mathematics

MDPI

Article

Analysis of the FO Transformation in the Lattice-Based Post-Quantum Algorithms

Miguel Ángel González de la Torre [1,*], Luis Hernández Encinas [1] and Araceli Queiruga-Dios [2]

[1] Instituto de Tecnologías Físicas y de la Información (ITEFI), Consejo Superior de Investigaciones Científicas (CSIC), 28006 Madrid, Spain
[2] Departamento de Matemática Aplicada, Universidad de Salamanca (USAL), 37008 Salamanca, Spain
* Correspondence: ma.gonzalez@csic.es

Abstract: Newer variants of the Fujisaki–Okamoto transformation are used in most candidates of the third round of the NIST Post-Quantum Cryptography standardization call in the category of public key encryption schemes. These transformations are applied to obtain a highly secure key encapsulation mechanism from a less secure public key encryption scheme. Furthermore, there are five candidates (three finalists and two alternatives) that passed to the third round of the process and whose security is based in lattice problems. This work analyzes the different ways in which the lattice-based candidates of the NIST call apply the Fujisaki–Okamoto transformation and the particularities of each application. The study of such differences and their repercussion in the design of the proposals will allow a better understanding of the algorithms. Moreover, we propose a modification of the Kyber algorithm—the only public key encryption candidate established as a PQC standard by NIST in its more recent publication—in order to avoid the re-encryption in the decapsulation algorithm and, in this way, to reduce the side channel attacks vulnerability.

Keywords: Fujisaki–Okamoto transformation; key encapsulation mechanism; lattice-based cryptography; post-quantum cryptography; public key encryption

MSC: 11T71; 68P25; 81P94; 94A60

check for updates

Citation: González de la Torre, M.Á.; Hernández Encinas, L.; Queiruga-Dios, A. Analysis of the FO Transformation in the Lattice-Based Post-Quantum Algorithms. *Mathematics* **2022**, *10*, 2967. https://doi.org/10.3390/math10162967

Academic Editors: Antanas Cenys, Ximeng Liu and Jonathan Blackledge

Received: 28 June 2022
Accepted: 15 August 2022
Published: 17 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Nowadays, no one doubts that the increase in computing power of quantum computers is a real threat for current asymmetric encryption systems and protocols. Shor's algorithms [1] and quantum computers with enough computing power will break the asymmetric encryption schemes whose security is based on the integer factorization or discrete logarithm problems.

Post-Quantum Cryptography (PQC) is quickly becoming a greater need; however, there are still several flaws in the general knowledge of how the current PQ schemes work. The Fujisaki–Okamoto (FO) transformation (with its particular variances) defines the structure of the final scheme and its final semantic security. In fact, there are other factors affected by the design choices made in the application of the transformation, such as vulnerability to side-channel attacks or the anonymity of the cryptosystem. The National Institute of Standards and Technology (NIST) launched an international call to select PQC standards, that is, sufficiently secure algorithms for resisting quantum computers attacks. This call affects to two different categories or asymmetric protocols types: Public Key Encryption (PKE)/Key Encapsulation Mechanism (KEM) and digital signature schemes.

In all the lattice-based candidates in the PKE/KEM schemes category that were in the third round of the NIST call, a variant of the Fujisaki–Okamoto transformation is used in order to obtain a highly secure KEM from a less secure PKE scheme. For this reason, it is important to analyze the different ways the lattice-based candidates of this round of NIST call apply the FO transformation and the particularities of each application. On the contrary,

the finalist and alternative proposals related to digital signatures in such a round do not use the FO transformation; hence, they are not part of this study. In our knowledge, an analysis showing such differences and their repercussion in the design of the lattice-proposals (NTRU, SABER, FrodoKEM, CRYSTALS-Kyber, and NTRUPrime) has not been made. We can just mention the work [2], where the authors analyzed the FO transformation for the code-based algorithms of the NIST call.

Public key encryption or asymmetric encryption is commonly used to develop hybrid schemes, which means schemes that use symmetric and asymmetric encryption. In hybrid schemes, the asymmetric scheme is used to perform a key exchange, and the symmetric scheme aims to cypher a message (this process may take two steps or just only one step for key exchange and cypher processes). Keeping in mind that symmetric encryption will withstand the quantum threat better than the asymmetric counterpart, the attention of the NIST standardization process focuses on public key schemes specialized in key exchanges, which are called KEM. Newer versions of the FO transformation consist of particular KEM constructions in which a PKE with a lower security assumption is used as the central part of the scheme.

The FO transformation was introduced by Fujisaki and Okamoto in 1999 [3]. Originally, this transformation used a PKE, a Symmetric Key Encryption scheme (SKE), and hash functions to define a hybrid encryption scheme [4], with a security reduction of the security of the PKE and the SKE. Improved versions of the FO transformation have been proposed over the years, and they are employed to design highly secure public key encryption schemes. Among others, Hofheinz et al. [5] proposed the PKE/KEM transformation and some variants in order to provide tighter security proofs. They also considered a new transformation that is secure under a quantum model. Other publications (see [6,7]) studied the quantum security of the transformations proposed by Hofheinz et al. In any case, all candidates included in the third round of the NIST call, in the PKE/KEM category, use the FO transformation.

The objective of this work is to analyze and to study the use of the different versions of the FO transformation in the NIST third round lattice-based finalist and alternative proposals. Moreover, as there are several side-channel attacks against such proposals, mainly based on the re-encryption process, we propose a modification of Kyber algorithm, as an example, to reduce the efficiency of such attacks.

The rest of this paper is organized in the following way. In Section 2, the theoretical background of lattices problems is introduced. The main side-channel attacks against lattice-based KEM proposals are shown in Section 3. Section 4 contains the definition of the different FO transformations and the security reduction of each one. In Section 5, the lattice-based algorithms that are the finalist or alternatives in the PKE/KEM category of the NIST call are studied. This study is focused on the variant of the FO transformation that is applied and how it is applied. Section 6 introduces a structure, in particular devised for Kyber algorithm, intended to protect the cryptosystem against side-channel attacks exploiting the use of re-encryption. Finally, in Section 7, the applications of the FO transformation are analyzed and conclusions are drawn on the conducted study.

2. Theoretical Notions

2.1. Notation

In this work, we will consider that a Probabilistic PKE (PPKE) is a set, π, composed by three algorithms: key generation, $\mathcal{G}'$, encryption, $\mathcal{E}$, and decryption, $\mathcal{D}$, and a couple of sets M and C, where M is the set of possible messages and C is an optional randomness set. In short, $\pi = \{\mathcal{G}', \mathcal{E}, \mathcal{D}, M, C\}$. In the particular case when the PKE is deterministic (DPKE), then C is not considered. Moreover, M can be omitted if it is not necessary to specify it.

A KEM is a set made of three algorithms: key generation, $\mathcal{G}$, encapsulation, $\mathcal{E}c$, and decapsulation, $\mathcal{D}c$. We will denote this set by $\kappa = \{\mathcal{G}, \mathcal{E}c, \mathcal{D}c\}$.

The correctness of a PKE is defined as the probability of generating invalid ciphertexts, that is, ciphertexts obtained by the encryption algorithm so that if the decryption algorithm

takes them as input, the decryption outputs give an error (say $\perp$). We say that a given PKE, $\pi = \{\mathcal{G}', \mathcal{E}, \mathcal{D}, M\}$, is perfectly correct if for any pair of public and secret keys, (pk, sk), generated by $\mathcal{G}'$, for any message, $m \in M$, and c defined as $c := \mathcal{E}(pk, m)$, then

$$Pr[\mathcal{D}(sk, c) = m | c = \mathcal{E}(pk, m)] = 1.$$

A PKE π is said to be γ-spread if for every pair of public-secret keys, $(pk, sk) \leftarrow \mathcal{G}'$, and every message $m \leftarrow M$, it is verified that the image of $\mathcal{E}$ is sufficiently random, that is,

$$\gamma(pk, m) = -\log \max_{c \in Im(\mathcal{E})} \Pr_{r \leftarrow_R C}[c = \mathcal{E}(pk, m, r)] \geq \gamma.$$

2.2. Some Security Aspects

The main reason to apply the FO transformation is to obtain schemes that provide a strong notion of security, starting from a weaker one.

One-Way Encryption:

The One-Wayness (OW) notion of security is frequently seen as a weak definition of asymmetric encryption security. We consider the PKE $\pi = \{\mathcal{G}, \mathcal{E}, \mathcal{D}, M, C\}$ and let A be an adversary against π. This adversary receives an encrypted message $c = \mathcal{E}(pk, m)$; then, he has to make a guess on the original plaintext, and the attacker can consult an oracle to make the guess. Depending on the oracle O^A, the attack is defined in a different way. The advantage of the attacker is defined in the same way, independently of the oracle, and is denoted as $OW - *$. For $k \in \mathbb{N}$, the advantage of A is defined as

$$Adv^{\text{OW-}*}_{A,\pi,M}(k) = Pr\Big[(pk, sk) \leftarrow \mathcal{G}, c \leftarrow \mathcal{E}(pk, m; r) \colon A^{O^A}(c, pk) = m\Big]$$

The possible oracles that can be considered provide in the following attacks:

- If A has no oracle, then the attack is defined as a *One-Way-Chosen-Plaintext Attack* (OW-CPA).
- If O^A is a Plaintext checking oracle (PCO), then A is defined as a *One-Way-Plaintext-Checking-Attack* (OW-PCA). This oracle works as follows: if $\mathcal{D}(sk, c') = m'$ then $\text{PCO}(m', c') = 1$; else $\text{PCO}(m', c') = 0$.
- If O^A is a Ciphertext validation oracle (CVO), then A is defined as a *One-Way-Validation-Attack* (OW-VA). This oracle works as follows: CVO takes c as input and calculates $m^* \leftarrow \mathcal{D}(sk, c)$; if $m^* \in M$ returns 1; else returns 0.
- If O^A encompasses both a plaintext checking oracle and a validation oracle, then A is defined as a *One-Way-Plaintext-Checking-Validation-Attack* (OW-PCVA).

The adversary, A, is not allowed to directly ask the oracles about the plaintext m or the ciphertext c.

Definition 1. *Let $ATK \in \{CPA, PCA, VA, PCVA\}$. A PKE, denoted by π, is said to be (ϵ, t, q)-secure in the OW-ATK sense if for all OW-ATK adversaries, A, which runs in time at most t and makes at most q queries to an O^A oracle, has*

$$Adv^{OW\text{-}ATK}_{\pi}(A) \leq \epsilon.$$

Strong Security Notions:

INDistinguishability under Chosen Ciphertext Attacks (IND-CCA) was established as the target for semantic security by the NIST. Here, we introduce the formal definition of IND-CCA and Indistinguishability under Chosen Plaintext Attacks in the Random Oracle Model (ROM).

Let $A = (A_1, A_2)$ be an adversary against a PKE, π, that behaves as follows. First of all, a key pair $(pk, sk) \leftarrow \mathcal{G}$ is generated and it is set as a random value $b \leftarrow_R \{0, 1\}$ (both

the keys and b are unknown for A). A_1 (A finds the way) takes the public key as input and generates two valid plaintexts, m_0, m_1, and a value s

$$A_1(pk) = (m_0, m_1, s).$$

To generate these outputs, A_1 can query two random oracles, H and G. Set $c = \mathcal{E}(pk, m_b)$ (remember that b is still unknown to the attacker). A_2 (A guesses the way) takes s and c as input and makes queries to the random oracles until it is able to make a guess and outputs $b' \in \{0,1\}$. The adversary is successful if $b' = b$. The advantage of A, as an IND-CPA adversary, is defined as follows:

$$Adv_{\pi}^{\text{IND-CPA}}(A) = 2 \cdot Pr\left[\begin{array}{c} G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{G}; \\ (m_0, m_1, s) \leftarrow A_1^{G,H}(pk); b \leftarrow_R \{0,1\}; \\ c \leftarrow \mathcal{E}(pk, m_b) \colon A_2^{G,H}(s, c) = b \end{array}\right] - 1.$$

Definition 2. *An adversary A (t, q_g, q_h, ϵ)-breaks π in the sense of IND-CPA in the ROM if A runs in at most time t, asks at most q_g queries to G, asks at most q_h queries to H, and achieves $Adv_{\pi}^{IND\text{-}CPA}(A) \leq \epsilon$. An encryption scheme, π, is (t, q_g, q_h, ϵ)-secure in the IND-CPA sense if there is no adversary that breaks it in that sense.*

A stronger security assumption than IND-CPA is INDistinguishability under Chosen Ciphertext Attacks. This definition of security can be defined for a general public key encryption scheme. However, since in this work, this definition is only used on KEMs, we presented here the KEM version (that slightly differs from the PKE one). In this case, access is given to the adversary to a decryption oracle, in addition to the other oracles that are the same as in IND-CPA. A decryption oracle is an oracle that takes any ciphertext and decrypts it, but it cannot take the challenged ciphertext as a valid input. The advantage for IND-CCA security is defined as follows.

$$Adv_{\pi}^{\text{IND-CCA}}(A) = 2 \cdot Pr\left[\begin{array}{c} G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{G}; b \leftarrow_R \{0,1\}; \\ (K_0, c^*) \leftarrow Encaps(pk); K_1 \leftarrow_R \mathsf{K} \colon A^{G,H,\mathcal{D}_{sk}}(s, c) = b \end{array}\right] - 1.$$

The final security notion considered is called Disjoint Simulatability (DS). Let D_M be a distribution over the message space, M, of a deterministic PKE. Then, the DPKE scheme is D_M disjoint simulatable if the ciphertext of a message that is distributed according to D_M can be simulated by a simulator that does not know the message, and the simulated ciphertext is invalid (i.e., it does not belong to the image of the encryption algorithm) with overwhelming probability [6].

Generally speaking, to prove the security of a primitive, say P, under the hardness of a given problem denoted by S, a reduction algorithm, called R, is constructed, which uses an adversary, A, against the security of P as a subroutine and can solve the problem S [6]. If (t, r) and (t', r') denote the running time and success rate, respectively, of A and R, it is said that a reduction is tight [8] if $t \approx t'$ and $r \approx r'$. Tight security guarantees that to break the security of the primitive P implies to break the problem S. Moreover, if a security reduction is non-tight, P is not guaranteed to be hard to break even when S is [7]. Usually, a parameter adjustment is needed to maintain the correct security reduction.

2.3. Lattice-Based Problems

Lattice-based cryptography has proven to be one of the most promising mathematical backgrounds to post-quantum algorithms. There are two classical problems used in lattice-based cryptography, the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). However, most of the NIST call candidates base their security in the Learning With Errors (LWE) problem.

Given a lattice L, the LWE problem can be stated as follows: given pairs $(\mathbf{a}_i, b_i)$, such that $\mathbf{a}_i \leftarrow_R L$ and $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$, where $e_i \leftarrow_R \chi$ is an error, sampled by an error distribu-

tion χ (Gaussian, binomial, etc.), the goal is to find the secret vector $\mathbf{s} \in L$. The notation $\leftarrow_R$ denotes that the elements are chosen uniformly at random. In general, if no algebraic structure on the lattice is considered, then $L = \mathbb{Z}_q^n$. The objective of the problem is to determine the vector $\mathbf{s}$ from several samples such as the following ones:

$$\begin{aligned} \mathbf{a}_1 &\in \mathbb{Z}_q^n, \quad b_1 = \langle \mathbf{s}, \mathbf{a}_1 \rangle + e_1, \\ \mathbf{a}_2 &\in \mathbb{Z}_q^n, \quad b_2 = \langle \mathbf{s}, \mathbf{a}_2 \rangle + e_2, \\ &\vdots \\ \mathbf{a}_r &\in \mathbb{Z}_q^n, \quad b_r = \langle \mathbf{s}, \mathbf{a}_r \rangle + e_r. \end{aligned}$$

If the error e_i is not added to the inner product of $\mathbf{s}$ and $\mathbf{a_i}$, then $\mathbf{s}$ can be recovered efficiently by the Gaussian elimination method in the expression $\mathbf{b} = \mathbf{As}$, where $\mathbf{A}$ is the matrix of vectors $\mathbf{a}_i$.

Given a ring, $\mathcal{R}$, the Ring LWE problem (RLWE) is the same problem defined above, but now, $(a, b) \in \mathcal{R} \times \mathcal{R}$. In general, the considered ring is $\mathcal{R} = \mathcal{R}_q = \mathbb{Z}_q[x]/(x^n \pm 1)$. The structure of the ring allows simpler computation with smaller keys; however, it may also have a higher vulnerability to attacks that can exploit such a structure. Moreover, the Module LWE (MLWE) problem is analogous to the RLWE problem one but considering a module structure instead of a ring structure. Finally, the Module Learning With Rounding (MLWR) problem is a variant of the MLWE in which the small error terms are determined from the beginning, instead of sampled, and this error is avoided by rounding from one modulus to a smaller one.

3. Side-Channel Attacks

As mentioned above, many KEMs constructions are based on the FO transformation, which allows building a CCA secure scheme from a CPA secure PKE scheme. The FO transformation decrypts the ciphertext with the PKE to retrieve the message m. Later, it re-encrypts m, in a deterministic way, to obtain a ciphertext c. In this way, any ciphertext, c', that is invalid, i.e., not generated by the CPA PKE scheme, will lead to a ciphertext c such that $c \neq c'$ with a negligible probability. In the case of having an invalid ciphertext, the CCA KEM will return either a random message that cannot be used by the adversary or simply and error message, say $\perp$. It has been proved [3,5–7,9] that the FO transformation is secure from a mathematical point of view, but several works have shown that this is not completely true when side-channel attacks are considered. These physical attacks exploit the leakage of intermediate computations to circumvent the mathematical security of the implementation. Briefly, an adversary can perform a CCA (chosen-ciphertext attack) against the part of the scheme that is only CPA secure.

For example, Ravi et al. in [10] demonstrated generic and practical electromagnetic (EM) side-channel assisted chosen ciphertext attacks over LWE- and Learning With Rounding-based PKE and KEM secure in the IND-CCA security model. They showed that such side-channel information can be efficiently used to instantiate a plaintext checking oracle, which provides binary information about the output of the decryption algorithm, which is typically concealed within IND-CCA secure PKE/KEMs.

Moreover, in [11], a side-channel attack on a first-order masked implementation of IND-CCA secure Saber KEM is presented. The authors showed how to recover both the session key and the long-term secret key from 24 traces using a deep neural network created at the profiling stage.

Two security games related to finding decryption failures were introduced in [12]. The first game consists of capturing the computationally hard task of using the public key to find a decryption failure, and the second one is to capture the statistically hard task of searching the random oracle for key-independent failures.

In [13], the authors presented a side-channel analysis against KEMs based on the FO transformation. In this case, the attack exploits a side-channel leakage that happens when the re-encryption is executed in the decapsulation of the KEM. In particular, the attack

studied the use of a pseudo-random function during the re-encryption. The leakage is used as a plaintext-checking oracle that tells whether the decryption result is equivalent to the reference plaintext. Due to the generality and practicality of such an oracle, the proposed attack can attain a full key recovery of various KEMs where an active attack on the underlying PKE is known. Ueno et al. demonstrated that the proposed attack can achieve a full key recovery on most NIST PQC third-round PKE/KEM candidates. The algorithms in the scope of these results are Kyber, Saber, FrodoKEM, NTRU Prime, NTRU, HQC, BIKE, and SIKE.

Xagawa et al. in [14] investigated all NIST PQC round 3 KEM candidates from the fault-injection attacks viewpoint: Classic McEliece, Kyber, NTRU, Saber, BIKE, FrodoKEM, HQC, NTRU Prime, and SIKE. As we know, all KEM schemes use variants of the FO transformation, so the equality test of re-encryption in decapsulation is critical. In fact, they surveyed effective key-recovery attacks when the equality test can be skipped. Moreover, as the open-source pqm4 library contains most KEM schemes (except Classic McEliece and HQC), they show that a single instruction-skipping fault in the decapsulation processes leads to skipping the equality test virtually for Kyber, NTRU, Saber, BIKE, and SIKE.

Xu et al. proposed in [15] adaptive EM side-channel attacks with carefully constructed ciphertexts on Kyber, and they demonstrated that specially chosen ciphertexts allow an adversary to modulate the leakage of a target device and enable full key extraction with a small number of traces through simple power analysis.

Another attack against Kyber was proposed in [16]. In this paper, the authors showed that it is possible to combine fault injections with the objective of mounting chosen-ciphertext attacks. In fact, they presented an attack on Kyber which combines ciphertext manipulation by flipping a single bit of an otherwise valid ciphertext with a fault that corrects the ciphertext during the decapsulation process. Later, Hermelink et al. used the Fujisaki–Okamoto transformation as an oracle and obtained inequalities involving secret data, from which they could recover the private key.

More recently, Azouaoui considered the case of Kyber as an example of Simple Power Analysis (SPA) against the re-encryption of schemes using the FO transformation and the Differential Power Analysis (DPA) against the decryption, with shortcut formulas in order to compare their strengths in function of the noise level [17]. They evaluated the cost of preventing them with masking and later discuss tweaks to improve the situation and enable a better leveling of the countermeasures. They concluded that current solutions for securing KEM (such as Kyber) are unlikely to be efficient in low-noise settings without improvements.

From the previous results, it is clear that some post-quantum KEMs are vulnerable to Chosen-Ciphertext Side-Channel Attacks (CC-SCA). These attacks target the re-encryption step in the FO transformation. To sufficiently protect PQC KEMs on embedded devices against such CC-SCA, masking at increasingly higher order is required, which induces a considerable overhead.

In [18], the authors proposed the use of a conceptually simple construction, the Encrypt-then-Sign ($\mathcal{E}t\mathcal{S}$) KEM that reduces the impact of CC-SCA. This construction uses the paradigm introduced by Zheng et al. [19] and analyzed by An et al. [20], and it instantiates a post-quantum authenticated KEM in the outsider-security model. In particular, Azouaoui et al. showed that a CC-SCA-protected $\mathcal{E}t\mathcal{S}$ KEM version of Kyber requires less than 10% of the cycles required for the CC-SCA-protected FO-based KEM, with the cost of additional data/communication overhead.

4. FO-like Transformations

Several results and optimizations of the FO transformation have been proposed since the publication of the seminal paper by Fujisaky and Okamoto [3]. Thus, Dent introduced KEM constructions similar to those used in PQC [9]. Although Dent results are perfectly valid today and also present tight security reductions in the ROM, those transformations are only applicable to a deterministic and perfectly correct PKE. This was also a requirement

for the initial FO transformation in [3,21], where another version of the transformation, named REAC/GEM, was described.

Probably, the most influential results in post-quantum cryptography regarding the FO transformation are the results of Hofheinz et al. [5]. In this paper, a generalization of Dent's transformations was defined, drooping the imposition of deterministic and perfectly correct PKE in some cases. Another feature of the transformations proposed in [5] is that they highlight the need for tight security proofs, which are provided in the ROM. Finally, in the article by Hofheinz et al., a transformation with a security reduction in the Quantum Random Oracle Model (QROM) was defined; however, the security reduction is non-tight.

Later works, such as the one by Jiang et al. [7] and Saito et al. [6], gave proofs of quantum resistance for certain transformations provided in [5], without considering some of the additional structure that Hofheinz et al. applied in their work.

4.1. Obtaining a KEM from a DPKE

Given a DPKE, $\pi = \{\mathcal{G}', \mathcal{E}, \mathcal{D}\}$, the scheme followed to construct a KEM, $\kappa = \{\mathcal{G}, \mathcal{E}c, \mathcal{D}c\}$, has been known since 2003 [9]. In this construction, the key generation algorithm is the same for the DPKE and the KEM, that is, $\mathcal{G}' = \mathcal{G}$. The encapsulation algorithm, $\mathcal{E}c$, generates a random plaintext, encrypts it by means of the encryption algorithm of the DPKE, $\mathcal{E}$, and considers the output of a hash function (or a key derivation function) of the chosen plaintext as a shared secret. Finally, the decapsulation algorithm, $\mathcal{D}c$, considers the ciphertext, decrypts it by using the decryption algorithm of the DPKE, $\mathcal{D}$, and generates the same shared secret. In this scheme, different intermediate steps are considered to obtain strong and tight security reductions. This generic scheme is shown in Table 1, and it is called the U transformation.

Table 1. Generic construction of a KEM from a DPKE (U transformation).

$\mathcal{E}c(pk)$	$\mathcal{D}c(sk, c)$
$m \leftarrow_R M$	$m' \leftarrow \mathcal{D}(sk, c)$
$c \leftarrow \mathcal{E}(pk, m)$	If $c \neq \mathcal{E}(pk, m')$ then
$K = H(m)$	return $\perp$
return (K, c)	else return $K = H(m')$

4.2. Obtaining a DPKE from a PPKE

If a PPKE is given, it is possible to transform it by means of the so-called T transformation [5,9] into a DPKE. In this way, one can use the previous generic construction to obtain a KEM from a PPKE by just using the T transformation.

If we consider a PPKE, π, and a hash function, G, then the T transformation is defined as follows:

$$\mathsf{T}[\pi, G] = \pi^{\tau} = \{\mathcal{G}^{\tau}, \mathcal{E}^{\tau}, \mathcal{D}^{\tau}\},$$

which gives the following DPKE: the key generation algorithm is the same for both PKEs, that is, $\mathcal{G}^{\tau} = \mathcal{G}$. The new encryption algorithm, $\mathcal{E}^{\tau}$, is defined by $\mathcal{E}^{\tau}(pk, m) = \mathcal{E}(pk, m, G(m)) = c$, where $G(m)$ plays the role of random coins for $\mathcal{E}$ in PPKE. Moreover, the new decryption algorithm, $\mathcal{D}^{\tau}$, decrypts a message in the same way that the original one, i.e., $m' = \mathcal{D}(sk, c)$ and checks if $m' = \mathcal{E}(pk, m', G(m'))$ (this computation is a re-encryption). If the response is positive, the decryption outputs $m' = m$ and returns error (denoted by $\perp$) in the other case. Table 2 contains the security reductions of the T transformation depending on the underlying security and provides information about the tightness of the security reductions in [5].

The T transformation has a key role in latticed-based cryptography, since several such algorithms rely on it to be deterministic and have the security required to apply a certain KEM construction. However, as was described before, this transformation introduced a re-encryption component.

Table 2. Security reductions proven in [5] for the T transformation.

Security	QROM	Tightness	Requirements
T: OW-CPA ⇒ OW-PCA	✓	–	none
T: IND-CPA ⇒ OW-PCA	✓	✓	none
T: OW-CPA ⇒ OW-PCVA	✓	–	γ-spread
T: IND-CPA ⇒ OW-PCVA	–	✓	γ-spread

4.3. Implicit and Explicit Rejection

The schemes proposed by Dent [9] were constructed with what is called "explicit rejection", which is denoted by using the exponent $^{\perp}$. This notion is related to the way in which the KEM deals with either errors or invalid ciphertext when the encryption or decryption algorithms of the PKE are used. In Table 1, one can see how the decryption algorithm is executed during the decapsulation. In fact, let us suppose that the output is $\perp$, which means there has been an error. If the transformation is designed with "explicit rejection", then the decapsulation automatically outputs $\perp$. On the contrary, if the transformation is considered with "implicit rejection" (they are denoted by using the exponent $^{\not\perp}$), the decapsulation outputs $H(s, c)$, which is the hash of a randomly chosen string s (in general, s is chosen from the message set; it is defined in the key generation algorithm and is saved as part of the secret key) and the ciphertext c (sometimes a pseudorandom function is used instead of a hash function).

In the NIST call algorithms, implicit rejection is more used than explicit rejection because the first one provides stronger security. Nevertheless, in these schemes, the secret key is longer and the original PKE key generation algorithm must be modified. In Table 3, the difference in the decapsulation between implicit ($\mathrm{U}^{\not\perp}$) and explicit ($\mathrm{U}^{\perp}$) rejection schemes can be appreciated.

Table 3. Decapsulation algorithm for $\mathrm{U}^{\not\perp}$ and $\mathrm{U}^{\perp}$ transformations.

$\mathbf{U}^{\not\perp}: \mathcal{D}c(sk, c)$	$\mathbf{U}^{\perp}: \mathcal{D}c(sk, c)$
$m' \leftarrow \mathcal{D}(c, sk)$	$m' \leftarrow \mathcal{D}(c, sk)$
if $m' = \perp$	if $m' = \perp$
return $K := H(s, c)$	return $\perp$
else return $K := H(m', c)$	else return $K := H(m', c)$

4.4. Shared Secret and Additional Hash

A classification of the U transformation, based on the definition of the shared secret, was introduced in [5]. As we have mentioned before, the shared secret is defined by using a randomly chosen plaintext, m, and a hash function H. The notation used for the transformations with shared secret, $K = H(m)$, is a subindex $_m$. In these cases, only a DPKE algorithm can be considered. A different definition of shared secret, given in the same paper, used the encryption of the randomly chosen plaintext, $\mathcal{E}(pk, m) = c$. In this case, the shared secret was defined as $K = H(m, c)$, no subindex was used and there were no requirements about the PKE.

In order to obtain quantum security, the use of an additional hash has been proposed in [9] (its security proof was given in [5]). The use of an additional hash was denoted by adding the letter Q in the notation of the transformation, and it is as follows: during the encapsulation process, a value is defined, which is part of the output, say $d = G(m)$. The decapsulation considers as input the pair (c, d), where c is the ciphertext. Then, when the decapsulation checks if there was an error during the decryption (i.e., if $\mathcal{E}(pk, \mathcal{D}(sk, c)) = c$), it also checks if $d = G(m')$.

4.5. Other Modifications

The design variations mentioned so far are the more common modifications in the applications of the FO transformation. However, there are other modifications introduced in the basic scheme (the transformation denoted as U). As it is further explained in Section 5, the Kyber, SABER and FrodoKEM algorithms use the $FO^{\not\perp}$ variant of the transformation, although they introduce a change in the definition of the shared secret. This change is the following: the shared secret is defined as $K = H_3(\hat{K}, c)$, where c is the ciphertext and $\hat{K}$ is generated by a different hash function that takes as inputs $H_2(pk)$ and m, hence

$$(\hat{K}, r) \leftarrow H_2(H_1(pk), m). \tag{1}$$

Notice that the notation for the hash functions is just based in the order in which they are applied. In [22], $\hat{K}$ is denominated as a "pre-key" so, consequently, we say that a transformation uses "pre-key" if it is designed in this way.

Algorithm 1 shows how the encapsulation works when the modification explained above is applied. The use of a "pre-key" also affects the decapsulation in the same sense. Before decrypting the ciphertext, Equation (1) is computed, and the shared secret is defined as before $K = H_3(\hat{K}, c)$ (the final output of the decapsulation depends on the design of the particular transformation). Algorithms such as FrodoKEM or Kyber also introduce changes in the key generation process, simply including the $H_1(pk)$ hash in the public key of the KEM. If nothing is specified, the public key of the KEM and the underlying PKE will be the same.

Algorithm 1 $FO^{\not\perp'}$ Encapsulation

$m \leftarrow_R M$

$(\hat{K}, r) \leftarrow H_2(H_1(pk), m)$

$c = \mathcal{E}(pk, m; r)$

$K = H_3(\hat{K}, c)$

return (K, c)

In the third-round submission of FrodoKEM [23], in regard to security, it is stated that the theorems are still viable while using "pre-key" with just minor adjustments. Grubbs et al. proved in [22] that the transformation $FO^{\not\perp'}$ still provides the same security reduction as $FO^{\not\perp}$ in the QROM.

Another modification of the basic scheme that Kyber [24] and SABER [25] introduced in the transformation is what is denoted as the "nested" hashing of ciphertext in the key generation in [22]. Again, this affects the definition of the shared secret. In this case, instead of the ciphertext, a hash of the same ciphertext, $K = H(\hat{K}, F(c))$ is used, in which F is a hash function (since Kyber and SABER are the algorithms where the nested hashing is applied, and in both algorithms, a pre-key is also used; we have followed their notation and we write $\hat{K}$). In their respective submissions, both algorithms established that this change does not invalidate the previous results that ensure the security of the transformation. However, in [22], it is argued that the same strategy that was used to prove the security of the $FO^{\not\perp}$ cannot be applied here, and hence, the tightness of the security reduction cannot be assured with this change introduced in the transformation.

If we consider the possible combinations of the presented structural aspects, we obtain the transformations introduced by Hofheinz et al. in [5]. The composition of T and U^* transformations (* can be any of the notations introduced in Section 4) is defined as the FO transformation in [5]. This composition is represented in Table 4 for the particular case of the $FO^{\not\perp}$ version. All the variants of the U transformation are represented in Table 5, in which the security reduction of each transformation is also included. It also includes what can

be found the security proofs in the QROM and the requirements of these transformations, i.e., when the underlying PKE has to be deterministic or perfectly correct.

As one can see, combining Tables 2 and 5, the transformation $\text{FO}^{\not\perp}$ presented in [5] has tight security reduction from IND-CPA to IND-CCA in the ROM. This is particularly relevant, since a noticeable number of the finalist and alternative algorithms in the NIST call use this transformation (FrodoKEM, SABER, BIKE, etc.). Hofheinz et al. also introduced the security reductions in the QROM, and the scope turned toward these security reductions in latter publications.

Table 4. Algorithms defining $\text{KEM}^{\not\perp} = \text{U}^{\not\perp}[\pi^\tau, H]$.

$\mathcal{G}$	$\mathcal{E}c(pk)$	$\mathcal{D}c(sk, c)$
$(pk', sk') \leftarrow \mathcal{G}'$	$m \leftarrow_R M$	$m' \leftarrow \mathcal{D}(sk, c)$
$s \leftarrow_R M$	$c \leftarrow \mathcal{E}^\tau(pk, m, H_1(m))$	If $m' = \perp$ and $\mathcal{E}(pk, m', H_1(m')) \neq c$
$sk := (sk', s)$	$K \leftarrow H_2(m, c)$	return $K := H_2(s, c)$
return (pk', sk)	return (K, c)	else return $K := H_2(m', c)$

Table 5. Transformations and the security proof sources.

Transformation	π Secur. ROM	π Secur. QROM	Tight. ROM	Sec. proof QROM	DPKE	Perf. Cor.
$\text{KEM}^{\not\perp} = \text{U}^{\not\perp}[\pi, H]$	OW-PCA	OW-qPCA	[5]	[7]	N	N
$\text{KEM}^{\perp} = \text{U}^{\perp}[\pi, H]$	OW-PCVA	OW-qPVCA	[5]	[7]	N	N
$\text{KEM}_m^{\not\perp} = \text{U}_m^{\not\perp}[\pi, H]$	OW-CPA	OW-CPA, DS	[5]	[6,7]	Y	N [7] Y [6]
$\text{KEM}_m^{\perp} = \text{U}_m^{\perp}[\pi, H]$	OW-VA	OW-VA	[5]	[7]	Y	N [7]
$\text{QKEM}_m^{\perp} = \text{QU}_m^{\perp}[\pi, H, H']$	OW-PCA, OW-CPA	OW-CPA	[5,9]	[5]	N [5] Y [9]	N [5] Y [9]
$\text{QKEM}_m^{\not\perp} = \text{QU}_m^{\not\perp}[\pi, H, H']$	OW-PCA	OW-CPA	[5]	[5]	N	N

Only some of these transformations are applied to the lattice-based schemes of the third-round NIST call; these are $\text{FO}^{\not\perp\prime}$, $\text{U}_m^{\not\perp}$ and $\text{QU}_m^{\perp}$. All of them have tight security proofs in the classical ROM [5]; however, the security proofs in the QROM are not tight. Only the $\text{U}_m^{\not\perp}$ transformation has an almost tight security proof in the QROM as was given in [6].

5. FO Transformation Application in Lattice-Based Algorithms

All the finalist and alternatives of the NIST PQC call use the FO transformation. In fact, most of the lattice-based algorithms use the $\text{FO}^{\not\perp}$ version presented before (see Section 4). In the following, we analyze these proposals but not the rest of algorithms, i.e., NTRU, SABER, FrodoKEM, CRYSTALS-Kyber and NTRUPrime.

5.1. NTRU

NTRU is a lattice-based KEM whose decisional version is reduced to the search RLWE problem and consists in an IND-CCA secure KEM based on the NTRU DPKE. The version of NTRU submitted to the third round is the culmination in the evolution of a family of algorithms based on a very similar underlying PKE.

There are several variants of the NTRU cryptosystem, and the applied transformation differs considerably on each case. Saito et al. [6] presented a version of NTRU that applies the $\text{U}_m^{\not\perp}$ transformation, and the version of NTRU submitted to the third round of the NIST call is quite similar to this one. The NIST version reduces the computational cost of the decapsulation in the following way: instead of using re-encryption to check if the decryption output is a correct plaintext as in [6], the decryption checks if the pair made of the message and the randomness fulfils that $(m, r) \in M \times C$. If verified, then the decryption outputs an additional value 0; else, it outputs an additional value 1. This way, during the NTRU KEM decapsulation, there is no need to encrypt the obtained plaintext, which

signifies an improvement in performance. The IND-CCA security of the KEM reduces to the OW-CPA security of the DPKE [6].

Algorithms 2–4 show how the $U_m^{\not\perp}$ transformation is applied in order to define the key generation, encapsulation and decapsulation algorithms for NTRU, respectively.

Algorithm 2 NTRU KEM KeyGen

$(pk', sk') \leftarrow \mathcal{G}'(1^\lambda)$

$s \leftarrow_R \{0,1\}^{256}$

$sk = (sk', pk', s)$

return (pk', sk)

Algorithm 3 NTRU KEM Encapsulation(pk)

$coins \leftarrow_R \{0,1\}^{256}$

$(r, m) \leftarrow \text{Sample}_{rm}(coins)$

$c \leftarrow \mathcal{E}(pk, (r, m))$

$K \leftarrow H_1(m)$

return (c, K)

Algorithm 4 NTRU KEM Decapsulation(sk, c)

Parse$(sk) = (sk' \| s)$

$(r, m, fail) \leftarrow \mathcal{D}(sk', c)$

$K \leftarrow H_1(m)$

$R \leftarrow H_2(s, c)$

if $fail = 0$

return K

else

return R

5.2. SABER

SABER [25] is a lattice-based KEM based on a PKE scheme whose security relies on the MLWR problem, in which the $FO^{\not\perp'}$ transformation is applied to construct an IND-CCA secure KEM. As was mentioned in Section 4.5, SABER design includes the use of pre-key and nested hashing of the ciphertext to define the shared secret. In the submission of this algorithm to NIST [25], the security proof from Hofheinz et al. [5] was considered. This proof provides (tightly) IND-CCA security to SABER KEM in the ROM based on the IND-CPA security of the underlying PKE. In the QROM, there is no tight security reduction, but there is still a security proof by Jiang et al. [7].

Grubbs et al. [22] state that the IND-CCA security in the QROM for SABER cannot be sustained as the security of the $FO^{\not\perp}$ transformation, since the use of pre-key and nested hashing change the transformation in a significant way, and the strategies used in the security proof are no longer applicable. However, the security proof of a variant of SABER, called proto-SABER [26], which only uses pre-key in the definition of the shared secret, can be adapted maintaining the same tightness as in [7].

Algorithms 5–7 show how the key generation, the encapsulation, and the decapsulation work in SABER.

Algorithm 5 SABER KeyGen

$(seed_A, b, s) \leftarrow \mathcal{G}'$

$pk = (seed_A, b)$

$pkh = H_1(pk)$

$z \leftarrow_R \{0,1\}^{256}$

$sk = (z, pkh, pk, s)$

return (pk, sk)

Algorithm 6 SABER Encapsulation(pk)

$m \leftarrow_R \{0,1\}^{256}$

$(r, \hat{K}) = H_2(H_1(pk)\|m)$

$c \leftarrow \mathcal{E}(pk, m; r)$

$K = H_3(H_3(c)\|\hat{K})$

return (c, K)

Algorithm 7 SABER Decapsulation(sk, c)

$m' \leftarrow \mathcal{D}(s, c)$

$(r', \hat{K}') = H_2(pkh\|m')$

$c' \leftarrow \mathcal{E}(pk, m'; r')$

if $c = c'$ then

return $K = H_3(H_3(c)\|\hat{K}')$

else **return** $K = H_3(H_3(c)\|z)$

5.3. FrodoKEM

FrodoKEM is a KEM whose security is based on the LWE problem. This algorithm is distinguished from other lattice-based algorithms because it does not use a ring or module structure, which makes the algorithm gain security, but it loses in key length and functionality. Currently, in the NIST proposal, FrodoKEM is considered as an alternative. Regardless of this, NIST maintains it as the most promising alternative. The Bundesamt für Sicherheit in der Informationstechnik (BSI) maintains its recommendation of FrodoKEM as a PQC mechanism with a high security margin against future attacks. BSI considers that FrodoKEM has not been included among the third-round finalists of the NIST PQC call due to considerations of the efficiency of the mechanism, but there are currently no doubts about its security [27].

In the case of FrodoKEM, the $FO^{\not\perp'}$ transformation is used, which is slightly different from the $FO^{\not\perp}$ transformation (see Section 4.5). The first difference, which is specific to the Frodo implementation, is that $FO^{\not\perp'}$ uses the same hash function to generate r and K. Another difference between the transformation applied in FrodoKEM and $FO^{\not\perp}$ is the use of a pre-key.

FrodoKEM is designed considering a hash function that takes as inputs a randomly chosen plaintext and the hash of the public key and as output a large bit string. This bit string is then parsed into r and k (K is generated from k and the ciphertext, which is the pre-key modification). It is claimed that the use of a pre-key has the potential to provide stronger multi-target security [23].

In relation to its security, in Th.5.1 of [28], it is proved that the IND-CCA security of FrodoKEM reduces to the IND-CPA security of the underlying PKE. Similarly to other

applications of this transformation, FrodoKEM proof of security in the QROM lies in the results of Jian et al. [7,29]. The proof given in Th.5.8 of [23] provides a non-tight bound for the IND-CCA security of any KEM in the QROM, which is constructed by applying the transformation $FO^{\not\perp}$ to a OW-CPA secure PKE in the QROM. In [22], it is pointed out that the security proof of Jian et al. is not compatible with the modifications made to the transformation in the FrodoKEM submission; however, Grubbs et al. gave an alternative security proof that maintains the same tightness.

The three algorithms defining FrodoKEM are shown as Algorithms 8–10. One can appreciate that there is a similarity with SABER in the application of the T transformation.

Algorithm 8 FrodoKEM KeyGen

$(pk, sk) \leftarrow \mathcal{G}'$

$s \leftarrow_R \{0,1\}^{len_s}$

$pkh = H_1(pk)$

$sk := (sk', s, pk, pkh)$

return (pk, sk')

Algorithm 9 FrodoKEM Encapsulation(pk)

$m \leftarrow_R \mathrm{M}$

$(r, k) = H_2(H_1(pk)\|m)$

$c \leftarrow \mathcal{E}(pk, m; r)$

$K = H_3(c\|k)$

return (c, K)

Algorithm 10 FrodoKEM Decapsulation(sk, c)

$m \leftarrow \mathcal{D}(sk', c)$

$(r', k') = H_2(pkh\|m)$

$K'_0 = H_3(c\|k'), K'_1 = H_3(c\|s)$

if $c = \mathcal{E}(pk, m'; r')$ then $K' = K'_0$

else $K' = K'_1$

return K'

5.4. CRYSTALS-Kyber

CRYSTALS-Kyber is another lattice-based algorithm. The security of the underlying PKE is based on the difficulty of solving the MLWE problem. The algorithm achieves IND-CCA security through the $FO^{\not\perp'}$ transformation to obtain an IND-CCA secure KEM, whose security reduces to IND-CPA security of the PKE.

Similar to SABER, the transformation applied in Kyber introduces both pre-key and nested hashing (see Section 4.5). In the submission of Kyber, there are not explicit results in regard to the security reduction of the algorithm. Instead, the results of Hofheinz et al. [5] are considered for the IND-CCA security of KyberKEM in the ROM. In particular, the security reduction involves the $FO^{\not\perp}$ transformation.

Kyber submission does not present any particular result with the security reduction of the algorithm, although to support the IND-CCA security of KyberKEM in the ROM, the proofs given in [5] that are applied to the $FO^{\not\perp}$ transformation are considered.

In the QROM, the results from [5,6] are considered. It is claimed that these sources provide a non-tight security reduction for the IND-CCA security of KyberKEM into the IND-

CPA security of KyberPKE. However, none of these articles contain a suitable security proof for the transformation applied in Kyber. In the Kyber submission [24], the construction of a deterministic PKE is also considered; it is called DKyber.CPAPKE, which is supposed to be pseudo-random in the QROM [6]. This PKE might be suitable for the $U_m^{\not\perp}$ transformation, which has a tight security proof in the QROM [6].

Even if it is not considered in the same way as FrodoKEM or SABER, the results of Jian et al. [7,29] can be applied to Kyber, giving it a non-tight security reduction in the QROM. Although, similarly to SABER, the proof is not valid, since the algorithm applies a nested hash, as is indicated in [22].

The hash functions G, H, and the key derivation function, KDF, are instantiated as follows: H is instantiated with SHA3-256, G with SHA3-512 and KDF with SHAKE-256. Note that these are not the only hash functions used in this algorithm, since the underlying PKE makes calls to other hash functions. More concretely, Algorithms 11–13 show the three algorithms defining KyberKEM: key generation, encapsulation, and decapsulation, respectively.

Algorithm 11 Kyber KeyGen

$z \leftarrow_R \{0,1\}^{256}$

$(pk,\ sk') \leftarrow \mathcal{G}'$

$sk := (sk' \| pk \| H(pk) \| z)$

return $(pk,\ sk)$

Algorithm 12 Kyber Encapsulation(pk)

$m' \leftarrow_R \{0,1\}^{256}$

$m \leftarrow H(m')$

$(\bar{K}, r) \leftarrow\ H'(m \| H(pk))$

$c \leftarrow \mathcal{E}(pk, m, r)$

$K = KDF(\bar{K} \| H(c))$

return $(c,\ K)$

Algorithm 13 Kyber Decapsulation(sk, c)

$h = sk + 24 \cdot k \cdot n/8 + 32$

$z = sk + 24 \cdot k \cdot n/8 + 64$

$m' \leftarrow \mathcal{D}(sk, c)$

$(\bar{K}', r') \leftarrow H'(m' \| h)$

$c' \leftarrow \mathcal{E}(pk, m', r')$

if $c = c'$ **return** $K = KDF(\bar{K}' \| H(c))$

else **return** $K = KDF(z \| H(c))$

5.5. NTRUPrime

NTRUPrime NIST submission [30] focuses much more on the PKE part of the algorithm than in the KEM structure and, consequently, in the applied transformation. The security claim quoted in the first round submission is due to Dent (Section 6 in [31]). Actually, he applied transformation in the same manner as in [5] and defined it as $QU_m^{\perp}$. As it was explained in previous sections, Dent's work has the drawback that the underlying PKE has to be perfectly correct. The two versions of NTRUPrime (NTRU LPRime and

Streamlined NTRU Prime) have sets of parameters that avoid decryption failures. It is acknowledged in the NTRUPrime submission that taking these sets of parameters suppose a worse functionality; however, it is prioritized, ensuring security over functionality.

Nevertheless, there are some changes introduced to the applied transformation in the second round. The session key is defined as a hash with two inputs: a random plaintext and the corresponding ciphertext, while the additional hash takes as input the public key:

$$K = H_1(m) \rightsquigarrow K = H_1(m, c); \quad c_1 = H_2(m) \rightsquigarrow c_1 = H_2(m, pk).$$

The second change that was introduced is that implicit rejection is used as an extra security measure that makes it more difficult to recognize invalid ciphertexts. These changes imply that the new transformation is $QU^{\not\perp}$. The IND-CCA ROM security claim is based in [32], and the security in the QROM is quoted by Saito et al. [6]. However, it is not proven that the underlying PKE of this algorithm achieves the initial security claims in [6], i.e., DS.

In the third round, no changes were proposed to the transformation, and a tighter security proof for the Streamlined NTRUPrime KEM based on the results in [33] is presented.

6. Transformation without Re-Encryption

As we have previously commented (see Section 3), side-channel attacks that exploit the re-encryption process in the decapsulation algorithm can be performed against the studied KEMs. So, in order to resist these attacks, or at least to reduce the impact of CC-SCA, it is desirable to have a KEM that does not use re-encryption. In what follows, we propose a light modification of the Kyber algorithm.

The transformation presented by FrodoKEM, Kyber, SABER and NTRUPrime applies re-encryption (see Section 4.5); i.e., in the decapsulation, after decrypting the message, the obtained plaintext is encrypted and the algorithm checks if the result is equal to the initial ciphertext. We focus our attention on Kyber, as it has been chosen to be the post-quantum standard. In [13], the use of re-encryption is studied from the perspective of formal security reductions.

The motivation of this proposal comes from the fact that several publications (v.gr. [10,13,16,17]) present attacks against the re-encryption part of the $FO^{\not\perp'}$ transformation applied to Kyber. As explained in [17], the re-encryption process consists in running the PKE, which in the case of Kyber is only IND-CPA, so an IND-CCA adversary may have a significant advantage against the scheme.

Our proposal to avoid the re-encryption is to use the algorithm with an ephemeral key set-up, which means that each key pair (pk, sk) is considered only once or for just one key exchange. The NIST call [34] indicates that IND-CCA security is the desirable semantic security notion that the established standard must reach. However, if the scheme is used with ephemeral keys, then IND-CPA security is considered to be sufficient. There is a precedent in this kind of practice (i.e., the implementation with ephemeral keys), for example BIKE [35], which is a code-based algorithm that was considered as an alternative after the second round of the NIST call, and currently, it is a candidate to be the code-based standard in the fourth round of the call. Of course, this structure is only useful if the security against side-channel attacks is considered a priority, since becoming lost in functionality is plausible. Nevertheless, this new structure presents a serie of drawbacks: (1) the key generation algorithm must be executed for every communication and (2) the possibility of receiving messages without any prior communication is lost, which is one of the main characteristics of public key encryption (although the KEM structure has already put this aside from a certain perspective).

The first mentioned drawback implies, in general, a higher computational cost; however, in our proposal, this increase is negligible in the initial exchanges, since the computational cost of just the PKE is lower (and close to our proposal) than the cost of executing the whole KEM. In fact, the re-encryption is one of the highest cost operations that the FO transformation adds to the scheme, so it should be avoided.

The structure of the resulting scheme, which we have denoted as KyberEph, based on KyberPKE $= \{\mathcal{G}', \mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{C}\}$ is shown as Algorithms 14–16, where an additional hash function H' is considered.

Algorithm 14 KyberEph KeyGen

$z \leftarrow_R \{0,1\}^{256}$

$(pk,\ sk') \leftarrow \mathcal{G}'$

$sk := (sk' \| pk \| H(pk) \| z)$

return $(pk,\ sk)$

Algorithm 15 KyberEph Encapsulation(pk)

$m' \leftarrow_R \{0,1\}^{256}$

$m \leftarrow H(m')$

$(\bar{K}, r) \leftarrow\ H''(m \| H(pk))$

$c_1 \leftarrow \mathcal{E}(pk, m, r)$

$H'(m) = c_2$

$K = KDF(\bar{K} \| H(c))$

return $(c := (c_1, c_2),\ K)$

Algorithm 16 KyberEph Decapsulation(sk, c)

Parse$(c) = (c_1, c_2)$

$h = sk + 24 \cdot k \cdot n/8 + 32$

$z = sk + 24 \cdot k \cdot n/8 + 64$

$m' \leftarrow \mathcal{D}(sk, c_1)$

if $H'(m') = c_2$ **return** $K = KDF(\bar{K}' \| H(c))$

else **return** $K = KDF(z \| H(c))$

In this proposal of the Kyber scheme, the key generation and the encapsulation algorithms are basically the same. Not using re-encryption may suppose a problem against other attacks, such as receiving false ciphertext. Hence, to be able to maintain the security, the re-encryption is replaced with the use of an additional hash. This is quite similar to the QFO transformation, where a hash of the message is added to the ciphertext, and it helps to check if the decrypted plaintext is correct. We also considered the use of a conditional clause that checks if the message and the randomness obtained during the decryption algorithm execution are well defined based on how NTRU avoids re-encryption. However, since Kyber is not a perfectly correct cryptosystem, this initial design was discarded.

Since re-encryption is not used, this means that the resulting KEM cannot rely on the security proof (from [5]) for the T transformation. As stated before, this loss in security is patched up using the scheme with only ephemeral keys. Then, the Kyber PKE reaches IND-CPA security (Th.1 of [24]) in the ROM, which reduces the hardness of the M-LWE problem. In the QROM, the Kyber submission also considers that the scheme is IND-CPA secure.

7. Conclusions and Future Work

In this work, we have analyzed and compared some variants of the transformation applied to the NIST PQC lattice-based candidates. The importance of this transformation is seen clearly, since it does not only apply to all lattice-based candidates in the third round of

the NIST PQC call but also in almost every algorithm submitted to the call, and it is the tool used by the algorithms to reach IND-CCA security.

In the particular case of the algorithms presented in this paper, there are several things to point out. First of all, it is clear that the nature and characteristics of the PKE designed on each submission have a considerable repercussion in the applied transformation to conform a KEM. If the PKE is deterministic and perfectly correct, then the results of Saito et al. [6] can be considered; if not, then the transformation $FO^{\not\perp\prime}$ seems to be the best option, since the three algorithms in this situation have chosen it.

The fact that several algorithms use the same transformation leaves the possibility of creating compatible implementations open, since the algorithms for the key generation, encapsulation, and decapsulation of the KEM are all the same except for the choice of certain hash functions. This may be applicable to FrodoKEM, Kyber and SABER.

The main open problem in this field is the search of a tight proof in the QROM for the transformation that is used in the algorithms. Although this is an important issue in several applications, we pointed out that having a tight security proof is not a requirement for the transformation to provide security. Just in the case the security proof is not tight, the parameters involved have to be adjusted to make the probability of success of any attacker negligible. This process of adjustment to the current bounds of security is not explained in detail in the algorithm submissions to the third round of the NIST call. However, to be aware of the range in which each parameter can variate without supposing a loss in the algorithm security may be of interest. There are still several open problems that should be studied in this field, which will be proposed in later works.

Other issues that need to be addressed are the statements in [22], which are related to the relevance of the changes introduced in the $FO^{\not\perp\prime}$ in Kyber, SABER and FrodoKEM. Specifically, Kyber and SABER introduce the use of pre-key and nested hashing, and for this case, neither Grubbs et al. [22] nor Xagawa [36] included a proof of security. Moreover, Grubbs and Xagawa both study how the Fujisaki–Okamoto transformation affects the anonymity of the KEMs submitted to the NIST call. Since we focus our scope in the security of the algorithms, this was out of the objective of this work and may be addressed in another work in the future. Still, it may be interesting to continue studying these results, since in their analysis of the anonymity, the transformations with explicit rejection have better results. This contradicts the clear tendency of the lattice-based algorithms, where all of them apply explicit rejection, since it provides better security reductions.

The scarce use of the transformations with additional hash (QFO^*) is clearly due to the work of Jian et al. [7] and Saito et al. [6], which provided tighter security proofs in the QROM for the U^* transformations. Even in the NTRUPrime submission, that uses additional hashes, in the modifications introduced for the second and third rounds, the possibility of removing the additional hash is foreseen. However, this possibility has not changed under the claim that there is no tight security proof or that the one available considers new security definitions for the underlying PKE that have not been explored and studied enough.

As for our proposal of using Kyber with ephemeral keys, it must be considered that this model is in its initial state. Our main focus is to avoid possible leakages of data due to side-channel attacks; however, other issues may be raised in this kind of implementation. One that will be addressed in future works is how the fact that Kyber is not perfectly correct could affect this implementation.

Finally, Table 6 summarizes the main characteristics of the lattice-based NIST algorithms: transformations that are applied, underlying security in the ROM, perfect correctness of the underlying PKE, deterministic PKE, source quoted for the security proof in ROM and in QROM.

Table 6. NIST lattice-based proposals and their main characteristics.

Name	Transf.	ROM (Sec.)	Perf. Cor.	DPKE	ROM	QROM
NTRU	$U_m^{\not\perp}$	OW-CPA	Y	Y	[5,6]	[6]
SABER	$FO^{\not\perp(')}$	IND-CPA	N	N	[5]	[7]
FrodoKEM	$FO^{\not\perp'}$	IND-CPA	N	N	[5]	[7]
NTRUPrime	$QU^{\not\perp}$	OW-CPA	Y	Y	[31,32]	[6]
Kyber	$FO^{\not\perp(')}$	IND-CPA	N	N	[5]	[7]

Author Contributions: Conceptualization, M.Á.G.d.l.T., L.H.E. and A.Q.-D.; Formal analysis, M.Á.G.d.l.T.; Investigation, M.Á.G.d.l.T. and L.H.E.; Methodology, L.H.E. and A.Q.-D.; Project administration, L.H.E.; Supervision, A.Q.-D.; Validation, M.Á.G.d.l.T. and L.H.E.; Writing—original draft, M.Á.G.d.l.T., L.H.E. and A.Q.-D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by ORACLE Project, with reference PCI2020-120691-2, funded by MCIN/AEI/10.13039/501100011033, and European Union "NextGenerationEU/PRTR" and in part by the Spanish State Research Agency (AEI) of the Ministry of Science and Innovation (MCIN), project P^{2}QProMeTe (PID2020-112586RB-I00/AEI/10.13039/501100011033), and in part by the EU Horizon 2020 research and innovation programme, project SPIRS (Grant Agreement No. 952622).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We kindly acknowledge the help provided by the reviewers' suggestions, which have contributed to an improved quality of the present work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CC-SCA	Chosen-Ciphertext Side-Channel Attacks
CVO	Ciphertext validation oracle
CVP	Closest Vector Problem
DPA	Differential Power Analysis
DPKE	Deterministic Public Key Encryption
DS	Disjoint Simulatability
$\mathcal{E}t\mathcal{S}$	Encrypt-then-Sign
FO	Fujisaki–Okamoto
IND-CCA	INDistinguishability under Adaptive Chosen Ciphertext Attack
IND-CPA	INDistinguishability under Chosen Plaintext Attack
KEM	Key Exchange Mechanism
LWE	Learning With Errors
MLWE	Module Learning With Errors
MLWR	Module Learning With Rounding
NIST	National Institute of Standards and Technology
OW	One-Wayness
OW-CPA	One-Way Chosen Plaintext Attacks
OW-PCA	One-Way Plaintext Checking Attack
OW-PCVA	One-Way-Plaintext-Checking-Validation-Attack
OW-VA	One-Way Validation Attack
PCO	Plaintext checking oracle
PKE	Public Key Encryption

PPKE	Probabilistic Public Key Encryption
PQC	Post-Quantum Cryptography
QROM	Quantum Random Oracle Model
RLWE	Ring Learning With Errors
ROM	Random Oracle Model
SKE	Symmetric Key Encryption
SPA	Simple Power Analysis
SVP	Shortest Vector Problem

References

1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. https://doi.org/10.1137/S0036144598347011.
2. González de la Torre, M.; Hernández Encinas, L. About the Fujisaki-Okamoto Transformation in the Code-based Algorithms of the NIST Post-Quantum Call. In Proceedings of the International Conference on Computational Intelligence in Security for Information Systems (CISIS 2022), Salamanca, Spain, 5–7 September 2022.
3. Fujisaki, E.; Okamoto, T. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In Proceedings of the 19th Annual International Cryptology Conference, Advances in Cryptology—CRYPTO'99, Santa Barbara, CA, USA, 15–19 August 1999; Volume 1666, pp. 537–554. https://doi.org/10.1007/3-540-48405-1_34.
4. Cramer, R.; Shoup, V. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. Cryptology ePrint Archive, Report 2001-108. 2001. Available online: https://eprint.iacr.org/2001/108 (accessed on 16 August 2022).
5. Hofheinz, D.; Hövelmanns, K.; Kiltz, E. A Modular Analysis of the Fujisaki-Okamoto Transformation. In Proceedings of the 15th International Conference Theory of Cryptography TCC'2017, Baltimore, MD, USA, 12–15 November 2017; Volume 10677, pp. 341–371. https://doi.org/10.1007/978-3-319-70500-2_12.
6. Saito, T.; Xagawa, K.; Yamakawa, T. Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology— EUROCRYPT 2018, Tel Aviv, Israel, 29 April–3 May 2018; Volume 10822, pp. 520–551. https://doi.org/10.1007/978-3-319-78372-7_17.
7. Jiang, H.; Zhang, Z.; Chen, L.; Wang, H.; Ma, Z. IND-CCA-secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited. Cryptology ePrint Archive, Report 2017-1096. 2017. Available online: https://eprint.iacr.org/2017/1096 (accessed on 16 August 2022).
8. Jiang, H.; Zhang, Z.; Ma, Z. On the Non-Tightness of Measurement-Based Reductions for Key Encapsulation Mechanism in the Quantum Random Oracle Model. Cryptology ePrint Archive, Paper 2019/494. 2019. Available online: https://eprint.iacr.org/2019/494 (accessed on 16 August 2022).
9. Dent, A.W. A Designer's Guide to KEMs. In Proceedings of the 9th IMA International Conference on Cryptography and Coding, Cirencester, UK, 16–18 December 2003; Volume 2898. https://doi.org/10.1007/978-3-540-40974-8_12.
10. Ravi, P.; Roy, S.S.; Chattopadhyay, A.; Bhasin, S. Generic Side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2020**, *3*, 307–335. https://doi.org/10.13154/tches.v2020.i3.307-335.
11. Ngo, K.; Dubrova, E.; Guo, Q.; Johansson, T. A Side-Channel Attack on a Masked IND-CCA Secure Saber KEM Implementation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *4*, 676–707. https://doi.org/10.46586/tches.v2021.i4.676-707.
12. Hövelmanns, K.; Hülsing, A.; Majenz, C. Failing Gracefully: Decryption Failures and the Fujisaki-Okamoto Transform. Cryptology ePrint Archive, Report 2022/365. 2022. Available online: https://eprint.iacr.org/2022/365 (accessed on 16 August 2022).
13. Ueno, R.; Xagawa, K.; Tanaka, Y.; Ito, A.; Takahashi, J.; Homma, N. Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *1*, 296–322. https://doi.org/10.46586/tches.v2022.i1.296-322.
14. Xagawa, K.; Ito, A.; Ueno, R.; Takahashi, J.; Homma, N. Fault-Injection Attacks against NIST's Post-Quantum Cryptography Round 3 KEM Candidates. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2021), Singapore, 6–10 December 2021; Volume 13091, pp. 33–61. https://doi.org/10.1007/978-3-030-92075-3_2.
15. Xu, Z.; Pemberton, O.; Roy, S.S.; Oswald, D.; Yao, W.; Zheng, Z. Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems with Chosen Ciphertexts: The Case Study of Kyber. *IEEE Trans. Comput.* **2020**, *71*, 2163–2176. https://doi.org/10.1109/TC.2021.3122997.
16. Hermelink, J.; Pessl, P.; Pöppelmann, T. Fault-enabled chosen-ciphertext attacks on Kyber. In Proceedings of the International Conference on Cryptology in India (INDOCRYPT 2021), Jaipur, India, 13–15 December 2021; Volume 13143, pp. 311–334. https://doi.org/10.1007/978-3-030-92518-5_15.
17. Azouaoui, M.; Bronchain, O.; Hoffmann, C.; Kuzovkova, Y.; Schneider, T.; Standaert, F.X. Systematic Study of Decryption and Re-Encryption Leakage: The Case of Kyber. In Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE 2022), Leuven, Belgium, 12–14 April 2022; Volume 13211, pp. 236–256. https://doi.org/10.1007/978-3-030-99766-3_11.

18. Azouaoui, M.; Kuzovkova, Y.; Schneider, T.; van Vredendaal, C. Post-Quantum Authenticated Encryption against Chosen-Ciphertext Side-Channel Attacks. Cryptology ePrint Archive, Report 2022/91. 2022. Available online: https://eprint.iacr.org/2022/916 (accessed on 16 August 2022).
19. Zheng, Y. Signcryption and its applications in efficient public key solutions. In Proceedings of the International Workshop on Information Security (ISW 97), Tatsunokuchi, Japan, 17–19 September 1997; Volume 1396, pp. 291–312. https://doi.org/10.1007/BFb0030430.
20. An, J.H.; Dodis, Y.; Rabin, T. On the security of joint signature and encryption. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology—EUROCRYPT 2002, Amsterdam, The Netherlands, 28 April–2 May 2002; Volume 2332, pp. 83–107. https://doi.org/10.1007/3-540-46035-7_6.
21. Coron, J.S.; Handschih, H.; Joye, M.; Pailier, P.; Pointcheval, D.; Tymen, C. GEM: A Generic Chosen-Ciphertext Secure Encryption Method. In Proceedings of the Topics in Cryptology—CT-RSA 2002: The Cryptographers' Track at the RSA Conference 2002, San Jose, CA, USA, 18–22 February 2002; Volume 2271, pp. 263–276. https://doi.org/10.1007/3-540-45760-7_18.
22. Grubbs, P.; Maram, V.; Paterson, K.G. Anonymous, Robust Post-Quantum Public Key Encryption. Cryptology ePrint Archive, Report 2021/708. 2021. Available online: https://ia.cr/2021/708 (accessed on 16 August 2022).
23. Alkim, E.; Bos, J.W.; Ducas, L.; Longa, P.; Mironov, I.; Naehrig, M.; Nikolaenko, V.; Peikert, C.; Raghunathan, A.; Stebila, D. FrodoKEM Learning with Errors Key Encapsulation (Round 3 Submission). Online Publication. 2021. Available online: https://frodokem.org/#spec (accessed on 16 August 2022).
24. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber. Online Publication. 2020. Available online: https://pq-crystals.org/ (accessed on 16 August 2022).
25. Basso, A.; Mera, J.M.B.; D'Anvers, J.P.; Karmakar, A.; Sinha, S.; Beirendonck, M.V.; Vercauteren, F. SABER: Mod-LWR Based KEM (Round 3 Submission). Online Publication. 2020. Available online: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/ (accessed on 16 August 2022).
26. D'Anvers, J.P.; Karmakar, A.; Roy, S.S.; Vercauteren, F. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. Cryptology ePrint Archive, Report 2018/230. 2018. Available online: https://ia.cr/2018/230 (accessed on 16 August 2022).
27. BSI. *Cryptographic Mechanisms: Recommendations and Key Lengths, Version 2022-01*; BSI TR-02102-1, 2022/01/28; Bundesamt für Sicherheit in der Informationstechnik: London, UK, 2022. Available online: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf (accessed on 16 August 2022).
28. Bos, J.W.; Costello, C.; Ducas, L.; Mironov, I.; Naehrig, M.; Nikolaenko, V.; Raghunathan, A.; Stebila, D. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS'16, Vienna, Austria, 24–28 October 2016; pp. 1006–1018. https://doi.org/10.1145/2976749.2978425.
29. Jiang, H.; Zhang, Z.; Ma, Z. Tighter Security Proofs for Generic Key Encapsulation Mechanism in the Quantum Random Oracle Model. Cryptology ePrint Archive, Report 2019/134. 2019. Available online: https://eprint.iacr.org/2019/134 (accessed on 16 August 2022).
30. Bernstein, D.J.; Brumley, B.B.; Chen, M.S.; Chuengsatiansup, C.; Lange, T.; Marotzke, A.; Peng, B.Y.; Tuveri, N.; van Vredendaal, C.; Yang, B.Y. NTRU Prime: Round 3. Online Publication. 2020. Available online: https://ntruprime.cr.yp.to/papers.html (accessed on 16 August 2022).
31. Dent, A.W. A Designer's Guide to KEMs. Cryptology ePrint Archive, Report 2002-174. 2002. Available online: https://eprint.iacr.org/2002/174 (accessed on 16 August 2022).
32. Bernstein, D.J.; Persichetti, E. Towards KEM Unification. Cryptology ePrint Archive, Report 2018/526. 2018. Available online: https://eprint.iacr.org/2018/526 (accessed on 16 August 2022).
33. Bindel, N.; Hamburg, M.; Hövelmanns, K.; Hülsing, A.; Persichetti, E. Tighter Proofs of CCA Security in the Quantum Random Oracle Model. Cryptology ePrint Archive, Report 2019/590. 2019. Available online: https://eprint.iacr.org/2019/590 (accessed on 16 August 2022).
34. NIST. PQC Standardization Process: Third Round Candidate Announcement. Online Publication. 2020. Available online: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement (accessed on 16 August 2022).
35. Aragon, N.; Barreto, P.; Bettaieb, S.; Bidoux, L.; Blazy, O.; Deneuville, J.C.; Gaborit, P.; Gueron, S.; Guneysu, T.; Aguilar Melchor, C.; et al. *BIKE (Bit Flipping Key Encapsulation)*. NIST, Round 2. 2020. Available online: https://bikesuite.org/files/v4.0/BIKE_Spec.2020.05.03.1.pdf (accessed on 16 August 2022).
36. Xagawa, K. Anonymity of NIST PQC Round 3 KEMs. Cryptology ePrint Archive, Report 2021/1323. 2021. Available online: https://eprint.iacr.org/2021/1323 (accessed on 16 August 2022).