A Modular Analysis of the Fujisaki-Okamoto Transformation

Dennis Hofheinz [1] Kathrin Hövelmanns [2] Eike Kiltz [2]

August 8, 2017

[1] Karlsruhe Institute of Technology
Dennis.Hofheinz@kit.edu
[2] Ruhr Universität Bochum
{Kathrin.Hoevelmanns,Eike.Kiltz}@rub.de

Abstract

The Fujisaki-Okamoto (FO) transformation (CRYPTO 1999 and Journal of Cryptology 2013) turns any weakly secure public-key encryption scheme into a strongly (i.e., IND-CCA) secure one in the random oracle model. Unfortunately, the FO analysis suffers from several drawbacks, such as a non-tight security reduction, and the need for a perfectly correct scheme. While several alternatives to the FO transformation have been proposed, they have stronger requirements, or do not obtain all desired properties.

In this work, we provide a fine-grained and modular toolkit of transformations for turning weakly secure into strongly secure public-key encryption schemes. All of our transformations are robust against schemes with correctness errors, and their combination leads to several tradeoffs among tightness of the reduction, efficiency, and the required security level of the used encryption scheme. For instance, one variant of the FO transformation constructs an IND-CCA secure scheme from an IND-CPA secure one with a tight reduction and very small efficiency overhead. Another variant assumes only an OW-CPA secure scheme, but leads to an IND-CCA secure scheme with larger ciphertexts.

We note that we also analyze our transformations in the quantum random oracle model, which yields security guarantees in a post-quantum setting.

Keywords: public-key encryption, Fujisaki-Okamoto transformation, tight reductions, quantum random oracle model

Contents

1 Introduction

The notion of INDistinguishability against Chosen-Ciphertext Attacks (IND-CCA) [RS92] is now widely accepted as the standard security notion for asymmetric encryption schemes. Intuitively, IND-CCA security requires that no efficient adversary can recognize which of two messages is encrypted in a given ciphertext, even if the two candidate messages are chosen by the adversary himself. In contrast to the similar but weaker notion of INDistinguishability against Chosen-Plaintext Attacks (IND-CPA), an IND-CCA adversary is given access to a decryption oracle throughout the attack.

Generic Transformations achieving IND-CCA Security. While IND-CCA security is in many applications the desired notion of security, it is usually much more difficult to prove than IND-CPA security. Thus, several transformations have been suggested that turn a public-key encryption (PKE) scheme with weaker security properties into an IND-CCA one generically. For instance, in a seminal paper, Fujisaki and Okamoto [FO99, FO13] proposed a generic transformation (FO transformation) combining any One-Way (OW-CPA) secure asymmetric encryption scheme with any one-time secure symmetric encryption scheme into a Hybrid encryption scheme that is (IND-CCA) secure in the random oracle model [BR93]. Subsequently, Okamoto and Pointcheval [OP01] and Coron et al. [CHJ+02] proposed two more generic transformations (called REACT and GEM) that are considerably simpler but require the underlying asymmetric scheme to be One-Way against Plaintext Checking Attacks (OW-PCA). OW-PCA security is a non-standard security notion that provides the adversary with a plaintext checking oracle $\textsc{Pco}(c, m)$ that returns 1 iff decryption of ciphertext c yields message m. A similar transformation was also implicitly used in the "Hashed ElGamal" encryption scheme by Abdalla et al. [ABR01].

KEMs. In his "A Designer's Guide to KEMs" paper, Dent [Den03] provides "more modern" versions of the FO [Den03, Table 5] and the REACT/GEM [Den03, Table 2] transformations that result in

IND-CCA secure key-encapsulation mechanisms (KEMs). Recall that any IND-CCA secure KEM can be combined with any (one-time) chosen-ciphertext secure symmetric encryption scheme to obtain a IND-CCA secure PKE scheme [CS03]. Due to their efficiency and versatility, in practice one often works with such hybrid encryption schemes derived from a KEM. For that reason the primary goal of our paper will be constructing IND-CCA secure KEMs.

We remark that all previous variants of the FO transformation require the underlying PKE scheme to be γ-spread [FO99], which essentially means that ciphertexts (generated by the probabilistic encryption algorithm) have sufficiently large entropy.

Security against Quantum Adversaries. Recently, the above mentioned generic transformations have gathered renewed interest in the quest of finding an IND-CCA secure asymmetric encryption scheme that is secure against quantum adversaries, i.e., adversaries equipped with a quantum computer. In particular, the NIST announced a competition with the goal to standardize new asymmetric encryption systems [NIS17] with security against quantum adversaries. Natural candidates base their IND-CPA security on the hardness of certain problems over lattices and codes, which are generally believed to resists quantum adversaries. Furthermore, quantum computers may execute all "offline primitives" such as hash functions on arbitrary superpositions, which motivated the introduction of the quantum (accessible) random oracle model [BDF+11]. Targhi and Unruh recently proved a variant of the FO transformation secure in the quantum random oracle model [TU16]. Helping to find IND-CCA secure KEM with provable (post-quantum) security will thus be an important goal in this paper.

Discussion. Despite their versatility, the above FO and REACT/GEM transformations have a couple of small but important disadvantages.

- **Tightness.** The security reduction of the FO transformation [FO99, FO13] in the random oracle model is not tight, i.e., it loses a factor of q_{G}, the number of random oracle queries. A non-tight security proof requires to adapt the system parameters accordingly, which results in considerably less efficient schemes. The REACT/GEM transformations have a tight security reduction, but they require the underlying encryption scheme to be OW-PCA secure. As observed by Peikert [Pei14], due to their decision/search equivalence, many natural lattice-based encryption scheme are not OW-PCA secure and it is not clear how to modify them to be so. In fact, the main technical difficulty is to build an IND-CPA or OW-PCA secure encryption scheme from an OW-CPA secure one, with a tight security reduction.

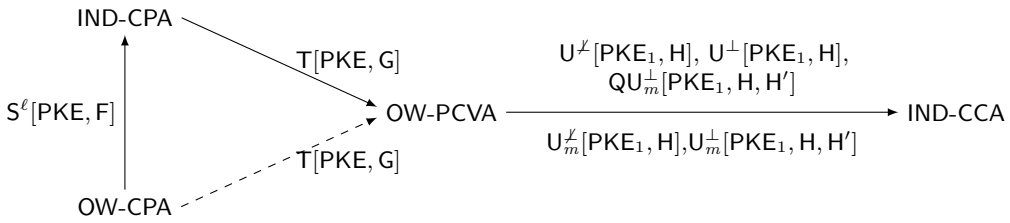
- **Correctness error.** The FO, as well as the REACT/GEM transformation require the underlying asymmetric encryption scheme to be perfectly correct, i.e., not having a decryption error. In general, one cannot exclude the fact that even a (negligibly) small decryption error could be exploited by a concrete IND-CCA attack against FO-like transformed schemes. Dealing with imperfectly correct schemes is of great importance since many (but not all) practical lattice-based encryption schemes have a small correctness error, see, e.g., DXL [DXL12], Peikert [Pei14], BCNS [BCNS15], New Hope [ADPS16], Frodo [BCD+16], Lizard [CKLS16], and Kyber [BDK+17].[1]

These deficiencies were of little or no concern when the FO and REACT/GEM transformations were originally devised. Due to the emergence of large-scale scenarios (which benefit heavily from tight security reductions) and the increased popularity of lattice-based schemes with correctness defects, however, we view these deficiencies as acute problems.

1.1 Our contributions

Our main contribution is a modular treatment of FO-like transformations. That is, we provide fine-grained transformations that can be used to turn an OW-CPA secure PKE scheme into an IND-CCA secure one in several steps. For instance, we provide separate OW-CPA $\rightarrow$ OW-PCA and OW-PCA $\rightarrow$ IND-CCA transformations that, taken together, yield the original FO transformation. However, we also provide variants of these individual transformations that achieve different security goals and tightness properties.

[1] Lattice-based encryption schemes can be made perfectly correctness by putting a limit on the noise and setting the modulus of the LWE instance large enough, see e.g. [BCLvV16, HGSW05]. But increasing the size of the modulus makes the LWE problem easier to solve in practice, and thus the dimension of the problem needs to be increased in order to obtain the same security levels. Larger dimension and modulus increase the public-key and ciphertext length.



Transformation	Security implication	QROM?	ROM Tightness?	Requirements
$\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ (§3.1)	OW-CPA ⇒ OW-PCA	✓	—	none
$\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ (§3.1)	IND-CPA ⇒ OW-PCA	✓	✓	none
$\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ (§3.1)	OW-CPA ⇒ OW-PCVA	✓	—	γ-spread
$\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ (§3.1)	IND-CPA ⇒ OW-PCVA	—	✓	γ-spread
$\mathsf{KEM}^{\not\perp} = \mathsf{U}^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}]$ (§3.2.2)	OW-PCA ⇒ IND-CCA	—	✓	none
$\mathsf{KEM}^{\perp} = \mathsf{U}^{\perp}[\mathsf{PKE}_1, \mathsf{H}]$ (§3.2.1)	OW-PCVA ⇒ IND-CCA	—	✓	none
$\mathsf{KEM}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$ (§3.2.3)	OW-CPA ⇒ IND-CCA	—	✓	det. PKE_1
$\mathsf{KEM}^{\perp}_m = \mathsf{U}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$ (§3.2.3)	OW-VA ⇒ IND-CCA	—	✓	det. PKE_1
$\mathsf{QKEM}^{\perp}_m = \mathsf{QU}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$ (§4.3.1)	OW-PCA ⇒ IND-CCA	✓	✓	none
$\mathsf{PKE}_\ell = \mathsf{S}^{\ell}[\mathsf{PKE}, \mathsf{F}]$ (§3.4)	OW-CPA ⇒ IND-PCA	—	✓	none

Figure 1: Our modular transformations. Top: solid errors indicate tight reductions, dashed arrows indicate non-tight reductions. Bottom: properties of the transformations. The tightness row only refers to tightness in the standard random oracle model; all our reduction in the quantum random oracle model are non-tight.

All of our individual transformations are robust against PKE schemes with correctness errors (in the sense that the correctness error of the resulting schemes can be bounded by the correctness error of the original scheme).

The benefit of our modular treatment is not only a conceptual simplification, but also a larger variety of possible combined transformations (with different requirements and properties). For instance, combining two results about our transformations T and $\mathsf{U}^{\not\perp}$, we can show that the original FO transformation yields IND-CCA security from IND-CPA security with a *tight* security reduction. Combining S^{ℓ} with T and $\mathsf{U}^{\not\perp}$, on the other hand, yields tight IND-CCA security from the weaker notion of OW-CPA security, at the expense of a larger ciphertext. (See Figure 1 for an overview.)

1.1.1 Our transformations in detail

In the following, we give a more detailed overview over our transformations. We remark that all our transformations require a PKE scheme (and not a KEM). We view it as an interesting open problem to construct similar transformations that only assume (and yield) KEMs, since such transformations have the potential of additional efficiency gains.

T: from OW-CPA to OW-PCA Security ("Derandomization"+"re-encryption"). Starting from an encryption scheme PKE and a hash function G, we build a deterministic encryption scheme $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ by defining

$$\mathsf{Enc}_1(pk, m) := \mathsf{Enc}(pk, m; \mathsf{G}(m)),$$

where $\mathsf{G}(m)$ is used as the random coins for Enc. Note that Enc_1 is deterministic. $\mathsf{Dec}_1(sk, c)$ first decrypts c into m' and rejects if $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$ ("re-encryption"). Modeling G as a random oracle, OW-PCA security of PKE_1 non-tightly reduces to OW-CPA security of PKE and tightly reduces to IND-CPA security of PKE. If PKE furthermore is γ-spread (for sufficiently large γ), then PKE_1 is even OW-PCVA secure. OW-PCVA security[2] is PCA security, where the adversary is additionally given access to a validity oracle

[2] OW-PCVA security is called $\mathsf{OW\text{-}CPA}^{+}$ security with access to a Pco oracle in [Den03].

$\textsc{Cvo}(c)$ that checks c's validity (in the sense that it does not decrypt to $\perp$, see also Definition 2.1).

$\mathsf{U}^{\not\perp}$ ($\mathsf{U}^{\perp}$): FROM OW-PCA (OW-PCVA) TO IND-CCA SECURITY ("HASHING"). Starting from an encryption scheme PKE_1 and a hash function H, we build a key encapsulation mechanism $\mathsf{KEM}^{\not\perp} = \mathsf{U}^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}]$ with "implicit rejection" by defining

$$\mathsf{Encaps}(pk) := (c \leftarrow \mathsf{Enc}_1(pk, m), K := \mathsf{H}(c, m)), \tag{1}$$

where m is picked at random from the message space.

$$\mathsf{Decaps}^{\not\perp}(sk, c) = \begin{cases} \mathsf{H}(c, m) & m \neq \perp \\ \mathsf{H}(c, s) & m = \perp \end{cases}, \tag{2}$$

where $m := \mathsf{Dec}(sk, c)$ and s is a random seed which is contained in sk. Modeling H as a random oracle, IND-CCA security of $\mathsf{KEM}^{\not\perp}$ tightly reduces to OW-PCA security of PKE_1.

We also define $\mathsf{KEM}^{\perp} = \mathsf{U}^{\perp}[\mathsf{PKE}_1, \mathsf{H}]$ with "explicit rejection" which differs from $\mathsf{KEM}^{\not\perp}$ only in decapsulation:

$$\mathsf{Decaps}^{\perp}(sk, c) = \begin{cases} \mathsf{H}(c, m) & m \neq \perp \\ \perp & m = \perp \end{cases}, \tag{3}$$

where $m := \mathsf{Dec}(sk, c)$. Modeling H as a random oracle, IND-CCA of $\mathsf{KEM}^{\perp}$ security tightly reduces to OW-PCVA security of PKE_1. We remark that transformation $\mathsf{U}^{\perp}$ is essentially [Den03, Table 2], i.e., a KEM variant of the REACT/GEM transformations.

$\mathsf{U}^{\not\perp}_m$ ($\mathsf{U}^{\perp}_m$): FROM DETERMINISTIC OW-CPA (OW-VA) TO IND-CCA SECURITY ("HASHING"). We consider two more variants of $\mathsf{U}^{\not\perp}$ and $\mathsf{U}^{\perp}$, namely $\mathsf{U}^{\not\perp}_m$ and $\mathsf{U}^{\perp}_m$. Transformation $\mathsf{U}^{\not\perp}_m$ ($\mathsf{U}^{\perp}_m$) is a variant of $\mathsf{U}^{\not\perp}$ ($\mathsf{U}^{\perp}$), where $K = \mathsf{H}(c, m)$ from Equations (1)-(3) is replaced by $K = \mathsf{H}(m)$. We prove that IND-CCA security of $\mathsf{KEM}^{\not\perp}_m := \mathsf{U}^{\not\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$ ($\mathsf{KEM}^{\perp}_m := \mathsf{U}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$) in the random oracle model tightly reduces to IND-CPA (IND-VA[3]) security of PKE_1, if encryption of PKE_1 is deterministic.

$\mathsf{QU}^{\perp}_m$: FROM OW-PCA TO IND-CCA SECURITY IN THE QUANTUM ROM. We first prove that transformation T also works in the quantum random oracle model. Next, to go from OW-PCA to IND-CCA in the QROM, we build a key encapsulation mechanism $\mathsf{QKEM}^{\perp}_m = \mathsf{QU}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$ with explicit rejection by defining

$$\mathsf{QEncaps}_m(pk) := ((c \leftarrow \mathsf{Enc}_1(pk, m), d := \mathsf{H}'(m)), K := \mathsf{H}(m)),$$

where m is picked at random from the message space.

$$\mathsf{QDecaps}^{\perp}_m(sk, c, d) = \begin{cases} \mathsf{H}(m') & m' \neq \perp \\ \perp & m' = \perp \vee \mathsf{H}'(m') \neq d \end{cases},$$

where $m' := \mathsf{Dec}(sk, c)$. $\mathsf{QU}^{\perp}_m$ differs from $\mathsf{U}^{\not\perp}$ only in the additional hash value $d = \mathsf{H}'(m)$ from the ciphertext and H' is a random oracle with matching domain and image. This trick was introduced in [Unr15] and used in [TU16] in the context of the FO transformation. Modeling H and H' as a quantum random oracles, IND-CCA security of KEM reduces to OW-PCA security of PKE_1.

1.1.2 The resulting FO transformations

Our final transformations $\mathsf{FO}^{\not\perp}$ ("FO with implicit rejection"), $\mathsf{FO}^{\perp}$ ("FO with explicit rejection"), $\mathsf{FO}^{\not\perp}_m$ ("FO with implicit rejection, $K = \mathsf{H}(m)$"), $\mathsf{FO}^{\perp}_m$ ("FO with explicit rejection, $K = \mathsf{H}(m)$"), and $\mathsf{QFO}^{\perp}_m$ ("Quantum FO with explicit rejection, $K = \mathsf{H}(m)$") are defined in the following table.

Transformation	**QROM?**	**ROM Tightness?**	**Requirements**
$\mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$	—	✓	none
$\mathsf{FO}^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$	—	✓	γ-spread
$\mathsf{FO}^{\not\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\not\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$	—	✓	none
$\mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$	—	✓	γ-spread
$\mathsf{QFO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}'] := \mathsf{QU}^{\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}, \mathsf{H}']$	✓	✓	none

[3] OW-VA security is OW-CPA security, where the adversary is given access to a validity oracle $\textsc{Cvo}(c)$ that checks c's validity (cf. Definition 2.1).

As corollaries of our modular transformation we obtain that IND-CCA security of $\mathsf{FO}^{\not\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, $\mathsf{FO}^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, $\mathsf{FO}^{\not\perp}_m[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, and $\mathsf{FO}^{\perp}_m[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ non-tightly reduces to the OW-CPA security of PKE, and tightly reduces to the IND-CPA security of PKE, in the random oracle model. We remark that transformation $\mathsf{FO}^{\perp}_m$ essentially recovers a KEM variant [Den03, Table 5] of the original FO transformation [FO99]. Whereas the explicit rejection variants $\mathsf{FO}^{\perp}$ and $\mathsf{FO}^{\perp}_m$ require PKE to be γ-spread, there is no such requirement on $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\not\perp}_m$. Further, IND-CCA security of $\mathsf{QFO}^{\perp}_m[\mathsf{PKE},\mathsf{G},\mathsf{H},\mathsf{H}']$ reduces to the OW-CPA security of PKE, in the quantum random oracle model. Our transformation $\mathsf{QFO}^{\perp}_m$ essentially recovers a KEM variant of the modified FO transformation by Targhi and Unruh [TU16]. As it is common in the quantum random oracle model, all our reductions are (highly) non-tight. We leave it as an open problem to derive a tighter security reduction of T, for example to IND-CPA security of PKE.

Correctness Error. We stress that all our security reductions also take non-zero correctness error into account. Finding the "right" definition of correctness that is achievable (say, by currently proposed lattice-based encryption schemes) and at the same time sufficient to prove security turned out to be a bit subtle. This is the reason why our definition of correctness (see Section 2.1) derives from the ones previously given in the literature (e.g. [DNR04, BV17]). The concrete bounds of $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}^{\not\perp}_m$, and $\mathsf{FO}^{\perp}_m$ give guidance on the required correctness error of the underlying PKE scheme. Concretely, for "κ bits security", PKE requires a correctness error of $2^{-\kappa}$.

1.1.3 Example Instantiations

In the context of ElGamal encryption one can apply $\{\mathsf{FO}^{\not\perp},\mathsf{FO}^{\perp},\mathsf{FO}^{\not\perp}_m,\mathsf{FO}^{\perp}_m\}$ to obtain the schemes of [KML03, BLK00, GMMV05] whose IND-CCA security non-tightly reduces to the CDH assumption, and tightly reduces to the DDH assumption. Alternatively, one can directly use $\mathsf{U}^{\not\perp}/\mathsf{U}^{\perp}$ to obtain the more efficient schemes of [OP01, CHJ+02, ABR01, Sho04a] whose IND-CCA security tightly reduces to the gap-DH (a.k.a. strong CDH) assumption. In the context of deterministic encryption schemes such as RSA, Paillier, etc, one can apply $\mathsf{U}^{\not\perp}/\mathsf{U}^{\perp}$ to obtain schemes mentioned in [Sho04a, Den03] whose IND-CCA security tightly reduces to one-way security. Finally, in the context of lattices-based encryption (e.g., [Reg05, LPR13]), one can apply $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\perp}$, $\mathsf{FO}^{\not\perp}_m$, $\mathsf{FO}^{\perp}_m$, and $\mathsf{QFO}^{\perp}_m$ to achieve IND-CCA security.

1.1.4 Transformation S^{ℓ}: from OW-CPA to IND-CPA, Tightly

Note that T requires PKE to be IND-CPA secure to achieve a tight reduction. In case one has to rely on OW-CPA security, transformation S^{ℓ} offers the following tradeoff between efficiency and tightness. It transforms an OW-CPA secure PKE into an IND-CPA secure PKE_{ℓ}, where ℓ is a parameter. The ciphertext consists of ℓ independent PKE ciphertexts:

$$\mathsf{Enc}_{\ell}(pk,m) := (\mathsf{Enc}(pk,x_1),\ldots,\mathsf{Enc}(pk,x_{\ell}),m\oplus\mathsf{G}(x_1,\ldots,x_{\ell})).$$

The reduction (to the OW-CPA security of PKE) loses a factor of $q_{\mathsf{G}}^{1/\ell}$, where q_{G} is the number of G-queries an adversary makes.

Observe that the only way to gather information about m is to explicitly query $\mathsf{G}(x_1,\ldots,x_n)$, which requires to find all x_i. The reduction can use this observation to embed an OW-CPA challenge as one $\mathsf{Enc}(pk,x_{i^*})$ and hope to learn x_{i^*} from the G-queries of a successful IND-CPA adversary. In this, the reduction will know all x_i except x_{i^*}. The difficulty in this reduction is to identify the "right" G-query (that reveals x_{i^*}) in all of the adversary's G-queries. Intuitively, the more instances we have, the easier it is for the reduction to spot the G-query $(x_1,\ldots,x_{\ell})$ (by comparing the x_i for $i\neq i^*$), and the less guessing is necessary. Hence, we get a tradeoff between the number of instances ℓ (and thus the size of the ciphertext) and the loss of the reduction.

1.2 Related work

As already pointed out, $\mathsf{FO}^{\perp}_m = \mathsf{U}^{\perp}_m \circ \mathsf{T}$ is essentially a KEM variant of the Fujisaki-Okamoto transform from [Den03, Table 5]. Further, $\mathsf{U}^{\perp}$ is a KEM variant [Den03] of the GEM/REACT transform [OP01, CHJ+02, ABR01]. Our modular view suggest that the FO transform implicitly contains the GEM/REACT transform, at least the proof technique. With this more general view, the FO transform and its variants remains the only known transformation from CPA to CCA security. It is an interesting open problem

to come up with alternative transformations that get rid of derandomization or that dispense with re-encryption (which preserving efficiency). Note that for the ElGamal encryption scheme, the "twinning" technique [CKS08, CKS09] does exactly this, but it uses non-generic zero-knowledge proofs that are currently not available for all schemes (e.g., for lattice-based schemes).

In concurrent and independent work, [AOP+17] considers the IND-CCA security of LIMA which in our notation can be described as $\mathsf{FO}_m^{\perp}[\mathsf{RLWE}, \mathsf{G}, \mathsf{H}]$. Here RLWE is a specific encryption scheme based on lattices associated to polynomial rings from [LPR10], which is IND-CPA secure under the Ring-LWE assumption. As the main result, [AOP+17] provides a tight reduction of LIMA's IND-CCA security to the Ring-LWE assumption, in the random oracle model. The proof exploits "some weakly homomorphic properties enjoyed by the underlying encryption scheme" and therefore does not seem to be applicable to other schemes. The tight security reduction from Ring-LWE is recovered as a special case of our general security results on $\mathsf{FO}_m^{\perp}$. We note that the security reduction of [AOP+17] does not take the (non-zero) correctness error of RLWE into account.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. For a set S, $|S|$ denotes the cardinality of S. For a finite set S, we denote the sampling of a uniform random element x by $x \xleftarrow{\$} S$, while we denote the sampling according to some distribution $\mathfrak{D}$ by $x \leftarrow \mathfrak{D}$. For a polynomial $p(X)$ with integer coefficients, we denote by $\mathsf{Roots}(p)$ the (finite) set of (complex) roots of p. By $[\![B]\!]$ we denote the bit that is 1 if the Boolean Statement B is true, and otherwise 0.

Algorithms. We denote deterministic computation of an algorithm A on input x by $y := A(x)$. We denote algorithms with access to an oracle O by A^{O}. Unless stated otherwise, we assume all our algorithms to be probabilistic and denote the computation by $y \leftarrow A(x)$.

Random Oracles. We will at times model hash functions $\mathsf{H} : \mathfrak{D}_{\mathsf{H}} \to \mathfrak{R}(\mathsf{H})$ as random oracles. To keep record of the queries issued to H, we will use a hash list $\mathfrak{L}_{\mathsf{H}}$ that contains all tuples $(x, \mathsf{H}(x))$ of arguments $x \in \mathfrak{D}_{\mathsf{H}}$ that H was queried on and the respective answers $\mathsf{H}(x)$. We make the convention that $\mathsf{H}(x) = \perp$ for all $x \notin \mathfrak{D}_{\mathsf{H}}$.

Games. Following [Sho04b, BR06], we use code-based games. We implicitly assume boolean flags to be initialized to false, numerical types to 0, sets to $\varnothing$, and strings to the empty string ϵ. We make the convention that a procedure terminates once it has returned an output.

2.1 Public-Key Encryption

Syntax. A public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three algorithms and a finite message space $\mathcal{M}$ (which we assume to be efficiently recognizable). The key generation algorithm Gen outputs a key pair (pk, sk), where pk also defines a randomness space $\mathcal{R} = \mathcal{R}(pk)$. The encryption algorithm Enc, on input pk and a message $m \in \mathcal{M}$, outputs an encryption $c \leftarrow \mathsf{Enc}(pk, m)$ of m under the public key pk. If necessary, we make the used randomness of encryption explicit by writing $c := \mathsf{Enc}(pk, m; r)$, where $r \xleftarrow{\$} \mathcal{R}$ and $\mathcal{R}$ is the randomness space. The decryption algorithm Dec, on input sk and a ciphertext c, outputs either a message $m = \mathsf{Dec}(sk, c) \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$ to indicate that c is not a valid ciphertext.

Correctness. We call a public-key encryption scheme PKE δ-*correct* if

$$\mathbf{E}[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(sk, c) \neq m \mid c \leftarrow \mathsf{Enc}(pk, m)]] \leq \delta,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}$. Equivalently, δ-correctness means that for all (possibly unbounded) adversaries A, $\Pr[\mathsf{COR}^{\mathsf{A}}_{\mathsf{PKE}} \Rightarrow 1] \leq \delta$, where the correctness game COR is defined as in Figure 2 (left). That is, an (unbounded) adversary obtains the public and the secret key and wins if it finds a message inducing a correctness error. Note that our definition of correctness slightly derives from previous definitions (e.g. [DNR04, BV17]) but it has been carefully crafted such that it is sufficient to prove our main theorems (i.e., the security of the Fujisaki-Okamoto transformation) and at the same time it is fulfilled by all recently proposed lattice-based encryption schemes with correctness error.

If $\mathsf{PKE} = \mathsf{PKE}^{\mathsf{G}}$ is defined relative to a random oracle G, then defining correctness is a bit more subtle as the correctness bound might depend on the number of queries to G.[4] We call a public-key encryption scheme PKE in the random oracle model $\delta(q_{\mathsf{G}})$-*correct* if for all (possibly unbounded) adversaries A making at most q_{G} queries to random oracle G, $\Pr[\mathsf{COR\text{-}RO}^{\mathsf{A}}_{\mathsf{PKE}} \Rightarrow 1] \leq \delta(q_{\mathsf{G}})$, where the correctness game $\mathsf{COR\text{-}RO}$ is defined as in Figure 2 (right). If PKE is defined relative to two random oracles G, H, then the correctness error δ is a function in q_{G} and q_{H}.

Note that our correctness definition in the standard model is a special case of the one in the random oracle model, where the number of random oracle queries is zero and hence $\delta(q_{\mathsf{G}})$ is a constant.

GAME COR:	**GAME** COR-RO:
01 $(pk, sk) \leftarrow \mathsf{Gen}$	05 $(pk, sk) \leftarrow \mathsf{Gen}$
02 $m \leftarrow \mathsf{A}(sk, pk)$	06 $m \leftarrow \mathsf{A}^{\mathsf{G}(\cdot)}(sk, pk)$
03 $c \leftarrow \mathsf{Enc}(pk, m)$	07 $c \leftarrow \mathsf{Enc}(pk, m)$
04 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$	08 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$

Figure 2: Correctness game COR for PKE in the standard model (left) and $\mathsf{COR\text{-}RO}$ for PKE defined relative to a random oracle G (right).

Min-Entropy. [FO13] For $(pk, sk) \leftarrow \mathsf{Gen}$ and $m \in \mathcal{M}$, we define the *min-entropy* of $\mathsf{Enc}(pk, m)$ by $\gamma(pk, m) := -\log \max_{c \in \mathcal{C}} \Pr_{r \leftarrow \mathcal{R}}[c = \mathsf{Enc}(pk, m; r)]$. We say that PKE is γ-*spread* if, for every key pair $(pk, sk) \leftarrow \mathsf{Gen}$ and every message $m \in \mathcal{M}$, $\gamma(pk, m) \geq \gamma$. In particular, this implies that for every possible ciphertext $c \in \mathcal{C}$, $\Pr_{r \leftarrow \mathcal{R}}[c = \mathsf{Enc}(pk, m; r)] \leq 2^{-\gamma}$.

Security. We now define three security notions for public-key encryption: One-Wayness under Chosen Plaintext Attacks ($\mathsf{OW\text{-}CPA}$), One-Wayness under Plaintext Checking Attacks ($\mathsf{OW\text{-}PCA}$) and One-Wayness under Plaintext and Validity Checking Attacks ($\mathsf{OW\text{-}PCVA}$).

Definition 2.1 ($\mathsf{OW\text{-}ATK}$). Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$. For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{PCA}, \mathsf{VA}, \mathsf{PCVA}\}$, we define $\mathsf{OW\text{-}ATK}$ games as in Figure 3, where

$$\mathrm{O}_{\mathsf{ATK}} := \begin{cases} - & \mathsf{ATK} = \mathsf{CPA} \\ \textsc{Pco}(\cdot,\cdot) & \mathsf{ATK} = \mathsf{PCA} \\ \textsc{Cvo}(\cdot) & \mathsf{ATK} = \mathsf{VA} \\ \textsc{Pco}(\cdot,\cdot), \textsc{Cvo}(\cdot) & \mathsf{ATK} = \mathsf{PCVA} \end{cases}.$$

We define the $\mathsf{OW\text{-}ATK}$ *advantage function of an adversary* A *against* PKE as $\mathrm{Adv}^{\mathsf{OW\text{-}ATK}}_{\mathsf{PKE}}(\mathsf{A}) := \Pr[\mathsf{OW\text{-}ATK}^{\mathsf{A}}_{\mathsf{PKE}} \Rightarrow 1]$.

A few remarks are in place. Our definition of the plaintext checking oracle $\textsc{Pco}(m, c)$ (c.f. Figure 3) implicitly disallows queries on messages $m \in \mathcal{M}$. (With the convention that $\textsc{Pco}(m \notin \mathcal{M}, c)$ yields $\perp$.) This restriction is important since otherwise the ciphertext validity oracle $\textsc{Cvo}(\cdot)$ could be simulated as $\textsc{Cvo}(m) = \textsc{Pco}(\perp, c)$. Similarly, the ciphertext validity oracle $\textsc{Cvo}(c)$ implicitly disallows queries on the challenge ciphertext c^*.

[4] For an example why the number of random oracle queries matters in the context of correctness, we refer to Theorem 3.1.

GAME OW-ATK:	$\textsc{Pco}(m \in \mathcal{M}, c)$
09 $(pk, sk) \leftarrow \mathsf{Gen}$	14 **return** $[\![\mathsf{Dec}(sk, c) = m]\!]$
10 $m^* \xleftarrow{\$} \mathcal{M}$	
11 $c^* \leftarrow \mathsf{Enc}(pk, m^*)$	$\textsc{Cvo}(c \neq c^*)$
12 $m' \leftarrow \mathsf{A}^{\mathrm{O}_{\mathsf{ATK}}}(pk, c)$	15 $m := \mathsf{Dec}(sk, c)$
13 **return** $\textsc{Pco}(m', c^*)$	16 **return** $[\![m \in \mathcal{M}]\!]$

Figure 3: Games $\mathsf{OW\text{-}ATK}$ ($\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{PCA}, \mathsf{VA}, \mathsf{PCVA}\}$) for PKE, where $\mathrm{O}_{\mathsf{ATK}}$ is defined in Definition 2.1. $\textsc{Pco}(\cdot,\cdot)$ is the Plaintext Checking Oracle and $\textsc{Cvo}(\cdot)$ is the Ciphertext Validity Oracle.

GAME IND-CPA	**GAME** IND-CCA	DECAPS($c \neq c^*$)
01 $(pk, sk) \leftarrow \mathsf{Gen}$	07 $(pk, sk) \leftarrow \mathsf{Gen}$	13 $K := \mathsf{Decaps}(sk, c)$
02 $b \xleftarrow{\$} \{0,1\}$	08 $b \xleftarrow{\$} \{0,1\}$	14 **return** K
03 $(m_0^*, m_1^*, st) \leftarrow \mathsf{A}_1(pk)$	09 $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
04 $c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$	10 $K_1^* \xleftarrow{\$} \mathcal{K}$	
05 $b' \leftarrow \mathsf{A}_2(pk, c^*, st)$	11 $b' \leftarrow \mathsf{A}^{\text{DECAPS}}(c^*, K_b^*)$	
06 **return** $[\![b' = b]\!]$	12 **return** $[\![b' = b]\!]$	

Figure 4: Games IND-CPA for PKE and IND-CCA game for KEM.

Usually, the adversary wins the one-way game iff its output m' equals the challenge message m^*. Instead, in game OW-ATK the correctness of m' is checked using the PCO oracle, i.e., it returns 1 iff $\mathsf{Dec}(sk, c^*) = m'$. The two games have statistical difference δ, if PKE is δ-correct.

Additionally, we define Indistinguishability under Chosen Plaintext Attacks (IND-CPA).

Definition 2.2 (IND-CPA). Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$. We define the IND-CPA game as in Figure 4, and the IND-CPA advantage function of an adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ against PKE (such that A_2 has binary output) as $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}) := |\Pr[\mathsf{IND\text{-}CPA}^{\mathsf{A}} \Rightarrow 1] - 1/2|$.

We also define OW-ATK and IND-CPA security in the random oracle model model, where PKE and adversary A are given access to a random oracle H. We make the convention that the number q_{H} of the adversary's random oracle queries count the total number of times H is executed in the experiment. That is, the number of A explicit queries to $\mathsf{H}(\cdot)$ plus the number of implicit queries to $\mathsf{H}(\cdot)$ made by the experiment.

It is well known that IND-CPA security of PKE with sufficiently large message space implies its OW-CPA security.

Lemma 2.3 *For any adversary* B *there exists an adversary* A *with the same running time as that of* B *such that* $\mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}}(\mathsf{B}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}) + 1/|\mathcal{M}|$.

2.2 Key Encapsulation

SYNTAX. A key encapsulation mechanism $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of three algorithms. The key generation algorithm Gen outputs a key pair (pk, sk), where pk also defines a finite key space $\mathcal{K}$. The encapsulation algorithm Encaps, on input pk, outputs a tuple (K, c) where c is said to be an encapsulation of the key K which is contained in key space $\mathcal{K}$. The deterministic decapsulation algorithm Decaps, on input sk and an encapsulation c, outputs either a key $K := \mathsf{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation. We call KEM *δ-correct* if

$$\Pr\left[\mathsf{Decaps}(sk, c) \neq K \mid (pk, sk) \leftarrow \mathsf{Gen}; (K, c) \leftarrow \mathsf{Encaps}(pk)\right] \leq \delta \ .$$

Note that the above definition also makes sense in the random oracle model since KEM ciphertexts do not depend on messages.

SECURITY. We now define a security notion for key encapsulation: Indistinguishbility under Chosen Ciphertext Attacks (IND-CCA).

Definition 2.4 (IND-CCA). We define the IND-CCA game as in Figure 4 and the IND-CCA *advantage function of an adversary* A *(with binary output) against* KEM as $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) := |\Pr[\mathsf{IND\text{-}CCA}^{\mathsf{A}} \Rightarrow 1] - 1/2| \ .$

3 Modular FO Transformations

In Section 3.1, we will introduce T that transforms any OW-CPA secure encryption scheme PKE into a OW-PCA secure encryption scheme PKE_1. If PKE is furthermore IND-CPA, then the reduction is tight. Furthermore, if PKE is λ-spread, then PKE_1 even satisfied the stronger security notion of OW-PCVA

security. Next, in Section 3.2.2 (Section 3.2.1), we will introduce transformations $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\not\perp}_m$ ($\mathsf{U}^{\perp}$, $\mathsf{U}^{\perp}_m$) that transform any OW-PCA (OW-PCVA) secure encryption scheme PKE_1 into an IND-CCA secure KEM. The security reduction is tight. Transformations $\mathsf{U}^{\not\perp}_m$ and $\mathsf{U}^{\perp}_m$ can only be applied for deterministic encryption schemes. Combining T with $\{\mathsf{U}^{\not\perp}, \mathsf{U}^{\not\perp}_m, \mathsf{U}^{\perp}, \mathsf{U}^{\perp}_m\}$, in Section 3.3 we provide concrete bounds for the IND-CCA security of the resulting KEMs. Finally, in Section 3.4 we introduce S^{ℓ} that transforms any OW-CPA secure scheme into an IND-CPA secure one, offering a tradeoff between tightness and ciphertext size.

3.1 Transformation T: from OW-CPA/IND-CPA to OW-PCVA

T transforms an OW-CPA secure public-key encryption scheme into an OW-PCA secure one.

The Construction. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, and random oracle $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, we associate $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$. The algorithms of $\mathsf{PKE}_1 = (\mathsf{Gen}, \mathsf{Enc}_1, \mathsf{Dec}_1)$ are defined in Figure 5. Note that Enc_1 deterministically computers the ciphertext as $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$.

$\mathsf{Enc}_1(pk, m)$	$\mathsf{Dec}_1(sk, c)$
01 $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	03 $m' := \mathsf{Dec}(sk, c)$.
02 **return** c	04 **if** $m' = \perp$ **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
	05 **return** $\perp$
	06 **else return** m'

Figure 5: OW-PCVA-secure encryption scheme $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ with deterministic encryption.

Non-tight Security from OW-CPA. The following theorem establishes that OW-PCVA security of PKE_1 (cf. Definition 2.1) non-tightly reduces to the OW-CPA security of PKE, in the random oracle model, given that PKE is γ-spread (for sufficiently large γ). If PKE is not γ-spread, then PKE_1 is still OW-PCA secure.

Theorem 3.1 (PKE OW-CPA $\overset{\text{ROM}}{\Rightarrow}$ PKE_1 OW-PCVA)**.** *If* PKE *is δ-correct, then* PKE_1 *is δ_1-correct in the random oracle model with $\delta_1(q_\mathsf{G}) = q_\mathsf{G} \cdot \delta$. Assume* PKE *to be γ-spread. Then, for any* OW-PCVA *adversary* B *that issues at most q_G queries to the random oracle* G*, q_P queries to a plaintext checking oracle* Pco*, and q_V queries to a validity checking oracle* Cvo*, there exists an* OW-CPA *adversary* A *such that*

$$\mathrm{Adv}^{\mathsf{OW\text{-}PCVA}}_{\mathsf{PKE}_1}(\mathsf{B}) \leq q_\mathsf{G} \cdot \delta + q_V \cdot 2^{-\gamma} + (q_\mathsf{G} + 1) \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$$

and the running time of A *is about that of* B*.*

The main idea of the proof is that since Enc_1 is deterministic, the $\mathsf{PCA}(\cdot,\cdot)$ oracle can be equivalently implemented by "re-encryption" and the $\textsc{Cvo}(\cdot)$ oracle by controlling the random oracles. Additional care has to be taken to account for the correctness error.

Proof. To prove correctness, consider an adversary A playing the correctness game COR-RO (Figure 2) of PKE_1 in the random oracle model. Game COR-RO makes at most q_G (distinct) queries $\mathsf{G}(m_1), \ldots, \mathsf{G}(m_{q_\mathsf{G}})$ to G. We call such a query $\mathsf{G}(m_i)$ *problematic* iff it exhibits a correctness error in PKE_1 (in the sense that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m_i; \mathsf{G}(m_i))) \neq m_i$). Since G outputs independently random values, each $\mathsf{G}(m_i)$ is problematic with probability at most δ (averaged over (pk, sk)), since we assumed that PKE is δ-correct. Hence, a union bound shows that the probability that at least one $\mathsf{G}(m_i)$ is problematic is at most $q_\mathsf{G} \cdot \delta$. This proves $\Pr[\mathsf{COR\text{-}RO}^{\mathsf{A}} \Rightarrow 1] \leq q_\mathsf{G} \cdot \delta$ and hence PKE_1 is δ_1-correct with $\delta_1(q_\mathsf{G}) = q_\mathsf{G} \cdot \delta$.

To prove security, let B be an adversary against the OW-PCVA security of PKE_1, issuing at most q_G queries to G, at most q_P queries to Pco, and at most q_V queries to Cvo. Consider the sequence of games given in Figure 6.

Game G_0. This is the original OW-PCVA game. Random oracle queries are stored in set $\mathfrak{L}_G$ with the convention that $\mathsf{G}(m) = r$ iff $(m, r) \in \mathfrak{L}_G$. Hence,

$$\Pr[G^{\mathsf{B}}_0 \Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}PCVA}}_{\mathsf{PKE}_1}(\mathsf{B}) \ .$$

GAMES G_0-G_3

01 $(pk, sk) \leftarrow \mathsf{Gen}$
02 $m^* \xleftarrow{\$} \mathcal{M}$
03 $c^* \leftarrow \mathsf{Enc}(pk, m^*)$
04 $m' \leftarrow \mathsf{B}^{\mathsf{G}(\cdot),\textsc{Pco}(\cdot,\cdot),\textsc{Cvo}(\cdot)}(pk, c^*)$
05 **return** $[\![m' = m^*]\!]$

$\underline{\mathsf{G}(m)}$
06 **if** $\exists r$ s. th.$(m, r) \in \mathfrak{L}_G$
07 **return** r
08 **if** $m = m^*$ // G_3
09 QUERY := true // G_3
10 **abort** // G_3
11 $r \xleftarrow{\$} \mathcal{R}$
12 $\mathfrak{L}_G := \mathfrak{L}_G \cup \{(m, r)\}$
13 **return** r

$\underline{\textsc{Pco}(m \in \mathcal{M}, c)}$
14 $m' := \mathsf{Dec}(sk, c)$ // G_0-G_1
15 **return** $[\![m' = m]\!]$ **and** $[\![\mathsf{Enc}(pk, m'; \mathsf{G}(m')) = c]\!]$ // G_0-G_1
16 **return** $[\![\mathsf{Enc}(pk, m, \mathsf{G}(m)) = c]\!]$ // G_2-G_3

$\underline{\textsc{Cvo}(c \neq c^*)}$
17 $m' := \mathsf{Dec}(sk, c)$ // G_0-G_1
18 **return** $[\![m' \in \mathcal{M}]\!]$ **and** $[\![\mathsf{Enc}(pk, m'; G(m')) = c]\!]$ // G_0
19 **return** $[\![\exists (m, r) \in \mathfrak{L}_G$ **and** $\mathsf{Enc}(pk, m; r) = c$ **and** $m' = m]\!]$ // G_1
20 **return** $[\![\exists (m, r) \in \mathfrak{L}_G$ **and** $\mathsf{Enc}(pk, m; r) = c]\!]$ // G_2-G_3

Figure 6: Games G_0-G_3 for the proof of Theorem 3.1.

GAME G_1. In game G_1 the ciphertext validity oracle $\textsc{Cvo}(c \neq c^*)$ is replaced with one that first computes $m' = \mathsf{Dec}(sk, c)$ and returns 1 iff there exists a previous query (m, r) to G such that $\mathsf{Enc}(pk, m; r) = c$ and $m = m'$.

Consider a single query $\textsc{Cvo}(c)$ and define $m' := \mathsf{Dec}(sk, c)$. If $\textsc{Cvo}(c) = 1$ in G_1, then $\mathsf{G}(m') = \mathsf{G}(m) = r$ and hence $\mathsf{Enc}(pk, m'; G(m')) = c$, meaning $\textsc{Cvo}(c) = 1$ in G_0. If $\textsc{Cvo}(c) = 1$ in G_0, then we can only have $\textsc{Cvo}(c) = 0$ in G_1 only if $\mathsf{G}(m')$ was not queried before. This happens with probability $2^{-\gamma}$, where γ is the parameter from the γ-spreadness of PKE. By the union bound we obtain

$$|\Pr[G_1^{\mathsf{B}} \Rightarrow 1] - \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \leq q_V \cdot 2^{-\gamma}.$$

GAME G_2. In game G_2 we replace the plaintext checking oracle $\textsc{Pco}(m, c)$ and the ciphertext validity oracle $\textsc{Cvo}(c)$ by a simulation that does not check whether $m = m'$ anymore, where $m' = \mathsf{Dec}(sk, c)$

We claim

$$|\Pr[G_1^{\mathsf{B}} \Rightarrow 1] - \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \leq q_{\mathsf{G}} \cdot \delta \ . \tag{4}$$

To show Equation (4), observe that the whole Game G_1 (and also the whole Game G_2) makes at most q_{G} (distinct) queries $\mathsf{G}(m_1), \ldots, \mathsf{G}(m_{q_{\mathsf{G}}})$ to G. Again, we call such a query $\mathsf{G}(m_i)$ *problematic* iff it exhibits a correctness error in PKE_1 (in the sense that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m_i; \mathsf{G}(m_i))) \neq m_i$). Clearly, if B makes a problematic query, then there exists an adversary F that wins the correctness game $\mathsf{COR\text{-}RO}$ in the random oracle model. Hence, the probability that at least one $\mathsf{G}(m_i)$ is problematic is at most $\delta_1(q_{\mathsf{G}}) \leq q_{\mathsf{G}} \cdot \delta$.

However, conditioned on the event that no query $\mathsf{G}(m_i)$ is problematic, Game G_1 and Game G_2 proceed identically (cf. Figure 6). Indeed, the two games only differ if B submits a $\textsc{Pco}$ query (m, c) or a $\textsc{Cvo}$ query c together with a G query m such that $\mathsf{G}(m)$ is problematic and $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$. (In this case, G_1 will answer the query with 0, while G_2 will answer with 1.) This shows Equation (4).

GAME G_3. In Game G_3, we add a flag QUERY in line 09 and abort when it is raised. Hence, G_2 and G_3 only differ if QUERY is raised, meaning that B made a query G on m^*, or, equivalently, $(m^*, \cdot) \in \mathfrak{L}_G$. Due to the difference lemma [Sho04b],

$$|\Pr[G_3^{\mathsf{B}} \Rightarrow 1] - \Pr[G_2^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{QUERY}].$$

We first bound $\Pr[G_3^{\mathsf{B}} \Rightarrow 1]$ by constructing an adversary C in Figure 7 against the $\mathsf{OW\text{-}CPA}$ security of the original encryption scheme PKE. C inputs $(pk, c^* \leftarrow \mathsf{Enc}(pk, m^*))$ for random, unknown m^*, perfectly simulates game G_3 for B, and finally outputs $m' = m^*$ if B wins in game G_3.

$$\Pr[G_3^{\mathsf{B}} \Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C}) \ .$$

So far we have established the bound

$$\mathrm{Adv}^{\mathsf{OW\text{-}PCVA}}_{\mathsf{PKE}_1}(\mathsf{B}) \leq q_{\mathsf{G}} \cdot \delta + q_V \cdot 2^{-\gamma} + \Pr[\text{QUERY}] + \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C}) \ . \tag{5}$$

$\underline{\mathsf{C}(pk, c^*)}$	$\underline{\mathsf{D}(pk, c^*)}$
01 $m' \leftarrow \mathsf{B}^{\mathsf{G}(\cdot),\textsc{Pco}(\cdot,\cdot)}(pk, c^*)$	03 $m \leftarrow \mathsf{B}^{\mathsf{G}(\cdot),\textsc{Pco}(\cdot,\cdot)}(pk, c^*)$
02 **return** m'	04 $(m', r') \xleftarrow{\$} \mathfrak{L}_G$
	05 **return** m'

Figure 7: Adversaries C and Dagainst OW-CPA for the proof of Theorem 3.1. Oracles Pco, Cvo are defined as in game G_3, and G is defined as in game G_2 of Figure 6.

$\underline{\mathsf{D}_1(pk)}$	$\underline{\mathsf{D}_2(pk, c^*, st)}$
06 $st := (m_0^*, m_1^*) \xleftarrow{\$} \mathcal{M}^2$	08 $m' \leftarrow \mathsf{B}^{\mathsf{G}(\cdot),\textsc{Pco}(\cdot),\textsc{Cvo}(\cdot)}(pk, c^*)$
07 **return** st	09 $b' := \begin{cases} 0 & \lvert\mathfrak{L}_G(m_0^*)\rvert > \lvert\mathfrak{L}_G(m_1^*)\rvert \\ 1 & \lvert\mathfrak{L}_G(m_1^*)\rvert < \lvert\mathfrak{L}_G(m_0^*)\rvert \\ \xleftarrow{\$} \{0,1\} & \text{otherwise} \end{cases}$
	10 **return** b'

Figure 8: Adversary $\mathsf{D} = (\mathsf{D}_1, \mathsf{D}_2)$ against IND-CPA for the proof of Theorem 3.2. For fixed $m \in \mathcal{M}$, $\mathfrak{L}_G(m)$ is the set of all $(m, r) \in \mathfrak{L}_G$. Oracles Pco, Cvo are defined as in game G_3, and G is defined as in game G_2 of Figure 6.

Finally, in Figure 7 we construct an adversary D against the OW-CPA security of the original encryption scheme PKE, that inputs $(pk, c^* \leftarrow \mathsf{Enc}(pk, m^*))$, perfectly simulates game G_3 for B. If flag QUERY is set in G_3 then there exists en entry $(m^*, \cdot) \in \mathfrak{L}_\mathsf{G}$ and D returns the correct $m' = m^*$ with probability at most $1/q_\mathsf{G}$. We just showed

$$\Pr[\text{QUERY}] \leq q_\mathsf{G} \cdot \mathrm{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathsf{D}) \ .$$

Combining the latter bound with Equation (5) and folding C and D into one single adversary A against OW-CPA yields the required bound of the theorem. □

By definition, OW-PCA security is OW-PCVA security with $q_V := 0$ queries to the validity checking oracle. Hence, the bound of Theorem 3.1 shows that PKE_1 is in particular OW-PCA secure, without requiring PKE to be γ-spread.

Tight Security from IND-CPA. Whereas the reduction to OW-CPA security in Theorem 3.1 was non-tight, the following theorem establishes that OW-PCVA security of PKE_1 tightly reduces to IND-CPA security of PKE, in the random oracle model, given that PKE is γ-spread. If PKE is not γ-spread, then PKE_1 is still OW-PCA secure.

Theorem 3.2 (PKE IND-CPA $\overset{\text{ROM}}{\Rightarrow}$ PKE_1 OW-PCVA)**.** *Assume* PKE *to be* δ*-correct and* γ*-spread. Then, for any* OW-PCVA *adversary* B *that issues at most* q_G *queries to the random oracle* G*,* q_P *queries to a plaintext checking oracle* Pco*, and* q_V *queries to a validity checking oracle* Cvo*, there exists an* IND-CPA *adversary* A *such that*

$$\mathrm{Adv}^{\text{OW-PCVA}}_{\mathsf{PKE}_1}(\mathsf{B}) \leq q_\mathsf{G} \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{2q_\mathsf{G} + 1}{|\mathcal{M}|} + 3 \cdot \mathrm{Adv}^{\text{IND-CPA}}_{\mathsf{PKE}}(\mathsf{A})$$

and the running time of A *is about that of* B*.*

Proof. Considering the games of Figure 6 from the proof of Theorem 3.1 we obtain by Equation (5)

$$\begin{aligned} \mathrm{Adv}^{\text{OW-PCVA}}_{\mathsf{PKE}_1}(\mathsf{B}) &\leq q_\mathsf{G} \cdot \delta + q_V \cdot 2^{-\gamma} + \Pr[\text{QUERY}] + \mathrm{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathsf{C}) \\ &\leq q_\mathsf{G} \cdot \delta + q_V \cdot 2^{-\gamma} + \Pr[\text{QUERY}] + \frac{1}{|\mathcal{M}|} + \mathrm{Adv}^{\text{IND-CPA}}_{\mathsf{PKE}}(\mathsf{C}) \ , \end{aligned}$$

where the last inequation uses Lemma 2.3.

In Figure 8 we construct an adversary $\mathsf{D} = (\mathsf{D}_1, \mathsf{D}_2)$ against the IND-CPA security of the original encryption scheme PKE that wins if flag QUERY is set in G_3. The first adversary D_1 picks two random

messages m_0^*, m_1^*. The second adversary D_2 inputs $(pk, c^* \leftarrow \mathsf{Enc}(pk, m_b^*), st)$, for an unknown random bit b, and runs B on (pk, c^*), simulating its view in game G_3. Note that by construction message m_b^* is uniformly distributed.

Consider game $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ with random challenge bit b. Let BADG be the event that B queries random oracle G on m_{1-b}^*. Since m_{1-b}^* is uniformly distributed and independent from B's view, we have $\Pr[\text{BADG}] \leq q_{\mathsf{G}}/|\mathcal{M}|$. For the remainder of the proof we assume BADG did not happen, i.e. $|\mathfrak{L}_G(m_{1-b}^*)| = 0$.

If QUERY happens, then B queried the random oracle G on m_b^*, which implies $|\mathfrak{L}_G(m_b^*)| > 0 = |\mathfrak{L}_G(m_{1-b}^*)|$ and therefore $b = b'$. If QUERY does not happen, then B did not query random oracle G on m_b^*. Hence, $|\mathfrak{L}_G(m_b^*)| = |\mathfrak{L}_G(m_{1-b}^*)| = 0$ and $\Pr[b = b'] = 1/2$ since A picks a random bit b'. Overall, we have

$$\begin{aligned} \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathsf{D}) + \frac{q_{\mathsf{G}}}{|\mathcal{M}|} &\geq \left| \Pr[b = b'] - \frac{1}{2} \right| \\ &= \left| \Pr[\text{QUERY}] + \frac{1}{2} \Pr[\neg\text{QUERY}] - \frac{1}{2} \right| \\ &= \frac{1}{2} \Pr[\text{QUERY}]. \end{aligned}$$

Folding C and D into one single $\mathsf{IND\text{-}CPA}$ adversary A yields the required bound of the theorem. □

With the same argument as in Theorem 3.1, a tight reduction to $\mathsf{OW\text{-}PCA}$ security is implied without requiring PKE to be γ-spread.

3.2 Transformations $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\not\perp}_m$, $\mathsf{U}^{\perp}$, $\mathsf{U}^{\perp}_m$

In this section we introduce four variants of a transformation U, namely $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\not\perp}_m$, $\mathsf{U}^{\perp}$, $\mathsf{U}^{\perp}_m$, that convert a public-key encryption scheme PKE_1 into a key encapsulation mechanism KEM. Their differences are summarized in the following table.

Transformation	Rejection of invalid ciphertexts	KEM key	PKE_1's requirements
$\mathsf{U}^{\not\perp}$	implicit	$K = \mathsf{H}(m, c)$	$\mathsf{OW\text{-}PCA}$
$\mathsf{U}^{\perp}$	explicit	$K = \mathsf{H}(m, c)$	$\mathsf{OW\text{-}PCVA}$
$\mathsf{U}^{\not\perp}_m$	implicit	$K = \mathsf{H}(m)$	det. + $\mathsf{OW\text{-}CPA}$
$\mathsf{U}^{\perp}_m$	explicit	$K = \mathsf{H}(m)$	det. + $\mathsf{OW\text{-}VA}$

3.2.1 Transformation $\mathsf{U}^{\perp}$: from $\mathsf{OW\text{-}PCVA}$ to $\mathsf{IND\text{-}CCA}$

$\mathsf{U}^{\perp}$ transforms an $\mathsf{OW\text{-}PCVA}$ secure public-key encryption scheme into an $\mathsf{IND\text{-}CCA}$ secure key encapsulation mechanism. The $^{\perp}$ in $\mathsf{U}^{\perp}$ means that decapsulation of an invalid ciphertext results in the rejection symbol $\perp$ ("explicit rejection").

The Construction. To a public-key encryption scheme $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ with message space $\mathcal{M}$, and a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$, we associate $\mathsf{KEM}^{\perp} = \mathsf{U}^{\perp}[\mathsf{PKE}_1, \mathsf{H}]$. The algorithms of $\mathsf{KEM}^{\perp} = (\mathsf{Gen}_1, \mathsf{Encaps}, \mathsf{Decaps}^{\perp})$ are defined in Figure 9.

$\underline{\mathsf{Encaps}(pk)}$	$\underline{\mathsf{Decaps}^{\perp}(sk, c)}$
01 $m \xleftarrow{\$} \mathcal{M}$	05 $m' := \mathsf{Dec}_1(sk, c)$
02 $c \leftarrow \mathsf{Enc}_1(pk, m)$	06 **if** $m' = \perp$ **return** $\perp$
03 $K := \mathsf{H}(m, c)$	07 **else return**
04 **return** (K, c)	$K := \mathsf{H}(m', c)$

Figure 9: $\mathsf{IND\text{-}CCA}$-secure key encapsulation mechanism $\mathsf{KEM}^{\perp} = \mathsf{U}^{\perp}[\mathsf{PKE}_1, \mathsf{H}]$.

Security. The following theorem establishes that $\mathsf{IND\text{-}CCA}$ security of $\mathsf{KEM}^{\perp}$ tightly reduces to the $\mathsf{OW\text{-}PCVA}$ security of PKE_1, in the random oracle model.

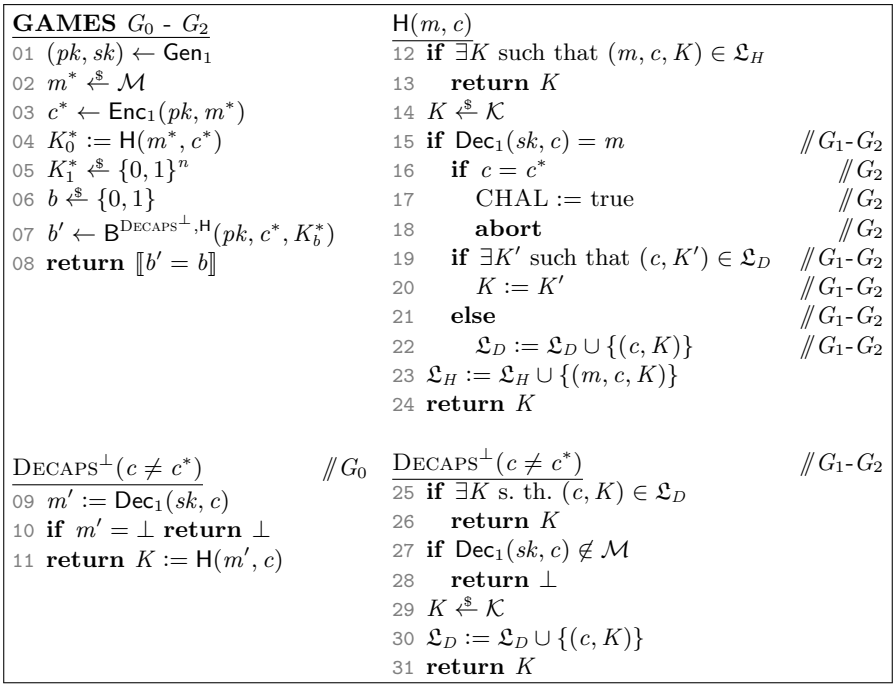


Figure 10: Games G_0 - G_2 for the proof of Theorem 3.3.

Theorem 3.3 (PKE_1 OW-PCVA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{KEM}^\perp$ IND-CCA)**.** *If* PKE_1 *is* δ_1*-correct, so is* $\mathsf{KEM}^\perp$*. For any* IND-CCA *adversary* B *against* $\mathsf{KEM}^\perp$*, issuing at most* q_D *queries to the decapsulation oracle* $\textsc{Decaps}^\perp$ *and at most* q_H *queries to the random oracle* H*, there exists an* OW-PCVA *adversary* A *against* PKE_1 *that makes at most* q_H *queries both to the* $\textsc{Pco}$ *oracle and to the* $\textsc{Cvo}$ *oracle such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^\perp}(\mathsf{B}) \leq \mathrm{Adv}^{\mathsf{OW\text{-}PCVA}}_{\mathsf{PKE}_1}(\mathsf{A})$$

and the running time of A *is about that of* B*.*

The main idea of the proof is to simulate the decapsulation oracle without the secret-key. This can be done by answering decryption queries with a random key and then later patch the random oracle using the plaintext checking oracle $\textsc{Pco}(\cdot,\cdot)$ provided by the OW-PCVA game. Additionally, the ciphertext validity oracle $\textsc{Cvo}(\cdot)$ is required to reject decapsulation queries with inconsistent ciphertexts.

Proof. It is easy to verify the correctness bound. Let B be an adversary against the IND-CCA security of $\mathsf{KEM}^\perp$, issuing at most q_D queries to $\textsc{Decaps}^\perp$ and at most q_H queries to H. Consider the games given in Figure 10.

$\textsc{Game}$ G_0. Since game G_0 is the original IND-CCA game,

$$\left|\Pr[G_0^\mathsf{B} \Rightarrow 1] - \frac{1}{2}\right| = \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^\perp}(\mathsf{B}) \ .$$

$\textsc{Game}$ G_1. In game G_1, the oracles H and $\textsc{Decaps}^\perp$ are modified such that they make no use of the secret key any longer except by testing if $\mathsf{Dec}_1(sk', c) = m$ for given (m, c) in line 15 and if $\mathsf{Dec}_1(sk, c) \in \mathcal{M}$ for given c in line 27. Game G_1 contains two sets: hash list $\mathfrak{L}_H$ that contains all entries (m, c, K) where H was queried on (m, c), and set $\mathfrak{L}_D$ that contains all entries (c, K) where either H was queried on (m', c), $m' := \mathsf{Dec}_1(sk', c)$, or $\textsc{Decaps}^\perp$ was queried on c. In order to show that the view of B is identical in games G_0 and G_1, consider the following cases for a fixed ciphertext c and $m' := \mathsf{Dec}_1(sk', c)$.

- Case 1: $m' \notin \mathcal{M}$. Since $\textsc{Cvo}(c) = 0$ is equivalent to $m' = \perp$, $\textsc{Decaps}^\perp(c)$ returns $\perp$ as in both games.

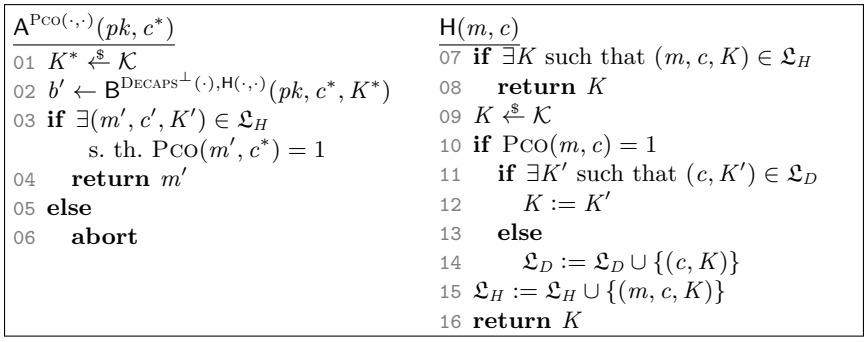


Figure 11: Adversary A against OW-PCVA for the proof of Theorem 3.3, where $\text{DECAPS}^{\perp}$ is defined as in Game G_2 of Figure 10.

- Case 2: $m' \in \mathcal{M}$. We will now show that H in game G_1 is "patched", meaning that it is ensures $\text{DECAPS}^{\perp}(c) = \mathsf{H}(m', c)$, where $m' := \mathsf{Dec}_1(sk, c)$, for all ciphertexts c with $m' \in \mathcal{M}$. We distinguish two sub-cases: B might either first query H on (m', c), then $\text{DECAPS}^{\perp}$ on c, or the other way round.
  - If H is queried on (m', c) first, it is recognized that $\mathsf{Dec}_1(sk, c) = m$ in line 15. Since DECAPS was not yet queried on c, no entry of the form (c, K) can already exist in $\mathfrak{L}_D$. Therefore, besides adding $(m, c, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_H$, H also adds (c, K) to $\mathfrak{L}_D$ in line 22, thereby defining $\text{DECAPS}^{\perp}(c) := K = \mathsf{H}(m', c)$.
  - If $\text{DECAPS}^{\perp}$ is queried on c first, no entry of the form (c, K) exists in $\mathfrak{L}_D$ yet. Therefore, $\text{DECAPS}^{\perp}$ adds $(c, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_D$, thereby defining $\text{DECAPS}^{\perp}(c) := K$. When queried on (m', c) afterwards, H recognizes that $\mathsf{Dec}_1(sk, c) = m'$ in line 15 and that an entry of the form (c, K) already exists in $\mathfrak{L}_D$ in line 19. By adding (m, c, K) to $\mathfrak{L}_H$ and returning K, H defines $\mathsf{H}(m', c) := K = \text{DECAPS}^{\perp}(c)$.

We have shown that B's view is identical in both games and

$$\Pr[G_1^{\mathsf{B}} \Rightarrow 1] = \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \ .$$

GAME G_2. From game G_2 on we proceed identical to the proof of Theorem 3.4. That is, we abort immediately on the event that B queries H on (m^*, c^*). Denote this event as CHAL. Due to the difference lemma,

$$|\Pr[G_2^{\mathsf{B}} \Rightarrow 1] - \Pr[G_1^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{CHAL}] \ .$$

In game G_2, $\mathsf{H}(m^*, c^*)$ will not be given to B; neither through a hash nor a decryption query, meaning bit b is independent from B's view. Hence,

$$\Pr[G_2^{\mathsf{B}}] = \frac{1}{2} \ .$$

It remains to bound Pr[CHAL]. To this end, we construct an adversary A against the OW-PCVA security of PKE_1 simulating G_2 for B as in Figure 11. Note that the simulation is perfect. Since CHAL implies that B queried $\mathsf{H}(m^*, c^*)$ which implies $(m^*, c^*, K') \in \mathfrak{L}_H$ for some K', and A returns $m' = m^*$. Hence,

$$\Pr[\text{CHAL}] = \text{Adv}^{\text{OW-PCVA}}_{\mathsf{PKE}}(\mathsf{A}) \ .$$

Collecting the probabilities yields the required bound. □

3.2.2 Transformation $\mathsf{U}^{\not\perp}$: from OW-PCA to IND-CCA

$\mathsf{U}^{\not\perp}$ is a variant of $\mathsf{U}^{\perp}$ with "implicit rejection" of inconsistent ciphertexts. It transforms an OW-PCA secure public-key encryption scheme into an IND-CCA secure key encapsulation mechanism.

THE CONSTRUCTION. To a public-key encryption scheme $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ with message space $\mathcal{M}$, and a random oracle $\mathsf{H} : \{0,1\}^* \to \mathcal{M}$ we associate $\mathsf{KEM}^{\not\perp} = \mathsf{U}^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}] = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}^{\not\perp})$.

The algorithms of $\mathsf{KEM}^{\not\perp}$ are defined in Figure 12, Encaps is the same as in $\mathsf{KEM}^{\perp}$ (Figure 9). Note that $\mathsf{U}^{\perp}$ and $\mathsf{U}^{\not\perp}$ essentially differ in decapsulation: $\mathsf{Decaps}^{\perp}$ from $\mathsf{U}^{\perp}$ rejects if c decrypts to $\perp$, whereas $\mathsf{Decaps}^{\not\perp}$ from $\mathsf{U}^{\not\perp}$ returns a pseudorandom key K.

$\underline{\mathsf{Gen}^{\not\perp}}$	$\underline{\mathsf{Encaps}(pk)}$	$\underline{\mathsf{Decaps}^{\not\perp}(sk, c)}$
01 $(pk', sk') \leftarrow \mathsf{Gen}_1$	05 $m \xleftarrow{\$} \mathcal{M}$	09 Parse $sk = (sk', s)$
02 $s \xleftarrow{\$} \mathcal{M}$	06 $c \leftarrow \mathsf{Enc}_1(pk, m)$	10 $m' := \mathsf{Dec}_1(sk', c)$
03 $sk := (sk', s)$	07 $K := \mathsf{H}(m, c)$	11 **if** $m' \neq \perp$
04 **return** (pk', sk)	08 **return** (K, c)	12 **return** $K := \mathsf{H}(m', c)$
		13 **else return** $K := \mathsf{H}(s, c)$

Figure 12: IND-CCA-secure key encapsulation mechanism $\mathsf{KEM}^{\not\perp} = \mathsf{U}^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}]$.

Security. The following theorem establishes that IND-CCA security of $\mathsf{KEM}^{\not\perp}$ tightly reduces to the OW-PCA security of PKE_1, in the random oracle model.

Theorem 3.4 (PKE_1 OW-PCA $\overset{\text{ROM}}{\Rightarrow}$ KEM IND-CCA)**.** *If* PKE_1 *is* δ_1*-correct, then* $\mathsf{KEM}^{\not\perp}$ *is* δ_1*-correct in the random oracle model. For any* IND-CCA *adversary* B *against* $\mathsf{KEM}^{\not\perp}$*, issuing at most* q_D *queries to the decapsulation oracle* $\textsc{Decaps}^{\not\perp}$ *and at most* q_H *queries to the random oracle* H*, there exists an* OW-PCA *adversary* A *against* PKE_1 *that makes at most* q_H *queries to the* Pco *oracle such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathsf{B}) \leq \frac{q_\mathsf{H}}{|\mathcal{M}|} + \mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}_1}(\mathsf{A})$$

and the running time of A *is about that of* B*.*

The proof is very similar to the one of Theorem 3.3. The difference is the handling of decapsulation queries with inconsistent ciphertexts. Since the OW-PCA experiment does not provide a Cvo oracle, the simulation of such queries has to be integrated into the random oracle patching technique.

Proof. It is easy to verify the correctness bound. Let B be an adversary against the IND-CCA security of KEM, issuing at most q_D queries to Dec and at most q_H queries to H. Consider the games given in Figure 13.

Game G_0. Since game G_0 is the original IND-CCA game,

$$\left|\Pr[G_0^{\mathsf{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathsf{B}) \ .$$

Game G_1. In game G_1 we make two changes. First, we raise flag QUERY and abort if $\mathsf{H}(s, \cdot)$ is queried (lines 18 and 19). Second, we make the pseudorandom keys output by $\textsc{Decaps}^{\not\perp}$ perfectly random. That is, in $\textsc{Decaps}^{\not\perp}(c)$, we replace $K = \mathsf{H}(s, c)$ by $K = \mathsf{H}'(c)$ if $m' = \mathsf{Dec}_1(sk', c) = \perp$ (lines 13) or if $m' = \mathsf{Dec}_1(sk', c) = s$ (line 14), where H' is an independent internal random oracles that cannot be accessed by B. The latter remains unnoticed by B unless $\mathsf{H}(s, \cdot)$ is queried, in which case G_1 aborts. (Also note that $m' := \mathsf{Dec}_1(sk, c)$ is unique.) Since B's view is independent of (the uniform secret) s unless G_1 aborts,

$$|\Pr[G_1^{\mathsf{B}} \Rightarrow 1] - \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \leq \frac{q_\mathsf{H}}{|\mathcal{M}|} \ .$$

Game G_2. In game G_2, the oracles H and $\textsc{Decaps}^{\not\perp}$ are modified such that $\textsc{Decaps}^{\not\perp}$ does not make use of the secret key any longer except by testing if $\mathsf{Dec}_1(sk', c) = m$ for given (m, c) in line 20. In game G_2 we will use two lists, $\mathfrak{L}_H$ and $\mathfrak{L}_D$, for bookkeeping. $(m, c, K) \in \mathfrak{L}_H$ indicates that H was queried on (m, c) and $\mathsf{H}(m, c) = K$ holds; $(c, K) \in \mathfrak{L}_D$ indicates that $\textsc{Decaps}^{\not\perp}(c) = K$ holds and either H was queried on $(m := \mathsf{Dec}_1(sk', c), c)$ or $\textsc{Decaps}^{\not\perp}$ was queried on c. In order to show that the view of B is identical in games G_1 and G_2, consider the following cases for a fixed ciphertext c and $m' := \mathsf{Dec}_1(sk', c)$.

- Case 1: $m' \in \{\perp, s\}$. Since H cannot be queried on (m', c) (i.e., $\mathsf{H}(\perp, \cdot)$ is not allowed and $\mathsf{H}(s, c)$ results in abort), the simulation of H can never add a tuple of the form (c, K) to $\mathfrak{L}_D$. Hence, querying $\textsc{Decaps}^{\not\perp}(c)$ in game G_2 will return a uniformly random key, as in Game G_1.

GAMES G_0 - G_3

01 $(pk', sk') \leftarrow \mathsf{Gen}_1$
02 $s \xleftarrow{\$} \mathcal{M}$
03 $sk := (sk', s)$
04 $m^* \xleftarrow{\$} \mathcal{M}$
05 $c^* \leftarrow \mathsf{Enc}_1(pk, m^*)$
06 $K_0^* := \mathsf{H}(m^*, c^*)$
07 $K_1^* \xleftarrow{\$} \{0,1\}^n$
08 $b \xleftarrow{\$} \{0,1\}$
09 $b' \leftarrow \mathsf{B}^{\text{DECAPS}^{\not\perp}(\cdot),\mathsf{H}(\cdot)}(pk', c^*, K_b^*)$
10 **return** $[\![b' = b]\!]$

$\mathsf{H}(m, c)$

16 **if** $\exists K$ s. th. $(m, c, K) \in \mathfrak{L}_H$ **return** K
17 $K \xleftarrow{\$} \mathcal{K}$
18 **if** $m = s$ // G_1-G_3
19 QUERY := true; **abort** // G_1-G_3
20 **if** $\mathsf{Dec}_1(sk', c) = m$ // G_2-G_3
21 **if** $c = c^*$ // G_3
22 CHAL := true; **abort** // G_3
23 **if** $\exists K'$ such that $(c, K') \in \mathfrak{L}_D$ // G_2-G_3
24 $K := K'$ // G_2-G_3
25 **else** // G_2-G_3
26 $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$ // G_2-G_3
27 $\mathfrak{L}_H := \mathfrak{L}_H \cup \{(m, c, K)\}$
28 **return** K

$\text{DECAPS}^{\not\perp}(c \neq c^*)$ // G_0-G_1

11 $m' := \mathsf{Dec}_1(sk', c)$
12 **if** $m' = \perp$ **return** $K := \mathsf{H}(s, c)$ // G_0
13 **if** $m' = \perp$ **return** $K := \mathsf{H}'(c)$ // G_1
14 **if** $m' = s$ **return** $K := \mathsf{H}'(c)$ // G_1
15 **return** $K := \mathsf{H}(m', c)$

$\text{DECAPS}^{\not\perp}(c \neq c^*)$ // G_2-G_3

29 **if** $\exists K$ s. th. $(c, K) \in \mathfrak{L}_D$
30 **return** K
31 **else**
32 $K \xleftarrow{\$} \mathcal{K}$
33 $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$
34 **return** K

Figure 13: Games G_0 - G_3 for the proof of Theorem 3.4 . H' (lines 13 and 14) is an independent internal random oracle that cannot be accessed by B.

- Case 2: $m' \notin \{\perp, s\}$. We will now show that H in game G_2 is "patched", meaning that it is ensures $\text{DECAPS}^{\not\perp}(c) = \mathsf{H}(m', c)$, where $m' := \mathsf{Dec}_1(sk', c)$, for all valid ciphertexts c with $\mathsf{Dec}_1(sk', c) \neq s$. We distinguish two sub-cases: B might either first query H on (m', c), then $\text{DECAPS}^{\not\perp}$ on c, or the other way round.
  - If H is queried on (m', c) first, it is recognized that $\mathsf{Dec}_1(sk', c) = m'$ in line 20. Since $\text{DECAPS}^{\not\perp}$ was yet not queried on c, no entry of the form (c, K) already exists in $\mathfrak{L}_D$. Therefore, besides adding $(m', c, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_H$, H also adds (c, K) to $\mathfrak{L}_D$ in line 26, thereby defining $\text{DECAPS}^{\not\perp}(c) := K = \mathsf{H}(m', c)$.
  - If $\text{DECAPS}^{\not\perp}$ is queried on c first, no entry of the form (c, K) exists in $\mathfrak{L}_D$ yet. Therefore, $\text{DECAPS}^{\not\perp}$ adds $(c, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_D$ thereby defining $\text{DECAPS}^{\not\perp}(c) := K$. When queried on (m', c) afterwards, H recognizes that $\mathsf{Dec}_1(sk', c) = m'$ in line 20 and that an entry of the form (c, K) already exists in $\mathfrak{L}_D$ in line 23. By adding (m', c, K) to $\mathfrak{L}_H$ and returning K, H defines $\mathsf{H}(m', c) := K = \text{DECAPS}^{\not\perp}(c)$.

We have shown that B's view is identical in both games and

$$\Pr[G_2^{\mathsf{B}} \Rightarrow 1] = \Pr[G_1^{\mathsf{B}} \Rightarrow 1]| \ .$$

GAME G_3. In game G_3, we abort immediately (and raise flag CHAL) on the event that B queries H on (m^*, c^*), where m^* is the challenge message. Due to the difference lemma,

$$|\Pr[G_3^{\mathsf{B}} \Rightarrow 1] - \Pr[G_2^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{CHAL}] \ .$$

In game G_3, $\mathsf{H}(m^*, c^*)$ will not be given to B; neither through a hash nor a decryption query, meaning bit b is independent from B's view. Hence,

$$\Pr[G_3^{\mathsf{B}}] = \frac{1}{2} \ .$$

It remains to bound Pr[CHAL]. To this end, we construct an adversary A against the OW-PCA security of PKE_1 simulating G_3 for B as in Figure 14. Note that the simulation is perfect. Since CHAL

$\underline{\mathsf{A}^{\text{PCO}(\cdot)}(pk, c^*)}$
01 $K^* \xleftarrow{\$} \mathcal{K}$
02 $s \xleftarrow{\$} \mathcal{M}$
03 $b' \leftarrow \mathsf{B}^{\text{DECAPS}^{\not\perp}(\cdot),\mathsf{H}(\cdot)}(pk, c^*, K^*)$
04 **if** $\exists (m', c', K') \in \mathfrak{L}_H$
s. th. $\text{PCO}(m', c^*) = 1$
05 **return** m'
06 **else**
07 **abort**

$\underline{\mathsf{H}(m, c)}$
08 **if** $\exists K$ s. th. $(m, c, K) \in \mathfrak{L}_H$ **return** K
09 $K \xleftarrow{\$} \mathcal{K}$
10 **if** $m = s$
11 **abort**
12 **if** $\text{PCO}(m, c) = 1$
13 **if** $\exists K'$ s. th. $(c, K') \in \mathfrak{L}_D$
14 $K := K'$
15 **else**
16 $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$
17 $\mathfrak{L}_H := \mathfrak{L}_H \cup \{(m, c, K)\}$
18 **return** K

Figure 14: Adversary A against OW-PCA for the proof of Theorem 3.4. Oracle $\text{DECAPS}^{\not\perp}$ is defined as in game G_3 of Figure 13.

implies that B queried $\mathsf{H}(m^*, c^*)$ which implies $(m^*, c^*, K') \in \mathfrak{L}_H$ (for some K'), A returns $m' = m^*$ and wins its OW-PCA game. Hence,

$$\Pr[\text{CHAL}] = \text{Adv}^{\text{OW-PCA}}_{\mathsf{PKE}}(\mathsf{A}) \ .$$

Collecting the probabilities yields the required bound. □

3.2.3 Transformations $\mathsf{U}^{\not\perp}_m/\mathsf{U}^{\perp}_m$: from OW-CPA/OW-VA to IND-CCA for deterministic Encryption

Transformation $\mathsf{U}^{\not\perp}_m$ is a variant of $\mathsf{U}^{\not\perp}$ that derives the KEM key as $K = \mathsf{H}(m)$, instead of $K = \mathsf{H}(m, c)$. It transforms a OW-VA secure public-key encryption scheme with deterministic encryption (e.g., the ones obtained via T from Section 3.1) into an IND-CCA secure key encapsulation mechanism. We also consider an implicit rejection variant $\mathsf{U}^{\perp}_m$ that only requires OW-CPA security of the underlying encryption scheme PKE.

THE CONSTRUCTION. To a public-key encryption scheme $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ with message space $\mathcal{M}$, and a random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$, we associate $\mathsf{KEM}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m[\mathsf{PKE}_1, \mathsf{H}] = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}_m, \mathsf{Decaps}^{\not\perp}_m)$ and $\mathsf{KEM}^{\perp}_m = \mathsf{U}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}] = (\mathsf{Gen}_1, \mathsf{Encaps}_m, \mathsf{Decaps}^{\perp}_m)$. Algorithm $\mathsf{Gen}^{\not\perp}$ is given in Figure 12 and the remaining algorithms of $\mathsf{KEM}^{\not\perp}_m$ and $\mathsf{KEM}^{\perp}_m$ are defined in Figure 15.

$\underline{\mathsf{Encaps}_m(pk)}$
01 $m \xleftarrow{\$} \mathcal{M}$
02 $c := \mathsf{Enc}_1(pk, m)$
03 $K := \mathsf{H}(m)$
04 **return** (K, c)

$\underline{\mathsf{Decaps}^{\not\perp}_m(sk, c)}$
05 Parse $sk = (sk', s)$
06 $m' := \mathsf{Dec}_1(sk', c)$
07 **if** $m' \neq \perp$
08 **return** $K := \mathsf{H}(m')$
09 **else return** $K := \mathsf{H}(s, c)$

$\underline{\mathsf{Decaps}^{\perp}_m(sk, c)}$
10 $m' := \mathsf{Dec}_1(sk, c)$
11 **if** $m' = \perp$ **return** $\perp$
12 **else return**
$K := \mathsf{H}(m')$

Figure 15: IND-CCA-secure key encapsulation mechanisms $\mathsf{KEM}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$ and $\mathsf{KEM}^{\perp}_m = \mathsf{U}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}]$.

SECURITY OF $\mathsf{KEM}^{\perp}_m$. The following theorem establishes that IND-CCA security of $\mathsf{KEM}^{\perp}_m$ tightly reduces to the OW-VA security of PKE_1, in the random oracle model.

Theorem 3.5 (PKE_1 OW-VA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{KEM}^{\perp}_m$ IND-CCA). *If PKE_1 is δ_1-correct, then so is $\mathsf{KEM}^{\perp}_m$. Let G denote the random oracle that PKE_1 uses (if any), and let $q_{\mathsf{Enc}_1,\mathsf{G}}$ and $q_{\mathsf{Dec}_1,\mathsf{G}}$ denote an upper bound on the number of G-queries that Enc_1, resp. Dec_1 makes upon a single invocation. If Enc_1 is deterministic then, for any IND-CCA adversary B against $\mathsf{KEM}^{\perp}_m$, issuing at most q_D queries to the decapsulation oracle $\text{DECAPS}^{\perp}_m$ and at most q_{G}, resp. q_{H} queries to its random oracles G and H, there exists an OW-VA adversary A against PKE_1 that makes at most q_D queries to the CVO oracle such that*

$$\text{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}^{\perp}_m}(\mathsf{B}) \leq \text{Adv}^{\text{OW-VA}}_{\mathsf{PKE}_1}(\mathsf{A}) + \delta_1(q_{\mathsf{G}} + (q_{\mathsf{H}} + q_D)(q_{\mathsf{Enc}_1,\mathsf{G}} + q_{\mathsf{Dec}_1,\mathsf{G}}))$$

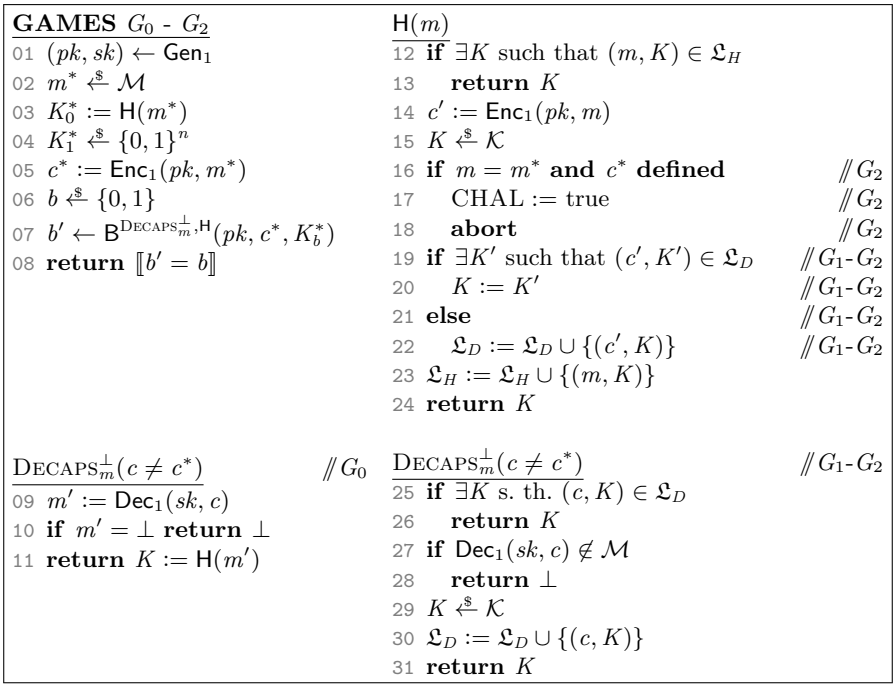


Figure 16: Games G_0 - G_3 for the proof of Theorem 3.5

and the running time of A *is about that of* B.

The proof is similar to the one of Theorem 3.3. A naive adaptation would reduce to PKE_1's OW-PCVA security and make $O(q_{\mathsf{H}} q_D)$ queries to the Pco oracle. Instead, we exploit the deterministic Enc_1 to (implicitly) simulate our own Pco oracle via re-encryption during te proof.

Proof. It is easy to verify the correctness bound. (Note that the correctness error δ_1 of $\mathsf{KEM}_m^{\perp}$ is independent of the number of H-queries that an adversary on $\mathsf{KEM}_m^{\perp}$'s correctness makes.)

To show security of $\mathsf{KEM}_m^{\perp}$, let B be an adversary against the IND-CCA security of $\mathsf{KEM}_m^{\perp}$, issuing at most q_D queries to $\textsc{Decaps}_m^{\perp}$ and at most q_{H} queries to H. Consider the games given in Figure 16.

Game G_0. Since game G_0 is the original IND-CCA game,

$$\left| \Pr[G_0^{\mathsf{B}} \Rightarrow 1] - \frac{1}{2} \right| = \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}_m^{\perp}}(\mathsf{B}) \ .$$

Game G_1. In game G_1, the oracles H and $\textsc{Decaps}_m^{\perp}$ are changed such that they make no use of the secret key any longer except for testing if $\mathsf{Dec}_1(sk, c) \in \mathcal{M}$ for given c in line 27. Game G_1 contains two sets: hash list $\mathfrak{L}_H$ which contains all entries (m, K) where H was queried on m, and set $\mathfrak{L}_D$ which contains all entries (c, K) where either $\textsc{Decaps}_m^{\perp}$ was queried on c, or H was queried on some message m such that both $c = \mathsf{Enc}_1(pk, m)$ and $\mathsf{Dec}_1(sk', c) = m$.

To analyze Game G_1 further, let QUERY denote the event that either

- $\mathfrak{L}_H$ contains an entry (m, K) with $\mathsf{Dec}_1(sk', \mathsf{Enc}_1(pk', m)) \neq m$, or
- $\mathfrak{L}_D$ contains an entry (c, K) with $\mathsf{Enc}_1(pk', \mathsf{Dec}_1(sk', c)) \neq c$

(or both). Intuitively, QUERY denotes the event that a correctness error of PKE_1 actually occurs.

We will show that the view of B is identical in games G_0 and G_1 unless QUERY occurs. To do so, consider the following cases for a fixed ciphertext c (placed as a $\textsc{Decaps}_m^{\perp}$ query) and $m := \mathsf{Dec}_1(sk', c)$.

- Case 1: $m \notin \mathcal{M}$. Since $\textsc{Cvo}(c) = 0$ is equivalent to $m = \perp$, $\textsc{Decaps}_m^{\perp}(c)$ returns $\perp$ in both games.

$\underline{\mathsf{A}^{\text{CVO}(\cdot)}(pk, c^*)}$
01 $K^* \xleftarrow{\$} \mathcal{K}$
02 $b' \leftarrow \mathsf{B}^{\text{DECAPS}_m^{\perp}(\cdot),\mathsf{H}(\cdot)}(pk, c^*, K^*)$
03 **if** $\exists(m', K') \in \mathfrak{L}_H$
s. th. $\mathsf{Enc}_1(pk, m') = c^*$
04 **return** m'
05 **else**
06 **abort**

$\underline{\text{DECAPS}_m^{\perp}(c \neq c^*)}$
07 **if** $\exists K$ s. th. $(c, K) \in \mathfrak{L}_D$
08 **return** K
09 **if** $\text{CVO}(c) = 0$
10 **return** $\perp$
11 $K \xleftarrow{\$} \mathcal{K}$
12 $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$
13 **return** K

Figure 17: Adversary A against OW-VA for the proof of Theorem 3.5, where H is defined as in Game G_2 of Figure 16.

- Case 2: $m \in \mathcal{M}$. We will now show that H in game G_1 is "patched", meaning that it is ensured that $\text{DECAPS}_m^{\perp}(c) = \mathsf{H}(m)$ for all ciphertexts c with $m \in \mathcal{M}$, conditioned on the additional prerequisite that $\mathsf{Enc}_1(pk, m) = c$. We distinguish two sub-cases: B might either first query H on m, then $\text{DECAPS}_m^{\perp}$ on c, or the other way round.
  - If H is queried on m first, it is recognized that $\mathsf{Dec}_1(sk, c') = \mathsf{Dec}_1(sk, c) = m$ in line 15 since we made the prerequisite that $c = c'$. Since DECAPS was not yet queried on c, no entry of the form (c, K) can already exist in $\mathfrak{L}_D$. Therefore, besides adding $(m, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_H$, H also adds (c, K) to $\mathfrak{L}_D$ in line 22, thereby defining $\text{DECAPS}_m^{\perp}(c) := K = \mathsf{H}(m)$.
  - We first prove that if $\text{DECAPS}_m^{\perp}$ is queried on c first, no entry of the form (c, K) exists in $\mathfrak{L}_D$ yet: Existence of such an entry implies that H already was queried on a message m' such that $m' \neq m$ (because $\text{DECAPS}_m^{\perp}$ is assumed to be queried first), with $\mathsf{Enc}_1(pk, m') = c$ and $\mathsf{Dec}_1(sk, c) = m'$, which clearly is contradictory to m' being unequal to m. Therefore, $\text{DECAPS}_m^{\perp}$ adds $(c, K \xleftarrow{\$} \mathcal{K})$ to $\mathfrak{L}_D$, thereby defining $\text{DECAPS}_m^{\perp}(c) := K$. When queried on m afterwards, H recognizes that $\mathsf{Dec}_1(sk, c') = \mathsf{Dec}_1(sk, c) = m$ in line 15 and that an entry of the form (c, K) already exists in $\mathfrak{L}_D$ in line 19. By adding (m, K) to $\mathfrak{L}_H$ and returning K, H defines $\mathsf{H}(m) := K = \text{DECAPS}_m^{\perp}(c)$.

We have shown that B's view is identical in both games unless a correctness error (in the form of QUERY) occurs. We can bound $\Pr[\text{QUERY}]$ with a straightforward reduction to the δ_1-correctness of PKE_1. In this reduction, an adversary on PKE_1's correctness simulates Game G_0 and additionally checks for QUERY upon every $\text{DECAPS}_m^{\perp}$ and every H query. In total, this takes $q_{\mathsf{G}} + (q_{\mathsf{H}} + q_D)(q_{\mathsf{Enc}_1,\mathsf{G}} + q_{\mathsf{Dec}_1,\mathsf{G}})$ queries to G: q_{G} queries from B, and $q_{\mathsf{Enc}_1,\mathsf{G}} + q_{\mathsf{Dec}_1,\mathsf{G}}$ additional queries upon each query to H and $\text{DECAPS}_m^{\perp}$ (in order to check for QUERY. Hence,

$$|\Pr[G_1^{\mathsf{B}} \Rightarrow 1] - \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{QUERY}] \leq \delta_1(q_{\mathsf{G}} + (q_{\mathsf{H}} + q_D)(q_{\mathsf{Enc}_1,\mathsf{G}} + q_{\mathsf{Dec}_1,\mathsf{G}})) \ .$$

GAME G_2. In game G_2, we abort (with uniformly random output) immediately on the event that B queries H on m^*. Denote this event as CHAL. Due to the difference lemma,

$$|\Pr[G_2^{\mathsf{B}} \Rightarrow 1] - \Pr[G_1^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{CHAL}] \ .$$

In game G_2, $\mathsf{H}(m^*)$ will not be given to B; neither through a hash nor a decryption query, meaning bit b is independent from B's view. Hence,

$$\Pr[G_2^{\mathsf{B}}] = \frac{1}{2} \ .$$

It remains to bound $\Pr[\text{CHAL}]$. To this end, we construct an adversary A against the OW-VA security of PKE_1 simulating G_2 for B as in Figure 17.

Note that the simulation is perfect until CHAL occurs. Furthermore, CHAL implies that B queried $\mathsf{H}(m^*)$, which implies that $(m^*, K') \in \mathfrak{L}_H$ for some K'. In this case, we have $\mathsf{Enc}_1(pk, m^*) = c^*$ (since Enc_1 is deterministic), and thus A returns m^*. Hence,

$$\Pr[\text{CHAL}] = \text{Adv}_{\mathsf{PKE}}^{\text{OW-VA}}(\mathsf{A}) \ .$$

Collecting the probabilities yields the required bound. □

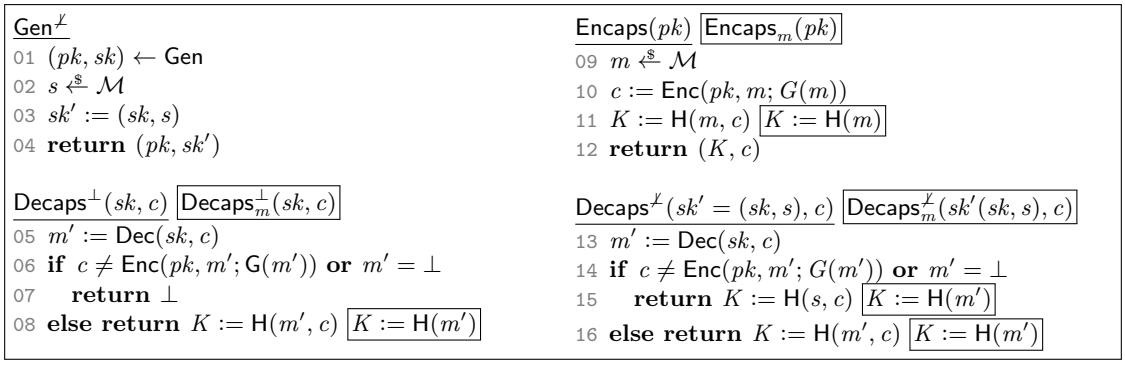


Figure 18: IND-CCA secure Key Encapsulation $\mathsf{KEM}^{\not\perp}$, $\mathsf{KEM}^{\perp}$, $\mathsf{KEM}^{\not\perp}_m$, and $\mathsf{KEM}^{\perp}_m$ obtained from PKE.

Security of $\mathsf{KEM}^{\not\perp}_m$. The following theorem establishes that IND-CCA security of $\mathsf{KEM}^{\not\perp}_m$ tightly reduces to the OW-CPA security of PKE_1, in the random oracle model. Its proof is easily obtained by combining the proofs of Theorem 3.4 and Theorem 3.5.

Theorem 3.6 (PKE_1 OW-CPA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{KEM}^{\not\perp}_m$ IND-CCA)**.** *If* PKE_1 *is* δ_1*-correct, then so is* $\mathsf{KEM}^{\not\perp}_m$*. Let* G *denote the random oracle that* PKE_1 *uses (if any), and let* $q_{\mathsf{Enc}_1,\mathsf{G}}$ *and* $q_{\mathsf{Dec}_1,\mathsf{G}}$ *denote an upper bound on the number of* G*-queries that* Enc_1*, resp.* Dec_1 *makes upon a single invocation. If* Enc_1 *is deterministic then, for any* IND-CCA *adversary* B *against* $\mathsf{KEM}^{\not\perp}_m$*, issuing at most* q_D *queries to the decapsulation oracle* $\textsc{Decaps}^{\not\perp}_m$ *and at most* q_{G}*, resp.* q_{H} *queries to its random oracles* G *and* H*, there exists an* OW-CPA *adversary* A *against* PKE_1 *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}_m}(\mathsf{B}) \leq \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}_1}(\mathsf{A}) + \frac{q_D}{|\mathcal{M}|} + \delta_1(q_{\mathsf{G}} + (q_{\mathsf{H}} + q_D)(q_{\mathsf{Enc}_1,\mathsf{G}} + q_{\mathsf{Dec}_1,\mathsf{G}}))$$

and the running time of A *is about that of* B.

3.3 The resulting KEMs

For completeness, we combine transformation T with $\{\mathsf{U}^{\not\perp}, \mathsf{U}^{\perp}, \mathsf{U}^{\not\perp}_m, \mathsf{U}^{\perp}_m\}$ from the previous sections to obtain four variants of the FO transformation $\mathsf{FO} := \mathsf{U}^{\not\perp} \circ \mathsf{T}$, $\mathsf{FO}^{\perp} := \mathsf{U}^{\perp} \circ \mathsf{T}$, $\mathsf{FO}^{\not\perp}_m := \mathsf{U}^{\not\perp}_m \circ \mathsf{T}$, and $\mathsf{FO}^{\perp}_m := \mathsf{U}^{\perp}_m \circ \mathsf{T}$. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, and hash functions $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$ we associate

$$\begin{aligned}
\mathsf{KEM}^{\not\perp} &= \mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}] = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}, \mathsf{Decaps}^{\not\perp}) \\
\mathsf{KEM}^{\perp} &= \mathsf{FO}^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}] = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp}) \\
\mathsf{KEM}^{\not\perp}_m &= \mathsf{FO}^{\not\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\not\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}] = (\mathsf{Gen}^{\not\perp}, \mathsf{Encaps}_m, \mathsf{Decaps}^{\not\perp}_m) \\
\mathsf{KEM}^{\perp}_m &= \mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}] = (\mathsf{Gen}, \mathsf{Encaps}_m, \mathsf{Decaps}^{\perp}_m)\ .
\end{aligned}$$

Their constituting algorithms are given in Figure 18.

The following table provides (simplified) concrete bounds of the IND-CCA security of $\mathsf{KEM} \in \{\mathsf{KEM}^{\not\perp}, \mathsf{KEM}^{\perp}, \mathsf{KEM}^{\not\perp}_m, \mathsf{KEM}^{\perp}_m\}$, directly obtained by combining Theorems 3.1–3.6. Here $q_{\mathsf{RO}} := q_{\mathsf{G}} + q_{\mathsf{H}}$ counts the total number of B's queries to the random oracles G and H and q_D counts the number of B's decryption queries.

KEM	Concrete bounds on $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{B}) \leq$	
$\mathsf{KEM}^{\not\perp}$	$q_{\mathsf{RO}} \cdot \delta + \frac{2q_{\mathsf{RO}}}{\lvert\mathcal{M}\rvert} + 2q_{\mathsf{RO}} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$	$q_{\mathsf{RO}} \cdot \delta + \frac{3q_{\mathsf{RO}}}{\lvert\mathcal{M}\rvert} + 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}')$
$\mathsf{KEM}^{\perp}$	$q_{\mathsf{RO}} \cdot (\delta + 2^{-\gamma}) + 2q_{\mathsf{RO}} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$	$q_{\mathsf{RO}} \cdot \left(\delta + 2^{-\gamma}\right) + \frac{3q_{\mathsf{RO}}}{\lvert\mathcal{M}\rvert} + 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}')$
$\mathsf{KEM}^{\not\perp}_m$	$(2q_{\mathsf{RO}} + q_D) \cdot \delta + \frac{2q_{\mathsf{RO}}}{\lvert\mathcal{M}\rvert} + 2q_{\mathsf{RO}} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$	$(2q_{\mathsf{RO}} + q_D) \cdot \delta + \frac{3q_{\mathsf{RO}}}{\lvert\mathcal{M}\rvert} + 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}')$
$\mathsf{KEM}^{\perp}_m$	$(2q_{\mathsf{RO}} + q_D) \cdot \delta + q_{\mathsf{RO}} \cdot 2^{-\gamma} + 2q_{\mathsf{RO}} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$	$(2q_{\mathsf{RO}} + q_D) \cdot \delta + q_{\mathsf{RO}} \cdot 2^{-\gamma} + 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}')$

CONCRETE PARAMETERS. For "κ bits of security" one generally requires that for all adversaries B with advantage $\mathrm{Adv}(\mathsf{B})$ and running in time $\mathrm{Time}(\mathsf{B})$, we have

$$\frac{\mathrm{Time}(\mathsf{B})}{\mathrm{Adv}(\mathsf{B})} \geq 2^{\kappa}.$$

The table below gives recommendations for the information-theoretic terms δ (correctness error of PKE, γ (γ-spreadness of PKE), and $\mathcal{M}$ (message space of PKE) appearing the concrete security bounds above.

Term in concrete bound	Minimal requirement for κ bits security
$q_{\mathsf{RO}} \cdot \delta$	$\delta \leq 2^{-\kappa}$
$q_{\mathsf{RO}} \cdot 2^{-\gamma}$	$\gamma \geq \kappa$
$q_{\mathsf{RO}}/\lvert\mathcal{M}\rvert$	$\lvert\mathcal{M}\rvert \geq 2^{\kappa}$

For example, if the concrete security bound contains the term $q_{\mathsf{RO}} \cdot \delta$, then with $\delta \leq 2^{-\kappa}$ one has

$$\frac{\mathrm{Time}(\mathsf{B})}{\mathrm{Adv}(\mathsf{B})} \geq \frac{q_{\mathsf{RO}}}{q_{\mathsf{RO}} \cdot \delta} = \frac{1}{\delta} \geq 2^{\kappa},$$

as required for κ bits security.

3.4 S^{ℓ}: from OW-CPA to IND-CPA Security, tightly

S^{ℓ} transforms an OW-CPA secure public-key encryption scheme into an IND-CPA secure scheme. The security reduction has a parameter ℓ which allows for a tradeoff between the security loss of the reduction and the compactness of ciphertexts.

THE CONSTRUCTION. Fix an $\ell \in \mathbb{N}$. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M} = \{0,1\}^n$ and a hash function $\mathsf{F} : \mathcal{M}^{\ell} \to \mathcal{R}$, we associate $\mathsf{PKE}_{\ell} = \mathsf{S}^{\ell}[\mathsf{PKE}, \mathsf{F}]$. The algorithms of PKE_{ℓ} are defined in Figure 19.

$\underline{\mathsf{Enc}_{\ell}(pk, m)}$	$\underline{\mathsf{Dec}_{\ell}(sk, c = (c_0, \ldots, c_{\ell}))}$
01 $\mathbf{x} := (x_1, \ldots, x_{\ell}) \xleftarrow{\$} (\{0,1\}^n)^{\ell}$	06 **for** $i = 1$ **to** ℓ **do**
02 $c_0 := m \oplus \mathsf{F}(\mathbf{x})$	07 $\quad x_i := \mathsf{Dec}(sk, c_i)$
03 **for** $i = 1$ **to** ℓ **do**	08 $\mathbf{x} := (x_1, \ldots, x_{\ell})$
04 $\quad c_i := \mathsf{Enc}(pk, x_i)$	09 **return** $c_0 \oplus \mathsf{F}(\mathbf{x})$
05 **return** $c := (c_0, \ldots, c_{\ell})$	

Figure 19: Tightly IND-CPA secure encryption PKE_{ℓ} obtained from PKE.

SECURITY. The following theorem shows that PKE_{ℓ} is IND-CPA secure, provided that PKE is OW-CPA secure.

Theorem 3.7 (PKE OW-CPA $\Rightarrow$ PKE_{ℓ} IND-CPA)**.** *If* PKE *is* δ*-correct (in the ROM), then* PKE_{ℓ} *is* $\ell \cdot \delta$*-correct. Moreover, for any* IND-CPA *adversary* B *that issues at most* q_{F} *queries to random oracle* F*, there exists an* OW-CPA *adversary* A *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_{\ell}}(\mathsf{B}) \leq q_{\mathsf{F}}^{1/\ell} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$$

and the running time of A *is about that of* B*.*

Proof. We first sketch correctness. Consider a public key pk and an encryption $c = (c_0, \ldots, c_{\ell})$ of generated by Enc_{ℓ}. Let x_i denote the respective value chosen by Enc_{ℓ} when generating c. Furthermore, let QUERY_i denote the event that, when decrypting c, the partial ciphertext c_i is decrypted to a value $x_i' \neq x_i$. If no QUERY_i occurs (for any i), then this implies that c is decrypted correctly. Hence, we have

$$\Pr[c \text{ decrypts incorrectly}] \leq \Pr[\bigvee_{i=1}^{\ell} \mathrm{QUERY}_i] \leq \sum_{i=1}^{\ell} \Pr[\mathrm{QUERY}_i] \overset{(*)}{=} \ell \cdot \delta,$$

where the probability is over the random coins of Gen_ℓ, Enc_ℓ, and Dec_ℓ, and $(*)$ follows from the δ-correctness of PKE. We note that this argument also applies verbatim in the ROM.

As for security, let $\mathsf{B} = (\mathsf{B}_1, \mathsf{B}_2)$ be an adversary against the IND-CPA security of PKE_ℓ, issuing at most q_F queries to F. Consider the games given in Figure 20.

GAMES G_0-G_1	$\mathsf{F}(\mathbf{x})$	
01 $(pk, sk) \leftarrow \mathsf{Gen}()$	11 **if** $\exists r$ s.t. $(\mathbf{x}, r) \in \mathfrak{L}_\mathsf{F}$	
02 $b \xleftarrow{\$} \{0,1\}$	12 **return** r	
03 $(m_0, m_1, st) \xleftarrow{\$} \mathsf{B}_1(pk)$	13 **if** $\mathbf{x} = \mathbf{x}^*$	// G_1
04 $\mathbf{x}^* := (x_1^*, \ldots, x_\ell^*) \xleftarrow{\$} (\{0,1\}^n)^\ell$	14 QUERY := true	// G_1
05 $c_0^* := m_b \oplus \mathsf{F}(\mathbf{x}^*)$	15 **abort**	// G_1
06 **for** $i = 1$ **to** ℓ **do**	16 $r \xleftarrow{\$} \mathcal{R}$	
07 $c_i^* := \mathsf{Enc}(pk, x_i)$	17 $\mathfrak{L}_\mathsf{F} := \mathfrak{L}_\mathsf{F} \cup \{(\mathbf{x}, r)\}$	
08 $c^* := (c_0^*, \ldots, c_\ell^*)$	18 **return** r	
09 $b' \xleftarrow{\$} \mathsf{B}_2(pk, c^*, st)$		
10 **return** $[\![b' = b]\!]$		

Figure 20: Games G_0 - G_1 for the proof of Theorem 3.7

GAME G_0. Since game G_0 is the original IND-CPA game,

$$\left|\Pr[G_0^\mathsf{B} \Rightarrow 1] - 1/2\right| = \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_\ell}(\mathsf{B}) \ . \tag{6}$$

GAME G_1. In Game G_1, we add lines 13-15, and in particular a flag QUERY in line 14, and abort (such that the game outputs an independently random bit) when QUERY is raised. QUERY is raised whenever random oracle F is queried with the vector $\mathbf{x}^*$ that was chosen during the generation of the challenge ciphertext c^*. Games G_0 and G_1 proceed identically until QUERY occurs. Hence, we have

$$\left|\Pr[G_0^\mathsf{B} \Rightarrow 1] - \Pr[G_1^\mathsf{B} \Rightarrow 1]\right| \ \leq \ \Pr[\mathrm{QUERY}] \ . \tag{7}$$

Moreover, observe that in Game G_1, B's view is independent of the bit b chosen by the game: b is only used in the computation of c_0^*, which in turn is blinded by $\mathsf{F}(\mathbf{x}^*)$. But since the game aborts (with a random output) as soon as B queries $\mathsf{F}(\mathbf{x}^*)$, this means that c_0^* is independently random in B's view. This means that also B's output b' and b are independent, which implies that the game's output $[\![b' = b]\!]$ is a uniformly random bit in case no abort occurs. But since the game also outputs a random bit upon an abort, we get that

$$\Pr[G_1^\mathsf{B} \Rightarrow 1] \ = \ 1/2. \tag{8}$$

Taking (6-8) together, we thus get

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_\ell}(\mathsf{B}) \ \leq \ \Pr[\mathrm{QUERY}] \ ,$$

and the theorem follows from the next lemma. □

Lemma 3.8 *In the situation of Game G_1, we have*

$$\Pr[\mathrm{QUERY}] \ \leq \ q_\mathsf{F}^{1/\ell} \cdot \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})$$

for an adversary A (of roughly the same complexity as Game G_1).

Proof. We may assume that $\Pr[\mathrm{QUERY}] > 0$ (so that it is possible to condition on QUERY). We describe adversary A in Figure 21.

To analyze B, let $\mathbf{x}^* := (x_1^*, \ldots, x_\ell^*)$, where $x_{i^*}^*$ is the value encrypted in A's own challenge $\widehat{c}$, and, for $i \neq i^*$, the x_i^* are defined in line 4 in Figure 21. (That is, up to decryption errors, $x_i^* = \mathsf{Dec}(sk, c_i^*)$ for all i.) Now observe that B's views in Game G_1 and in the simulation inside A are identical *until* B queries $\mathsf{F}(\mathbf{x}^*)$. In this latter case, Game G_1 would abort, while A would simply continue the simulation. In particular, if we let QUERY denote the event that B queries $\mathsf{F}(\mathbf{x}^*)$, then the probability of QUERY is the same in Game G_1 and in A's simulation. We can thus show the lemma by bounding the probability for QUERY in A's simulation.

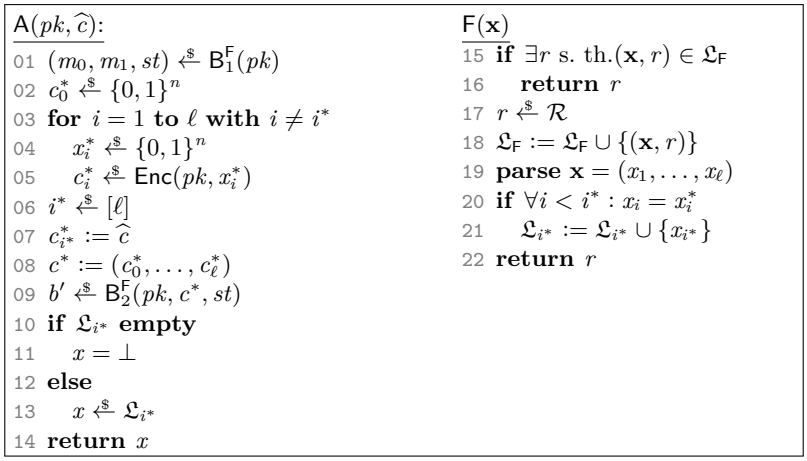


Figure 21: Adversary A against $\mathsf{IND\text{-}CPA}$ from B against $\mathsf{OW\text{-}PCA}$ for Lemma 3.8. Note that the sampling operation in line 13 refers to the *list* (not the *set*) $\mathfrak{L}_{i^*}$ (such that multiple F queries with the same x_{i^*} may raise the probability that that x_{i^*} is sampled).

To this end, for each $i \in [\ell]$, consider the probability

$$p_i \;:=\; \Pr[\, x_i = x_i^* \;\mid\; (x_1, \ldots, x_{i-1}) = (x_1^*, \ldots, x_{i-1}^*) \,\wedge\, \mathrm{QUERY}\,]$$

in an execution with A, where the probability is over a uniform choice of $\mathbf{x} = (x_1, \ldots, x_\ell)$ among the set of all of F-queries from B. (Note that the condition QUERY guarantees that at least one such $\mathbf{x}$ exists.) Intuitively, p_i denotes the probability that a F-query matches the challenge message in the i-th component when they already match in the first $i-1$ components (assuming that QUERY occurs).

It will be helpful to first note a useful property of the p_i: namely, we have

$$\prod_{i=1}^{\ell} p_i \;\overset{(i)}{=}\; \Pr[\, \mathbf{x} = \mathbf{x}^* \;\mid\; \mathrm{QUERY}\,] \;\overset{(ii)}{=}\; 1/q_{\mathsf{F}} \;, \tag{9}$$

where (i) follows by using $\Pr[A \mid B] \cdot \Pr[B] = \Pr[A \wedge B]$ for arbitrary events A, B (such that B is possible), and (ii) follows by definition of QUERY.

Furthermore, we can connect the p_i to A's output as follows. Observe that B's view in A's simulation does not depend on i^*, and thus, that the p_i do not change when conditioning on a specific choice of i^*. Now by construction of A and the list $\mathfrak{L}_{i^*}$, for each fixed choice of i^*, and assuming that QUERY occurs, we have that $x = x_{i^*}^*$ is sampled in line 13 with probability p_{i^*}. Note that in this case, A wins its own $\mathsf{OW\text{-}CPA}$ game. Formally:

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}) \;&=\; \Pr[\, \mathsf{A} \Rightarrow x_{i^*}^* \,] \;=\; \frac{1}{\ell} \sum_{i=1}^{\ell} \Pr[\, \mathsf{A} \Rightarrow x_i^* \mid i^* = i \,] \\
&=\; \frac{\Pr[\mathrm{QUERY}]}{\ell} \sum_{i=1}^{\ell} \Pr[\, \mathsf{A} \Rightarrow x_{i^*}^* \mid i^* = i \wedge \mathrm{QUERY} \,] \;=\; \frac{\Pr[\mathrm{QUERY}]}{\ell} \sum_{i=1}^{\ell} p_i \\
&\overset{(*)}{\geq}\; \Pr[\mathrm{QUERY}] \cdot \left(\prod_{i=1}^{\ell} p_i \right)^{1/\ell} \overset{(9)}{=}\; \Pr[\mathrm{QUERY}] \cdot \frac{1}{q_{\mathsf{F}}^{1/\ell}} \;,
\end{aligned}$$

where $(*)$ follows by the inequality between the arithmetic and geometric means. Rearranging terms yields the lemma. □

4 Modular FO Transformation in the QROM

In this section, we will revisit our transformations in the quantum random oracle model. In Section 4.1, we give a short primer on quantum computation and define the quantum random oracle model (QROM). In

Section 4.2, we will prove that transformation T from Figure 5 (Section 3.1) is also secure in the quantum random oracle model. Next, in Section 4.3 we will introduce $\mathsf{QU}^{\perp}_{m}$ ($\mathsf{QU}^{\not\perp}_{m}$), a variant of $\mathsf{U}^{\perp}_{m}$ ($\mathsf{U}^{\not\perp}_{m}$), which has provable security in the quantum random oracle model. Combining the two above transformations, in Section 4.4 we provide concrete bounds for the $\mathsf{IND\text{-}CCA}$ security of $\mathsf{QKEM}^{\perp}_{m} = \mathsf{QFO}^{\perp}_{m}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}']$ and $\mathsf{QKEM}^{\not\perp}_{m} = \mathsf{QFO}^{\not\perp}_{m}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}']$ in the QROM.

4.1 Quantum Computation

Qbits. For simplicity, we will treat a *qbit* as a vector $|b\rangle \in \mathbb{C}^2$, i.e., a linear combination $|b\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ of the two *basis states* (vectors) $|0\rangle$ and $|1\rangle$ with the additional requirement to the probability amplitudes $\alpha, \beta \in \mathbb{C}$ that $|\alpha|^2 + |\beta|^2 = 1$. The basis $\{|0\rangle, |1\rangle\}$ is called *standard orthonormal computational basis*. The qbit $|b\rangle$ is said to be *in superposition*. Classical bits can be interpreted as quantum bits via the mapping $(b \mapsto 1 \cdot |b\rangle + 0 \cdot |1 - b\rangle)$.

Quantum Registers. We will treat a quantum register as a collection of multiple qbits, i.e. a linear combination $\sum_{(b_1,\cdots,b_n)\in\{0,1\}^n} \alpha_{b_1\cdots b_n} \cdot |b_1 \cdots b_n\rangle$, where $\alpha_{b_1,\cdots,b_n} \in \mathbb{C}^n$, with the additional restriction that $\sum_{(b_1,\cdots,b_n)\in\{0,1\}^n} |\alpha_{b_1\cdots b_n}|^2 = 1$. As in the one-dimensional case, we call the basis $\{|b_1 \cdots b_n\rangle\}_{(b_1,\cdots,b_n)\in\{0,1\}^n}$ the *standard orthonormal computational basis*.

Measurements. Qbits can be measured with respect to a basis. In this paper, we will only consider measurements in the standard orthonormal computational basis, and denote this measurement by $\textsc{Measure}(\cdot)$, where the outcome of $\textsc{Measure}(|b\rangle)$ is a single qbit $|b\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ will be $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$, and the outcome of measuring a qbit register $\sum_{b_1,\cdots,b_n\in\{0,1\}} \alpha_{b_1\cdots b_n} \cdot |b_1 \cdots b_n\rangle$ will be $|b_1 \cdots b_n\rangle$ with probability $|\alpha_{b_1\cdots b_n}|^2$. Note that the amplitudes *collapse* during a measurement, this means that by measuring $\alpha \cdot |0\rangle + \beta \cdot |1\rangle$, α and β are switched to one of the combinations in $\{\pm(1,0), \pm(0,1)\}$. Likewise, in the n-dimensional case, all amplitudes are switched to 0 except for the one that belongs to the measurement outcome and which will be switched to 1.

Quantum oracles and quantum Adversaries. Following [BDF+11, BBC+98], we view a quantum oracle as a mapping

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathrm{O}(x)\rangle \ ,$$

where $\mathrm{O} : \{0,1\}^n \to \{0,1\}^m$, $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, and model quantum adversaries A with access to O by the sequence $U \circ \mathrm{O}$, where U is a unitary operation. We write $\mathsf{A}^{|\mathrm{O}\rangle}$ to indicate that the oracles are quantum-accessible (contrary to oracles which can only process classical bits).

Quantum random oracle model. We consider security games in the quantum random oracle model (QROM) as their counterparts in the classical random oracle model, with the difference that we consider quantum adversaries that are given **quantum** access to the random oracles involved, and **classical** access to all other oracles (e.g., plaintext checking or decapsulation oracles). Zhandry [Zha12] proved that no quantum algorithm $\mathsf{A}^{|f\rangle}$, issuing at most q quantum queries to $|f\rangle$, can distinguish between a random function $f : \{0,1\}^m \to \{0,1\}^n$ and a $2q$-wise independent function. It allows us to view quantum random oracles as polynomials of sufficient large degree. That is, we define a quantum random oracle $|\mathsf{H}\rangle$ as an oracle evaluating a random polynomial of degree $2q$ over the finite field $\mathbb{F}_{2^n}$.

Correctness of PKE in the QROM. Similar to the classical random oracle model, we need to define correctness of encryption in the quantum random oracle model. If $\mathsf{PKE} = \mathsf{PKE}^{\mathsf{G}}$ is defined relative to a random oracle $|\mathsf{G}\rangle$, then again the correctness bound might depend on the number of queries to $|\mathsf{G}\rangle$. We call a public-key encryption scheme PKE in the quantum random oracle model $\delta(q_{\mathsf{G}})$-*correct* if for all (possibly unbounded, quantum) adversaries A making at most q_{G} queries to quantum random oracle $|\mathsf{G}\rangle$, $\Pr[\mathsf{COR\text{-}QRO}^{\mathsf{A}}_{\mathsf{PKE}} \Rightarrow 1] \leq \delta(q_{\mathsf{G}})$, where the correctness game $\mathsf{COR\text{-}QRO}$ is defined as in Figure 22.

Algorithmic Oneway to Hiding. To a quantum oracle $|\mathsf{H}\rangle$ and an algorithm A (possibly with access to other oracles) we associate the following extractor algorithm $\mathsf{EXT}[\mathsf{A}, |\mathsf{H}\rangle]$ that returns a measurement x' of a randomly chosen query to $|\mathsf{H}\rangle$.

The following statement is a an algorithmic adaption of OW2H from [Unr14] and will be used heavily during our security proofs.

Lemma 4.1 *(Algorithmic Oneway to hiding (AOW2H)) Let $|\mathsf{H}\rangle : \{0,1\}^n \to \{0,1\}^m$ be a quantum random oracle, and let A be a quantum algorithm issuing at most q_{H} queries to $|\mathsf{H}\rangle$ that, on input*

GAME COR-QRO:
$(pk, sk) \leftarrow \mathsf{Gen}$
$m \leftarrow \mathsf{A}^{|\mathsf{G}\rangle}(sk, pk)$
return $[\![\mathsf{Dec}(sk, \mathsf{Enc}(pk, m; \mathsf{G}(m))) \neq m]\!]$

Figure 22: Correctness game COR-QRO for PKE_1 in the quantum random oracle model.

$\mathsf{EXT}[\mathsf{A}, |\mathsf{H}\rangle](inp)$
$i \xleftarrow{\$} [q_{\mathsf{H}}]$
Run $\mathsf{A}^{|\mathsf{H}\rangle}(inp)$ until the ith query $|\hat{x}\rangle$ to $|\mathsf{H}\rangle$
if $i >$ number of queries to $|\mathsf{H}\rangle$
  return $\perp$
else
  return $x' := \textsc{Measure}(|\hat{x}\rangle)$

Figure 23: Extractor algorithm $\mathsf{EXT}[\mathsf{A}, |\mathsf{H}\rangle](inp)$ for OW2H.

$x \in \{0,1\}^n, y \in \{0,1\}^m$ outputs either 0 or 1. Then, for all (probabilistic) algorithms F that input bit-stings in $\{0,1\}^{n+m}$ (and do not make any hash queries to $|\mathsf{H}\rangle$),

$$\left|\Pr\left[1 \leftarrow \mathsf{A}^{|\mathsf{H}\rangle}(inp) \mid x \xleftarrow{\$} \{0,1\}^n; inp \leftarrow \mathsf{F}(x, \mathsf{H}(x))\right] - \Pr\left[1 \leftarrow \mathsf{A}^{|\mathsf{H}\rangle}(inp) \mid (x,y) \xleftarrow{\$} \{0,1\}^{n+m}; inp \leftarrow \mathsf{F}(x,y)\right]\right|$$
$$\leq 2q_{\mathsf{H}} \cdot \sqrt{\Pr[x \leftarrow \mathsf{EXT}[\mathsf{A}, |\mathsf{H}\rangle](inp) \mid (x,y) \xleftarrow{\$} \{0,1\}^{n+m}; inp \leftarrow \mathsf{F}(x,y)]} \ .$$

Generic Quantum Search. For $\lambda \in [0,1]$ let B_λ be the Bernoulli distribution, i.e., $\Pr[b=1] = \lambda$ for $b \leftarrow B_\lambda$. The following lemma is a reformulation of [HRS16, Thrm. 1] whose proof is based on [Zha16].

Lemma 4.2 *(Generic Search Problem) Let $\lambda \in [0,1]$. Then, for any (unbounded, quantum) algorithm A issuing at most q quantum queries to $|f(\cdot)\rangle$, $\Pr[\mathsf{GSP}_\lambda^{\mathsf{A}} \Rightarrow 1] \leq 8 \cdot \lambda \cdot (q+1)^2$, where Game GSP_λ is defined in Figure 24.*

GAME GSP_λ
$x \leftarrow \mathsf{A}^{|f(\cdot)\rangle}$
return $f(x)$

$f(x)$
if $\exists b$ s. th. $(x,b) \in \mathfrak{L}_f$ **return** b
else
  $b \leftarrow B_\lambda$
  $\mathfrak{L}_f := \mathfrak{L}_f \cup \{(x,b)\}$
  return b

Figure 24: The generic quantum search game GSP_λ with Bernoulli parameter $\lambda \in [0,1]$.

4.2 Transformation T: from OW-CPA to OW-PCA in the QROM

Recall transformation T from Figure 5 of Section 3.1.

Lemma 4.3 *Assume PKE to be δ-correct. Then $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is δ_1-correct in the quantum random oracle model, where $\delta_1 = \delta_1(q_{\mathsf{G}}) \leq 8 \cdot (q_{\mathsf{G}}+1)^2 \cdot \delta$.*

Proof. Consider an (unbounded, quantum) adversary A in the quantum random oracle correctness game COR-QRO. For fixed (pk, sk) and message $m \in \mathcal{M}$, let

$$\mathcal{R}_{\text{bad}}(pk, sk, m) := \{r \in \mathcal{R} \mid \mathsf{Dec}(sk, \mathsf{Enc}(pk, m; r)) \neq m\}$$

denote the set of "bad" randomness. Further, define $\delta(pk, sk, m) := |\mathcal{R}_{\text{bad}}(pk, sk, m)|/|\mathcal{R}|$ as the fraction of bad randomness and $\delta(pk, sk) := \max_{m \in \mathcal{M}} \delta(pk, sk, m)$. Note that with this notation $\delta = \mathbf{E}[\delta(pk, sk)]$, where is expectation is taken over $(pk, sk) \xleftarrow{\$} \mathsf{Gen}$.

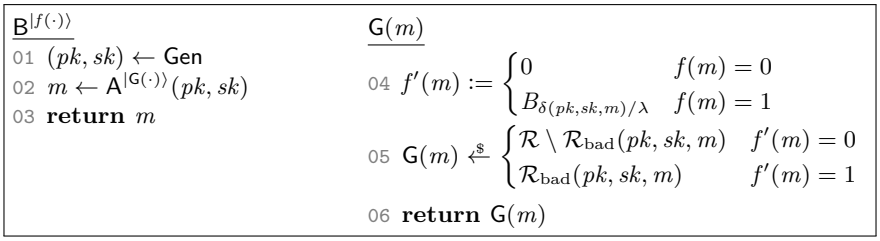


Figure 25: Adversary B executed in game $\mathsf{GSP}_{\delta(pk,sk)}$ for the proof of Theorem 4.4. We make the convention that the random variables $f'(m)$ and $\mathsf{G}(m)$ are only sampled once, i.e., once they have been assigned they are fixed.

To upper bound $\Pr[\mathsf{COR\text{-}QRO}^{\mathsf{A}} \Rightarrow 1]$, we construct an (unbounded, quantum) adversary B in Figure 25 against the generic search problem GSP_λ defined in Figure 24. B runs $(pk, sk) \xleftarrow{\$} \mathsf{Gen}$ and computes the Bernoulli parameter λ of the generic search problem as $\lambda := \delta(pk, sk) := \max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m]$.

To analyze B, we first fix (pk, sk). For each $m\mathcal{M}$, by the definition of game GSP_λ, the random variable $f(m)$ is distributed according to $B_\lambda = B_{\delta(pk,sk)}$. By construction and since $\delta(pk, sk, m) \geq \delta(pk, sk)$, $f'(m)$ defined in line 04 is distributed according to $B_{\delta(pk,sk,m)}$. Again by construction, the random variable $\mathsf{G}(m)$ defined in line 05 is uniformly distributed in $\mathcal{R}$ meaning G is a (quantum) random oracle. A wins its game $\mathsf{COR\text{-}QRO}$ iff it returns a message m such that $\mathsf{G}(m) \in \mathcal{R}_{\mathrm{bad}}(pk, sk, m)$ or, equivalently, $f'(m) = 1$. The latter condition $f'(m) = 1$ can only happen if $f(m) = 1$ in which case B wins game GSP_λ. To summarize, conditioned on a fixed (pk, sk) we obtain by Lemma 4.2

$$\Pr[\mathsf{COR\text{-}QRO}^{\mathsf{A}} \Rightarrow 1 \mid (pk, sk)] \leq \Pr[\mathsf{GSP}^{\mathsf{B}}_{\delta(pk,sk)} \Rightarrow 1] \leq 8 \cdot \delta(pk, sk) \cdot (q_{\mathsf{G}} + 1)^2 \ .$$

By averaging over $(pk, sk) \xleftarrow{\$} \mathsf{Gen}$ we finally obtain

$$\delta_1(q_{\mathsf{G}}) = \Pr[\mathsf{COR\text{-}QRO}^{\mathsf{A}} \Rightarrow 1] \leq 8 \cdot \delta \cdot (q_{\mathsf{G}} + 1)^2.$$

This completes the proof. □

The following theorem (whose proof is loosely based on [TU16]) establishes that $\mathsf{IND\text{-}PCA}$ security of PKE_1 reduces to the $\mathsf{OW\text{-}CPA}$ security of PKE, in the quantum random oracle model.

Theorem 4.4 (PKE $\mathsf{OW\text{-}CPA}$ $\overset{\mathrm{QROM}}{\Rightarrow}$ PKE_1 $\mathsf{OW\text{-}PCA}$)**.** *Assume* PKE *to be* δ*-correct. For any* $\mathsf{OW\text{-}PCA}$ *quantum adversary* B *that issues at most* q_{G} *queries to the quantum random oracle* $|\mathsf{G}\rangle$ *and* q_P *(classical) queries to the plaintext checking oracle* $\textsc{Pco}$*, there exists an* $\mathsf{OW\text{-}CPA}$ *quantum adversary* A *such that*

$$\mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}_1}(\mathsf{B}) \leq 8 \cdot \delta \cdot (q_{\mathsf{G}} + 1)^2 + (1 + 2q_{\mathsf{G}}) \cdot \sqrt{\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})} \ ,$$

and the running time of A *is about that of* B*.*

Similar to the proof of Theorem 3.1, in game G_1 the proof first implements the PCA oracle via "re-encryption". Next, we apply AOW2H to decouple the challenge ciphertext $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$ from the random oracle G. The decoupling allows for a reduction from $\mathsf{OW\text{-}CPA}$ security.

Proof. Let B be an adversary against the $\mathsf{OW\text{-}PCA}$ security of PKE_1, issuing at most q_{G} queries to $|\mathsf{G}\rangle$ and at most q_P queries to $\textsc{Pco}$. Consider the games given in Figure 26, where G is modeled as a random $2q_{\mathsf{G}}$-wise independent hash function.

$\textsc{Game}$ G_0. Since game G_0 is the original $\mathsf{OW\text{-}PCA}$ game,

$$\Pr[G^{\mathsf{B}}_0 \Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}_1}(\mathsf{B}) \ .$$

$\textsc{Game}$ G_1. In game G_1 the plaintext checking oracle $\textsc{Pco}(\cdot, \cdot)$ is replaced with a simulation that doesn't make use of the secret key anymore. We claim

$$|\Pr[G^{\mathsf{B}}_1 \Rightarrow 1] - \Pr[G^{\mathsf{B}}_0 \Rightarrow 1]| \leq 8 \cdot (q_{\mathsf{G}} + 1)^2 \cdot \delta \ . \tag{10}$$

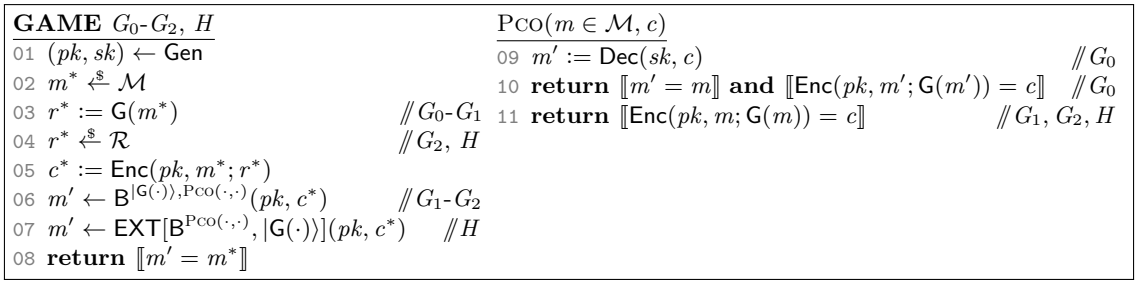


Figure 26: Games G_0, G_1, G_2, H for the proof of Theorem 4.4.

To show Equation (10), first note that both Game G_0 and Game G_1 proceed identically until the event that B submits a PCO query (m, c) such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$ and $\mathsf{Dec}(sk, c) \neq m$. We call this event BADR. Since both Game G_0 and Game G_1 proceed identically conditioned on the event that BADR does not happen,

$$|\Pr[G_1^{\mathsf{B}} \Rightarrow 1] - \Pr[G_0^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\mathrm{BADR}] .$$

Similar to the proof of Theorem 3.1 one can again show that there exists an adversary F against COR-QRO that perfectly simulates games G_0 and G_1 and wins iff BADR happens. Applying Lemma 4.3, we see that

$$\Pr[\mathrm{BADR}] \leq \Pr[\mathsf{COR\text{-}QRO}^{\mathsf{F}}] \leq 8 \cdot (q_{\mathsf{G}} + 1)^2 \cdot \delta \ .$$

GAME G_2. In game G_2, we replace $r^* := \mathsf{G}(m^*)$ with uniform randomness r^* in line 03. We apply Lemma 4.1 (AOW2H) to $x := m^*$, $y := r^*$, and algorithm F given in Figure 27. We obtain

$$|\Pr[G_2^{\mathsf{B}} \Rightarrow 1] - \Pr[G_1^{\mathsf{B}} \Rightarrow 1]| \leq 2 \cdot q_{\mathsf{G}} \cdot \sqrt{\Pr[H^{\mathsf{B}} \Rightarrow 1]} \ ,$$

where the extractor algorithm EXT of game H is defined in Figure 23.

Figure 27: Algorithm F for the application of AOW2H in the proof of Theorem 4.4.

Now that r^* is uniformly random we trivially construct an adversary C in Figure 28 against the OW-CPA security of the original encryption scheme PKE simulating game G_2 for B that outputs $m' = m^*$ if B wins in game G_2.

$$\Pr[G_2^{\mathsf{B}} \Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C}) \leq \sqrt{\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C})} \ .$$

Finally, we construct another trivial adversary D in Figure 28 against the OW-CPA security of the original encryption scheme PKE simulating game H for B with Advantage

$$\Pr[G_3^{\mathsf{B}} \Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{D}) \ .$$

Collecting the probabilities and combining adversaries C and D into one single adversary A proves the theorem. □

4.3 Transformations $\mathsf{QU}^{\perp}_m$, $\mathsf{QU}^{\not\perp}_m$

4.3.1 Transformation $\mathsf{QU}^{\perp}_m$: from OW-PCA to IND-CCA in the QROM

$\mathsf{QU}^{\perp}_m$ transforms an OW-PCA secure public-key encryption scheme into an IND-CCA secure key encapsulation mechanism with explicit rejection.

$\underline{\mathsf{C}(pk, c^*)}$	$\underline{\mathsf{D}(pk, c^*)}$
01 $m' \leftarrow \mathsf{B}^{\lvert\mathsf{G}(\cdot)\rangle, \textsc{Pco}(\cdot,\cdot)}(pk, c^*)$	03 $m' \leftarrow \mathsf{EXT}[\mathsf{B}^{\textsc{Pco}(\cdot,\cdot)}, \lvert\mathsf{G}(\cdot)\rangle](pk, c^*)$
02 **return** m'	04 **return** m'

Figure 28: Adversaries C (left) and D (right) for the proof of Theorem 4.4. Oracle $\textsc{Pco}$ is defined as in game G_2 of Figure 26.

The Construction. To a public-key encryption scheme $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ with message space $\mathcal{M} = \{0,1\}^n$, and hash functions $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}' : \{0,1\}^n \to \{0,1\}^n$, we associate $\mathsf{QKEM}^{\perp}_m = \mathsf{QU}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$. The algorithms of $\mathsf{QKEM}^{\perp}_m = (\mathsf{QGen} := \mathsf{Gen}_1, \mathsf{QEncaps}_m, \mathsf{QDecaps}^{\perp}_m)$ are defined in Figure 29. We stress that hash function H' has matching domain and range.

$\underline{\mathsf{QEncaps}_m(pk)}$	$\underline{\mathsf{QDecaps}^{\perp}_m(sk, c, d)}$
01 $m \xleftarrow{\$} \mathcal{M}$	06 $m' := \mathsf{Dec}_1(sk, c)$
02 $c \leftarrow \mathsf{Enc}_1(pk, m)$	07 **if** $m' = \perp$ **or** $\mathsf{H}'(m') \neq d$
03 $d := \mathsf{H}'(m)$	08 **return** $\perp$
04 $K := \mathsf{H}(m)$	09 **else return** $K := \mathsf{H}(m')$
05 **return** (K, c, d)	

Figure 29: IND-CCA-secure key encapsulation mechanism $\mathsf{QKEM}^{\perp}_m = \mathsf{QU}^{\perp}_m[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$.

Security. The following theorem (whose proof is again loosely based on [TU16]) establishes that IND-CCA security of $\mathsf{QKEM}^{\perp}_m$ reduces to the OW-PCA security of PKE_1, in the quantum random oracle model.

Theorem 4.5 (PKE_1 OW-PCA $\overset{\text{QROM}}{\Rightarrow}$ $\mathsf{QKEM}^{\perp}_m$ IND-CCA)**.** *If PKE_1 is δ_1-correct, so is $\mathsf{QKEM}^{\perp}_m$. For any IND-CCA quantum adversary B issuing at most q_D (classical) queries to the decapsulation oracle $\textsc{QDecaps}^{\perp}_m$, at most q_{H} queries to the quantum random oracle $\lvert\mathsf{H}\rangle$ and at most $q_{\mathsf{H}'}$ queries to the quantum random oracle $\lvert\mathsf{H}'\rangle$, there exists an OW-PCA quantum adversary A issuing $2q_D q_{\mathsf{H}'}$ queries to oracle $\textsc{Pco}$ such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{QKEM}^{\perp}_m}(\mathsf{B}) \leq (2q_{\mathsf{H}'} + q_{\mathsf{H}}) \cdot \sqrt{\mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}_1}(\mathsf{A})} \ ,$$

and the running time of A is about that of B.

Proof. Let B be an adversary against the IND-CCA security of $\mathsf{QKEM}^{\perp}_m$, issuing at most q_D queries to $\textsc{QDecaps}^{\perp}_m$, at most q_{H} queries to $\lvert\mathsf{H}\rangle$ and at most $q_{\mathsf{H}'}$ queries to $\lvert\mathsf{H}'\rangle$. Consider the games $G_{0,b}$, $G_{1,b}$, $H_{0,b}$, $H_{1,b}$ ($b \in \{0,1\}$) given in Figure 30.

GAMES $G_{0,b}$, $G_{1,b}$, $H_{0,b}$, $H_{1,b}$		$\underline{\textsc{QDecaps}^{\perp}_m((c,d) \neq (c^*,d^*))}$	// $G_{0,b}$, $G_{1,b}$, $H_{0,b}$
01 $(pk, sk) \leftarrow \mathsf{Gen}_1$		10 $m' := \mathsf{Dec}_1(sk, c)$	
02 $m^* \xleftarrow{\$} \{0,1\}^n$; $c^* \leftarrow \mathsf{Enc}_1(pk, m^*)$		11 **if** $m' \neq \perp$ **and** $\mathsf{H}'(m') = d$	
03 $K_0^* := \mathsf{H}(m^*)$; $K_1^* \xleftarrow{\$} \{0,1\}^n$		12 **return** $K := \mathsf{H}(m')$	
04 $d^* := \mathsf{H}'(m^*)$; $K^* := K_b^*$	// $G_{0,b}$	13 **else return** $\perp$	
05 $d^* \xleftarrow{\$} \{0,1\}^n$; $K^* \xleftarrow{\$} \{0,1\}^n$	// $G_{1,b}$, $H_{0,b}$, $H_{1,b}$		
06 **return** $b' \leftarrow \mathsf{B}^{\textsc{QDecaps}^{\perp}_m, \lvert\mathsf{H}\rangle, \lvert\mathsf{H}'\rangle}(pk, (c^*, d^*), K^*)$	// $G_{0,b}$, $G_{1,b}$	$\underline{\textsc{QDecaps}^{\perp}_m((c,d) \neq (c^*,d^*))}$	// $H_{1,b}$
07 $m' \xleftarrow{\$} \mathsf{EXT}[\mathsf{B}^{\textsc{QDecaps}^{\perp}_m}, \lvert\mathsf{H} \times \mathsf{H}'\rangle](pk, (c^*, d^*), K^*)$	// $H_{0,0}$, $H_{1,0}$	14 **if** $\exists m \in \mathsf{Roots}(\mathsf{H}'(X) - d)$ s.t. $\mathsf{Dec}_1(sk, c) = m$	
08 $m' \xleftarrow{\$} \mathsf{EXT}[\mathsf{B}^{\textsc{QDecaps}^{\perp}_m, \lvert\mathsf{H}\rangle}, \lvert\mathsf{H}'\rangle](pk, (c^*, d^*), K^*)$	// $H_{0,1}$, $H_{1,1}$	15 **return** $K := \mathsf{H}(m)$.	
09 **return** $[\![m' = m^*]\!]$	// $H_{0,b}$, $H_{1,b}$	16 **else return** $\perp$	

Figure 30: Games $G_{0,b}$, $G_{1,b}$, $H_{0,b}$, $H_{1,b}$ ($b \in \{0,1\}$) for the proof of Theorem 4.5.

GAMES $G_{0,b}$. Games $G_{0,0}$ and $G_{0,1}$ describe the IND-CCA game in its equivalent "left-or-right" style:

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{QKEM}^{\perp}_m}(\mathsf{B}) &= \frac{1}{2}\cdot\left|\Pr\left[\mathsf{IND\text{-}CCA}^{\mathsf{A}}\Rightarrow 0 \mid b=0\right]-\Pr\left[\mathsf{IND\text{-}CCA}^{\mathsf{A}}\Rightarrow 1 \mid b=1\right]\right| \\ &= \frac{1}{2}\left|\Pr[G^{\mathsf{B}}_{0,0}\Rightarrow 1]-\Pr[G^{\mathsf{B}}_{0,1}\Rightarrow 1]\right| \ .\end{aligned}$$

GAMES $G_{1,b}$. In games $G_{1,b}$, we replace $(d^* := \mathsf{H}'(m^*), K^* := K^*_b)$ with uniform random (d^*, K^*) in line 05. Since $G_{1,0} = G_{1,1}$, we obtain

$$|\Pr[G^{\mathsf{B}}_{0,0}\Rightarrow 1]-\Pr[G^{\mathsf{B}}_{0,1}\Rightarrow 1]| \ \leq\ |\Pr[G^{\mathsf{B}}_{0,0}\Rightarrow 1]-\Pr[G^{\mathsf{B}}_{1,0}\Rightarrow 1]|+|\Pr[G^{\mathsf{B}}_{0,1}\Rightarrow 1]-\Pr[G^{\mathsf{B}}_{1,1}\Rightarrow 1]|$$

Algorithm $\mathsf{F}_b(m^*, r^*)$
01 $(pk, sk) \leftarrow \mathsf{Gen}_1$
02 $c^* := \mathsf{Enc}_1(pk, m^*; r^*)$
03 $inp = (pk, c^*, d^*)$ // $b = 0$
04 $K^* \xleftarrow{\$} \{0,1\}^n$; $inp = (pk, c^*, d^*, K^*)$ // $b = 1$
05 **return** inp

Figure 31: Algorithms F_b for $b \in \{0,1\}$ for the application of AOW2H in the proof of Theorem 4.5.

We apply Lemma 4.1 (AOW2H) to $x := m^*$, and $y = (K^*, d^*)$ for $b = 0$ and $y = d^*$ for $b = 1$, and algorithm F_b from Figure 31 to obtain

$$\begin{aligned}|\Pr\left[G^{\mathsf{B}}_{0,0}\Rightarrow 1\right]-\Pr[G^{\mathsf{B}}_{1,0}\Rightarrow 1]| &\leq 2(q_{\mathsf{H}'}+q_{\mathsf{H}})\cdot\sqrt{\Pr[H^{\mathsf{B}}_{0,0}\Rightarrow 1]} \\ |\Pr\left[G^{\mathsf{B}}_{0,1}\Rightarrow 1\right]-\Pr[G^{\mathsf{B}}_{1,1}\Rightarrow 1]| &\leq 2q_{\mathsf{H}'}\cdot\sqrt{\Pr[H^{\mathsf{B}}_{0,1}\Rightarrow 1]} \ .\end{aligned}$$

GAME $H_{1,b}$. In games $H_{1,b}$, the oracle $\mathrm{QDECAPS}^{\perp}_m$ is changed such that it does not make use of the secret key any longer (except for line 14 by testing if $\mathsf{Dec}_1(sk, c) = m$ for given c and messages m). Recall that $\mathsf{H}' = \mathsf{H}(X)$ is a random polynomial of degree $2q_{\mathsf{H}'}$ over $\mathbb{F}_{2^n}$. Therefore, given that (c, d) is a valid encapsulation (i.e., $m' \in \mathcal{M}$ and $d = \mathsf{H}'(m')$, for $m' := \mathsf{Dec}_1(sk, c)$), m' lies within the roots of $\mathsf{H}'(X) - d$. In order to show that $\mathrm{QDECAPS}^{\perp}_m$ returns the same output in games $H_{1,b}$ and $H_{0,b}$ for every query $(c, d) \neq (c^*, d^*)$, consider the following cases, where we define $m' := \mathsf{Dec}_1(sk, c)$.

- Case 1: $\mathrm{QDECAPS}^{\perp}_m(c, d)$ returns $\perp$ in Game $H_{1,b}$, meaning that $m' \notin \mathsf{Roots}(\mathsf{H}'(X) - d)$. That latter happens iff $\mathsf{H}'(m') \neq d$ or $m' = \perp$, which is exactly the condition that $\mathrm{QDECAPS}^{\perp}_m(c, d)$ returns $\perp$ in Game $H_{0,b}$.
- Case 2: $\mathrm{QDECAPS}^{\perp}_m(c, d)$ does not return $\perp$ in Game $H_{1,b}$, meaning that $m' \in \mathsf{Roots}(\mathsf{H}'(X) - d)$ and $\mathsf{Dec}_1(sk, c) = m'$. Consequently, $\mathsf{H}'(m') = d$ and $\mathrm{QDECAPS}^{\perp}_m(c, d)$ returns $K = \mathsf{H}(m')$ in Games $H_{1,b}$. The latter is again exactly the condition that $\mathrm{QDECAPS}^{\perp}_m(c, d)$ returns $K = \mathsf{H}(m')$ in Game $H_{0,b}$.

It is easy to verify that the equivalence of $\mathrm{QDECAPS}^{\perp}_m$ in the two games follows by negation and combining both cases. We have just shown

$$\Pr[H^{\mathsf{B}}_{1,b}\Rightarrow 1] = \Pr[H^{\mathsf{B}}_{0,b}\Rightarrow 1] \ .$$

For $b \in \{0,1\}$, we trivial construct adversaries A_b against the OW-PCA security of PKE_1 simulating games $H_{1,b}$ for B as in Figure 32.

Hence,

$$\Pr[H^{\mathsf{B}}_{1,b}\Rightarrow 1] = \mathrm{Adv}^{\mathsf{OW\text{-}PCA}}_{\mathsf{PKE}_1}(\mathsf{A}_b) \ .$$

Note that both adversaries issue at most $2q_D q_{\mathsf{H}'}$ PCO-queries: For each query of B to $\mathrm{QDECAPS}^{\perp}_m$ on (c, d), both A_0 and A_1 compute the set $\mathsf{Roots}(\mathsf{H}'(X) - d)$ of complex roots, which has $2q_{\mathsf{H}'} - 1$ elements since $\mathsf{H}'(X) - d$ is a polynomial of degree $2q_{\mathsf{H}'} - 1$. In the worst case, they need to check for every element m' of $\mathsf{Roots}(\mathsf{H}'(X) - d)$ whether $\mathrm{PCO}(m', c) = 1$. Collecting the probabilities and folding adversaries A_0 and A_1 into one single adversary A proves the theorem. □

$\underline{\mathsf{A}_b^{\text{PCO}(\cdot,\cdot)}(pk, c^*)}$
$d^* \xleftarrow{\$} \{0,1\}^n; K^* \xleftarrow{\$} \{0,1\}^n$
$m' \xleftarrow{\$} \mathsf{EXT}[\mathsf{B}^{\text{QDECAPS}_m^{\perp}}|\mathsf{H} \times \mathsf{H}'\rangle](pk, c^*, d^*, K^*)$ $\quad /\!/ b = 0$
$m' \xleftarrow{\$} \mathsf{EXT}[\mathsf{B}^{\text{QDECAPS}_m^{\perp},|\mathsf{H}\rangle}, |\mathsf{H}'\rangle](pk, c^*, d^*, K^*)$ $\quad /\!/ b = 1$
return m'

Figure 32: Adversaries A_b ($b \in \{0,1\}$) against IND-PCA for the proof of Theorem 4.5. Oracle $\text{QDECAPS}_m^{\perp}(c, d)$ is defined as in game $H_{1,b}$ of Figure 30.

4.3.2 Transformation $\mathsf{QU}_m^{\not\perp}$: from OW-PCA to IND-CCA in the QROM

$\mathsf{QU}_m^{\not\perp}$ transforms an OW-PCA secure public-key encryption scheme into an IND-CCA secure key encapsulation mechanism with implicit rejection.

THE CONSTRUCTION. To a public-key encryption scheme $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ with message space $\mathcal{M} = \{0,1\}^n$, and hash functions $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}' : \{0,1\}^n \to \{0,1\}^n$, we associate $\mathsf{QKEM}_m^{\not\perp} = \mathsf{QU}_m^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}'] = (\mathsf{QGen} := \mathsf{Gen}^{\not\perp}, \mathsf{QEncaps}_m, \mathsf{QDecaps}_m^{\not\perp})$. Algorithm $\mathsf{Gen}^{\not\perp}$ is given in Figure 12 and the remaining algorithms of $\mathsf{QKEM}_m^{\not\perp}$ are defined in Figure 33. We stress again that hash function H' has matching domain and range.

$\underline{\mathsf{QEncaps}_m(pk)}$	$\underline{\mathsf{QDecaps}_m^{\not\perp}(sk' = (sk, s), c, d)}$
$m \xleftarrow{\$} \mathcal{M}$	$m' := \mathsf{Dec}_1(sk, c)$
$c \leftarrow \mathsf{Enc}_1(pk, m)$	**if** $m' = \perp$ **or** $\mathsf{H}'(m') \neq d$
$d := \mathsf{H}'(m)$	**return** $K := \mathsf{H}(s, c, d)$
$K := \mathsf{H}(m)$	**else return** $K := \mathsf{H}(m')$
return (K, c, d)	

Figure 33: IND-CCA-secure key encapsulation mechanism $\mathsf{QKEM}_m^{\not\perp} = \mathsf{QU}_m^{\not\perp}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{H}']$.

SECURITY. The following theorem establishes that IND-CCA security of $\mathsf{QKEM}_m^{\not\perp}$ reduces to the OW-PCA security of PKE_1, in the quantum random oracle model.

Theorem 4.6 (PKE_1 OW-PCA $\overset{\text{QROM}}{\Rightarrow}$ $\mathsf{QKEM}_m^{\not\perp}$ IND-CCA)**.** *If* PKE_1 *is* δ*-correct, so is* $\mathsf{QKEM}_m^{\perp}$*. For any* IND-CCA *quantum adversary* B *issuing at most* q_D *(classical) queries to the decapsulation oracle* $\text{QDECAPS}_m^{\perp}$*, at most* q_{H} *queries to the quantum random oracle* $|\mathsf{H}\rangle$ *and at most* $q_{\mathsf{H}'}$ *queries to the quantum random oracle* $|\mathsf{H}'\rangle$*, there exists an* OW-PCA *quantum adversary* A *issuing* $2q_D q_{\mathsf{H}'}$ *queries to oracle* PCO *such that*

$$\mathrm{Adv}^{\text{IND-CCA}}_{\mathsf{QKEM}_m^{\perp}}(\mathsf{B}) \leq (2q_{\mathsf{H}'} + q_{\mathsf{H}}) \cdot \sqrt{\mathrm{Adv}^{\text{OW-PCA}}_{\mathsf{PKE}_1}(\mathsf{A})} \ ,$$

and the running time of A *is about that of* B*.*

The proof is almost the same as the one of Theorem 4.5. The crucial observation is that in all games of the proof of Theorem 4.5, the simulation of $\text{QDECAPS}_m^{\perp}$ always knows if a given ciphertext (c, d) is valid or not. If it is not valid, it returns $\perp$. So for the proof of Theorem 4.6 one can simply replace $\perp$ by $\mathsf{H}(s, c, d)$. (The difference to the proof of Theorem 3.3 is the value d in the ciphertext that throughout the proof helps with the recognition of invalid ciphertexts.)

4.4 The resulting KEMs

For concreteness, we combine transformations T and $\{\mathsf{QU}_m^{\perp}, \mathsf{QU}_m^{\not\perp}\}$ from the previous sections to obtain $\mathsf{QFO}_m^{\perp} = \mathsf{T} \circ \mathsf{QU}_m^{\perp}$ and $\mathsf{QFO}_m^{\not\perp} = \mathsf{T} \circ \mathsf{QU}_m^{\not\perp}$. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M} = \{0,1\}^n$ and randomness space $\mathcal{R}$, and hash functions $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$ and $\mathsf{H}' : \{0,1\}^n \to \{0,1\}^n$, we associate

$$\begin{aligned} \mathsf{QKEM}_m^{\perp} &= \mathsf{QFO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}'] := \mathsf{QU}_m^{\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}, \mathsf{H}'] = (\mathsf{Gen}, \mathsf{QEncaps}_m, \mathsf{QDecaps}_m^{\perp}) \\ \mathsf{QKEM}_m^{\not\perp} &= \mathsf{QFO}_m^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}'] := \mathsf{QU}_m^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}, \mathsf{H}'] = (\mathsf{Gen}^{\not\perp}, \mathsf{QEncaps}_m, \mathsf{QDecaps}_m^{\not\perp}). \end{aligned}$$

Algorithm $\mathsf{Gen}^{\not\perp}$ is given in Figure 12 and the remaining algorithms are given in Figure 34.

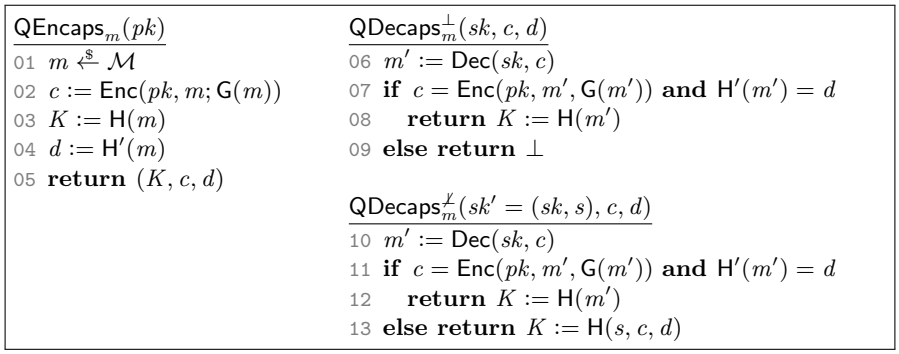


Figure 34: IND-CCA secure $\mathsf{QKEM}^{\perp}_m$ and $\mathsf{QKEM}^{\not\perp}_m$ obtained from PKE.

The following table provides (simplified) concrete bounds of the IND-CCA security of $\mathsf{KEM} \in \{\mathsf{QKEM}^{\not\perp}_m, \mathsf{QKEM}^{\perp}_m\}$ in the quantum random oracle model, directly obtained by combining Theorems 4.4–4.6. Here $q_{\mathsf{RO}} := q_{\mathsf{G}} + q_{\mathsf{H}}$ counts the total number of B's queries to the quantum random oracles G and H and q_D counts the number of B's (classical) decryption queries.

KEM	Concrete bound on $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{B}) \leq$
$\mathsf{QKEM}^{\not\perp}_m$, $\mathsf{QKEM}^{\perp}_m$	$8q_{\mathsf{RO}}\sqrt{\delta \cdot q_{\mathsf{RO}}^2 + q_{\mathsf{RO}} \cdot \sqrt{\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A})}}$

Acknowledgments

We would like to thank Andreas Hülsing, Christian Schaffner, and Dominique Unruh for interesting discussions on the FO transformation in the QROM. We are also grateful to Krzysztof Pietrzak and Victor Shoup for discussions on Section 3.4.

References

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 143–158. Springer, Heidelberg, April 2001. (Cited on page 2, 6.)

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016. (Cited on page 3.)

[AOP+17] Martin R. Albrecht, Emmanuela Orsini, Kenneth G. Paterson, Guy Peer, and Nigel P. Smart. Tightly secure ring-lwe based key encapsulation with short ciphertexts. Cryptology ePrint Archive, Report 2017/354, 2017. http://eprint.iacr.org/2017/354. (Cited on page 7.)

[BBC+98] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th FOCS*, pages 352–361. IEEE Computer Society Press, November 1998. (Cited on page 25.)

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 1006–1018. ACM Press, October 2016. (Cited on page 3.)

[BCLvV16] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime. Cryptology ePrint Archive, Report 2016/461, 2016. http://eprint.iacr.org/2016/461. (Cited on page 3.)

[BCNS15] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570. IEEE Computer Society Press, May 2015. (Cited on page 3.)

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. (Cited on page 3, 25.)

[BDK+17] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. Crystals – kyber: a cca-secure module-lattice-based kem. Cryptology ePrint Archive, Report 2017/634, 2017. http://eprint.iacr.org/2017/634. (Cited on page 3.)

[BLK00] Joonsang Baek, Byoungcheon Lee, and Kwangjo Kim. Secure length-saving ElGamal encryption under the computational Diffie-Hellman assumption. In Ed Dawson, Andrew Clark, and Colin Boyd, editors, *ACISP 00*, volume 1841 of *LNCS*, pages 49–58. Springer, Heidelberg, July 2000. (Cited on page 6.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. (Cited on page 2.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006. (Cited on page 7.)

[BV17] Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 592–606. Springer, Heidelberg, May 2017. (Cited on page 6, 7.)

[CHJ+02] Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. GEM: A generic chosen-ciphertext secure encryption method. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 263–276. Springer, Heidelberg, February 2002. (Cited on page 2, 6.)

[CKLS16] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. Lizard: Cut off the tail! practical post-quantum public-key encryption from lwe and lwr. Cryptology ePrint Archive, Report 2016/1126, 2016. http://eprint.iacr.org/2016/1126. (Cited on page 3.)

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 127–145. Springer, Heidelberg, April 2008. (Cited on page 7.)

[CKS09] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. *Journal of Cryptology*, 22(4):470–504, October 2009. (Cited on page 7.)

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. (Cited on page 3.)

[Den03] Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *LNCS*, pages 133–151. Springer, Heidelberg, December 2003. (Cited on page 2, 4, 5, 6.)

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 342–360. Springer, Heidelberg, May 2004. (Cited on page 6, 7.)

[DXL12] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. http://eprint.iacr.org/2012/688. (Cited on page 3.)

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, August 1999. (Cited on page 2, 3, 6.)

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. (Cited on page 2, 3, 8.)

[GMMV05] David Galindo, Sebastià Martín, Paz Morillo, and Jorge L. Villar. Fujisaki-okamoto hybrid encryption revisited. *Int. J. Inf. Sec.*, 4(4):228–241, 2005. (Cited on page 6.)

[HGSW05] Nick Howgrave-Graham, Joseph H. Silverman, and William Whyte. Choosing parameter sets forwithand. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 118–135. Springer, Heidelberg, February 2005. (Cited on page 3.)

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Heidelberg, March 2016. (Cited on page 26.)

[KML03] Eike Kiltz and John Malone-Lee. A general construction of IND-CCA2 secure public key encryption. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *LNCS*, pages 152–166. Springer, Heidelberg, December 2003. (Cited on page 6.)

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May 2010. (Cited on page 7.)

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Heidelberg, May 2013. (Cited on page 6.)

[NIS17] NIST. National institute for standards and technology. postquantum crypto project, 2017. http://csrc.nist.gov/groups/ST/post-quantum-crypto/. (Cited on page 3.)

[OP01] Tatsuaki Okamoto and David Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 159–175. Springer, Heidelberg, April 2001. (Cited on page 2, 6.)

[Pei14] Chris Peikert. Lattice cryptography for the internet. Cryptology ePrint Archive, Report 2014/070, 2014. http://eprint.iacr.org/2014/070. (Cited on page 3.)

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. (Cited on page 6.)

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992. (Cited on page 2.)

[Sho04a] Victor Shoup. ISO 18033-2: An emerging standard for public-key encryption. http://shoup.net/iso/std6.pdf, December 2004. Final Committee Draft. (Cited on page 6.)

[Sho04b] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. http://eprint.iacr.org/2004/332. (Cited on page 7, 11.)

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the fujisaki-okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016. (Cited on page 3, 5, 6, 27, 29.)

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, Heidelberg, May 2014. (Cited on page 25.)

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 755–784. Springer, Heidelberg, April 2015. (Cited on page 5.)

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012. (Cited on page 25.)

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508. Springer, Heidelberg, August 2016. (Cited on page 26.)