets $(id_j)_j := (H_j(C_0))_{j \in [k]}$. If $id_i \neq id^*$, abort. (See that this happens with probability $(m-1)/m$.) Otherwise, D computes $C_{id_j} \leftarrow \mathsf{Enc}(mpk, id_j, \mathsf{k}_1^{(b')})$, for all $(id_j)_{j \in [k] \setminus [i]}$, and $C_{id_j} \leftarrow \mathsf{Enc}(mpk, id_j, \mathsf{k}'_1)$, for all $(id_j)_{j \in [i-1]}$, for $\mathsf{k}'_1 \leftarrow_{\$} \mathcal{M}$.

D sets $C_{id_i} := C^*_{id^*}$ and sends $(\mathsf{pk}, C^* := (C_0, (C_{id_j})_j), \mathsf{k}^{(b')})$ to A, for $\mathsf{k}^{(b')} := \mathsf{k}_1^{(b')} \oplus \mathsf{k}_0$.

A has access to a $\mathsf{Punc}'(C)$-oracle which runs $\mathsf{sk}_{i+1} \leftarrow \mathsf{Punc}(\mathsf{sk}_i, C)$ for each invocation $i = 0, 1, \ldots, q$ and sets $\mathcal{L} := \mathcal{L} \cup \{C\}$ for initially empty set $\mathcal{L}$. The Cor-oracle returns sk_{i+1} iff $C^* \in \mathcal{L}$. Eventually, A outputs a guess b^* which D forwards as $b^* \oplus b'$ to its IBE-sIND-CPA-challenger.

In the hybrid between Games k and $k+1$: proceed as in Game k, but send $(\mathsf{pk}, C^* := (C_0, (C_{id_j})_j), \mathsf{k}')$, for uniform $\mathsf{k}' \leftarrow \mathcal{M}$ to A.

Analysis. In the hybrids between the Games $j-1$ and j, for $j \in [k]$, we have that if $b' = b = 0$ or $b' = b = 1$, then the distribution of the challenge ciphertext is correct and a successful A should output $b^* = 0$ where D forwards $b^* \oplus b' = b$ as guess to its challenger which yields a successful IBE-sIND-CPA distinguisher D. If $b' \neq b$, then A is used to distinguish the j-th challenge-ciphertext component, i.e., a successful A should output $b^* = 1$ where D forwards $b^* \oplus b' = b$ as guess to its challenger which, again, yields a successful IBE-sIND-CPA distinguisher D. In the hybrid between the Games k and $k+1$, the change is information-theoretic, i.e., the challenge ciphertext encapsulates uniformly random key-elements (independent of b') and the encapsulation key is sampled uniformly at random which yields $\Pr[S_{k+1}] = 1/2$. In each hybrid, we have that $\Pr[id_i = id^*] = 1/m$. Putting things together, for $k+1$ hybrids, Equation (1) holds. □

BFKEM-IND-CCA security of BFKEM'. We construct a slight variant of our BFKEM scheme above, dubbed BFKEM', via the FO transform [27] along the lines of [21]. We want to mention that the FO transform does not work generically for any BFKEM and no generic framework as in the case of KEMs exists. Hence, we consider the direct product compiler in Section 5.1 and, in the vein of [21], to prove BFKEM-IND-CCA security of our BFKEM, we introduce further properties (i.e., separable randomness, publicly-checkable puncturing, and γ-spreadness) in the full version. Furthermore, [21] requires perfect correctness for unpunctured keys which our BFKEM definition cannot guarantee. Hence, we have to reprove the BFKEM-IND-CCA-security for BFKEM', although the proof techniques are almost the same as presented in [21]. We construct a BFKEM-IND-CCA-secure BFKEM as follows. Let $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$ be a randomness-separable BFKEM-IND-CPA-secure BFKEM scheme with key space $\mathcal{K} = \{0,1\}^\lambda$ and correctness error $\delta = \delta(\lambda)$, we construct a BFKEM-IND-CCA-secure BFKEM scheme $\mathsf{BFKEM}' = (\mathsf{KGen}', \mathsf{Encaps}', \mathsf{Punc}', \mathsf{Decaps}')$ with key space $\mathcal{K}' = \mathcal{K}$ using a variant of the FO transform as follows. Let $\mathsf{G} \colon \mathcal{K}' \to \{0,1\}^{\rho+\lambda}$ be a hash function modeled as random oracle (RO) in the security proof.

$\mathsf{KGen}'(\lambda, m, k)$: same as $\mathsf{KGen}(\lambda, m, k)$.

$\mathsf{Encaps}'(\mathsf{pk})$: on input pk, sample $\mathsf{k}' \leftarrow_\$ \mathcal{K}'$, compute $(r, \mathsf{k}) := \mathsf{G}(\mathsf{k}') \in \{0,1\}^{\rho+\lambda}$ and $(C, \mathsf{k}') \leftarrow \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}'))$, and return (C, k).

$\mathsf{Punc}'(\mathsf{sk}, C)$: same as $\mathsf{Punc}(\mathsf{sk}, C)$.

$\mathsf{Decaps}'(\mathsf{sk}, C)$: on input secret key sk and ciphertext C, compute $\mathsf{k}' \leftarrow \mathsf{Decaps}(\mathsf{sk}, C)$ and return $\perp$ if $\mathsf{k}' = \perp$. Otherwise, compute $(r, \mathsf{k}) := \mathsf{G}(\mathsf{k}')$ and return k if $(C, \mathsf{k}') = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}'))$, else output $\perp$.

We prove the correctness (Definition 8), extended correctness (Definition 9), separable randomness, publicly-checkable puncturing, and γ-spreadness properties of BFKEM' in the full version.

Theorem 10. *If a BFKEM* BFKEM *is BFKEM-IND-CPA-secure with the separable randomness, publicly-checkable puncturing, and* γ*-spreadness properties, and negligible correctness error probability* $\delta = \delta(\lambda)$*, then* BFKEM' *is BFKEM-IND-CCA-secure. Concretely, for any PPT adversary* A *making at most* $q_{\mathsf{G}} = q_{\mathsf{G}}(\lambda)$ *queries to the random oracle* G *there is a distinguisher* D *in the BFKEM-IND-CPA-security experiment such that*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cca}}_{\mathsf{BFKEM}',A}(\lambda, m, k) \leq \mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},D}(\lambda, m, k) + 2\delta + \frac{q_{\mathsf{G}}}{2^{\gamma}}. \tag{2}$$

Due to space constraints, we show the proof in the full version.

5.3 Comparison of BFKEM Instantiations

To instantiate BFKEM' from post-quantum IBE schemes, we investigating instantiations based on a selectively IND-CPA secure lattice-based or code-based IBEs. As far as lattices are concerned, the first such construction was [31] after which numerous others followed [1, 19, 24, 64]. To compute the dimension of a lattice-based BFKEM, we start from the GVP-IBE instantiation of [24], for which an implementation and concrete dimensions were given for 80 and 192-bit quantum security. We set the parameter of the BFKEM as in [21], i.e., targeting the maximum number of allowed punctures to $n = 2^{20}$, which amounts to adding 2^{12} elements per day to the BF for a year, and allowing for a false-positive probability of 10^{-3}, we obtain $m = 1.5 \cdot 10^7$ and $k = 10$. A similar procedure can be applied to the code-based IBE of Gaborit et al. (GHPT) [29] achieving 128-bit quantum security. We note though that with recent advances in the cryptanalysis, these instances may provide less security. Table 4 provides an overview including the pairing-based BFKEM from [21]. For the latter, we assume the use of the pairing-friendly BLS12-381 curve with 120-bit classical security.

Table 4. Sizes of BFKEM when instantiated with GVP or GHPT.

IBE	assumption	sk	pk	C
GVP-80	lattice-based	19.21 GB	1.62 KB	17.46 KB
GVP-192	lattice-based	47.15 GB	3.78 KB	40.28 KB
GHPT-128	code-based	643.73 GB	252 KB	215.79 MB
Boneh-Franklin [21]	pairing-based	717.18 MB	95.5 B	255.5 B

Acknowledgements. We would like to thank the anonymous reviewers for their helpful comments. This project has received funding from the European

Union's Horizon 2020 research and innovation programme under grant agreement n°783119 (SECREDAS), n°826610 (COMP4DRONES), n°871473 (KRAKEN) and by the Austrian Science Fund (FWF) and netidee SCIENCE grant P31621-N38 (PROFET).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010
2. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: CRYPTO 2019, Part II
3. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Gueron, S., Guneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P., Zémor, G., Vasseur, V.: BIKE. Tech. rep., National Institute of Standards and Technology
4. Aragon, N., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.P., Zémor, G., Aguilar Melchor, C., Bettaieb, S., Bidoux, L., Bardet, M., Otmani, A.: ROLLO. Tech. rep., National Institute of Standards and Technology
5. Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J.A., Dukhovni, V., Käsper, E., Cohney, S., Engels, S., Paar, C., Shavitt, Y.: DROWN: Breaking TLS using SSLv2. In: USENIX Security 2016
6. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, P.: LEDAcrypt. Tech. rep., National Institute of Standards and Technology
7. Bauer, A., Gilbert, H., Renault, G., Rossi, M.: Assessment of the key-reuse resilience of NewHope. In: CT-RSA 2019
8. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: EUROCRYPT 2000
9. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: TCC 2019, Part II
10. Bindel, N., Schanck, J.M.: Decryption failure is more likely after success. In: PQCrypto 2020
11. Bitansky, N., Vaikuntanathan, V.: A note on perfect correctness by derandomization. In: EUROCRYPT 2017, Part II
12. Bleichenbacher, D.: Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In: CRYPTO'98
13. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM
14. Böck, H., Somorovsky, J., Young, C.: Return of bleichenbacher's oracle threat (ROBOT). In: USENIX Security 2018
15. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT 2011
16. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: CRYPTO 2001
17. 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: ACM CCS 2016
18. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: EUROCRYPT 2004

19. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: EUROCRYPT 2010
20. Derler, D., Gellert, K., Jager, T., Slamanig, D., Striecks, C.: Bloom filter encryption and applications to efficient forward-secret 0-rtt key exchange. IACR Cryptol. ePrint Arch. (to appear in Journal of Cryptology)
21. Derler, D., Jager, T., Slamanig, D., Striecks, C.: Bloom filter encryption and applications to efficient forward-secret 0-RTT key exchange. In: EUROCRYPT 2018, Part III
22. Dowling, B., Fischlin, M., Günther, F., Stebila, D.: A cryptographic analysis of the TLS 1.3 handshake protocol candidates. In: ACM CCS 2015
23. Drucker, N., Gueron, S., Kostic, D., Persichetti, E.: On the applicability of the fujisaki-okamoto transformation to the bike kem. IACR ePrint 2020/510
24. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: ASIACRYPT 2014, Part II
25. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: EUROCRYPT 2004
26. Fabsic, T., Hromada, V., Stankovski, P., Zajac, P., Guo, Q., Johansson, T.: A reaction attack on the QC-LDPC McEliece cryptosystem. In: PQCrypto 2017
27. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: CRYPTO'99
28. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology
29. Gaborit, P., Hauteville, A., Phan, D.H., Tillich, J.P.: Identity-based encryption from codes with rank metric. In: CRYPTO 2017, Part III
30. Garcia-Morchon, O., Zhang, Z., Bhattacharya, S., Rietman, R., Tolhuizen, L., Torre-Arce, J.L., Baan, H., Saarinen, M.J.O., Fluhrer, S., Laarhoven, T., Player, R.: Round5. Tech. rep., National Institute of Standards and Technology
31. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: 40th ACM STOC
32. Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In: CRYPTO'97
33. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: 2015 IEEE Symposium on Security and Privacy
34. Günther, F., Hale, B., Jager, T., Lauer, S.: 0-RTT key exchange with full forward secrecy. In: EUROCRYPT 2017, Part III
35. Guo, Q., Johansson, T., Nilsson, A.: A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. In: CRYPTO 2020, Part II
36. Guo, Q., Johansson, T., Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In: ASIACRYPT 2016, Part I
37. Guo, Q., Johansson, T., Yang, J.: A novel CCA attack using decryption errors against LAC. In: ASIACRYPT 2019, Part I
38. Hamburg, M.: Three Bears. Tech. rep., National Institute of Standards and Technology
39. Harris, B.: RSA key exchange for the secure shell (SSH) transport layer protocol. RFC
40. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: TCC 2017, Part I
41. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. Cryptology ePrint Archive, Report 2017/604

42. Hohenberger, S., Koppula, V., Waters, B.: Chosen ciphertext security from injective trapdoor functions. In: CRYPTO 2020, Part I
43. Hövelmanns, K., Kiltz, E., Schäge, S., Unruh, D.: Generic authenticated key exchange in the quantum random oracle model. In: PKC 2020, Part II
44. Howgrave-Graham, N., Nguyen, P.Q., Pointcheval, D., Proos, J., Silverman, J.H., Singer, A., Whyte, W.: The impact of decryption failures on the security of NTRU encryption. In: CRYPTO 2003
45. Jager, T., Kohlar, F., Schäge, S., Schwenk, J.: On the security of TLS-DHE in the standard model. In: CRYPTO 2012
46. Jager, T., Schinzel, S., Somorovsky, J.: Bleichenbacher's attack strikes again: Breaking PKCS#1 v1.5 in XML encryption. In: ESORICS 2012
47. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: CRYPTO 2018, Part III
48. Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In: PQCrypto 2019
49. Koppula, V., Waters, B.: Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In: CRYPTO 2019, Part II
50. Krawczyk, H., Paterson, K.G., Wee, H.: On the security of the TLS protocol: A systematic analysis. In: CRYPTO 2013, Part I
51. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.: Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In: EUROCRYPT 2020, Part III
52. Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM. Tech. rep., National Institute of Standards and Technology
53. Naor, M.: Bit commitment using pseudo-randomness. In: CRYPTO'89
54. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC
55. Nisan, N., Wigderson, A.: Hardness vs randomness. J. Comput. Syst. Sci.
56. Paquin, C., Stebila, D., Tamvada, G.: Benchmarking post-quantum cryptography in TLS. In: PQCrypto 2020
57. Ronen, E., Gillham, R., Genkin, D., Shamir, A., Wong, D., Yarom, Y.: The 9 lives of bleichenbacher's CAT: New cache ATtacks on TLS implementations. In: 2019 IEEE Symposium on Security and Privacy
58. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: EUROCRYPT 2018, Part III
59. Samardjiska, S., Santini, P., Persichetti, E., Banegas, G.: A reaction attack against cryptosystems based on LRPC codes. In: LATINCRYPT 2019
60. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D.: CRYSTALS-KYBER. Tech. rep., National Institute of Standards and Technology
61. Schwabe, P., Stebila, D., Wiggers, T.: Post-quantum tls without handshake signatures. Cryptology ePrint Archive, Report 2020/534
62. Sendrier, N., Vasseur, V.: On the decoding failure rate of QC-MDPC bit-flipping decoders. In: PQCrypto 2019
63. Sun, S., Sakzad, A., Steinfeld, R., Liu, J.K., Gu, D.: Public-key puncturable encryption: Modular and compact constructions. In: PKC 2020, Part I
64. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In: CRYPTO 2016, Part III