Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms

Ehsan Ebrahimi Targhi and Dominique Unruh

University of Tartu, Estonia

Abstract. In this paper, we present a hybrid encryption scheme that is chosen ciphertext secure in the quantum random oracle model. Our scheme is a combination of an asymmetric and a symmetric encryption scheme that are secure in a weak sense. It is a slight modification of the Fujisaki-Okamoto transform that is secure against classical adversaries. In addition, we modify the OAEP-cryptosystem and prove its security in the quantum random oracle model based on the existence of a partial-domain one-way injective function secure against quantum adversaries.

Keywords: Quantum, Random oracle, Indistinguishability against chosen ciphertext attacks.

1 Introduction

The interest in verifying the security of cryptosystems in the presence of a quantum adversary increased after the celebrated paper of Shor [10]. Shor showed that any cryptosystem based on the factoring problem and the discrete logarithm problem is breakable in the presence of a quantum adversary. Also, many efficient classical cryptosystems are proved to be secure in the random oracle model [3] and many of them still lack an equivalent proof in the quantum setting. Therefore, even if we find a cryptographic primitive immune to quantum attacks, to construct an efficient cryptosystem secure against quantum adversaries, we may have to consider its security in the quantum random oracle model in which the adversary has quantum access to the random oracle.

Fujisaki and Okamoto [8] constructed a hybrid encryption scheme that is secure against chosen ciphertext attacks (IND-CCA) in the random oracle model. Their scheme is a combination of a symmetric and an asymmetric encryption scheme using two hash functions where the symmetric and asymmetric encryption schemes are secure in a very weak sense. However, their proof of security works against only classical adversaries and it is not clear how one can fix their proof in the quantum setting. In the following, we mention the parts of the classical proof that may not work in the quantum setting.

(a) The classical proof uses the list of all queries made to the random oracles to simulate the decryption algorithm without possessing the secret key of the asymmetric encryption scheme. In the quantum case, where the adversary has quantum access to the random oracles and submits queries in superpositions, such a list is not a well-defined concept.

(b) Also, the classical proof uses the fact that using a random value h^* instead of a given random oracle output $H(x)$ cannot be noticed by the adversary, provided that the adversary never queries x from the random oracle. In the quantum setting, the adversary may in a certain sense always query all values x by querying the random oracle on the superposition $\sum_x |x\rangle$ of all values. The situation gets especially difficult since the value x depends in turn on messages produced by the adversary.

(c) Finally, the classical proof uses the fact that for a randomized encryption scheme, it is hard to find values $x \neq x'$ such that encrypting a message m with randomness $H(x)$ or $H(x')$ leads to the same ciphertext. (Note: this does not follow directly from the collision resistance of the random oracle H.)

Consequently, the quantum security of the scheme is left as an open problem by Boneh et al. [6] and Zhandry [17].

We show how to circumvent those problems. Problem (c) is solved by using a recent result showing the collision resistance of random functions with outputs sampled from a non-uniform distribution [12]. Problem (b) is solved by the "one-way to hiding" lemmas from [13, 14] which gives us a tool for handling the reprogramming of the random oracle. Problem (a) remains. In fact, we do not have a proof for the unmodified Fujisaki-Okamoto scheme. However, we show how to solve the problem by adding one more hash value $H'(\delta)$ to the ciphertext. Although in general, it may not be well-defined in the quantum setting what the list of queries to the random oracle is, we can show it to be well-defined in this case, using the fact that range and domain of H' have the same size. (A similar idea was used by [15] for the construction of quantum-secure non-interactive zero-knowledge proofs.)

Bellare and Rogaway [4] proposed another method, named OAEP, for converting a trapdoor permutation into an encryption scheme. It was believed that the OAEP-cryptosystem is provable secure in the random oracle model based on one-wayness of trapdoor permutation, but Shoup [11] showed it is an unjustified belief. Later, Fujisaki et al. [9] proved IND-CCA security of the OAEP-cryptosystem based on a stronger assumption, namely, partial-domain one-wayness of the underlying permutation. As pointed out by [6], the proof of OAEP security uses preimage awareness (i.e., that the preimage of a random oracle query is well-defined and known to the algorithm making it), a technique that does not seem to work in the quantum setting. This problem is the same as problem (a) above, we show that a similar approach works also in the case of OAEP.

Our Contribution. We modify the hybrid encryption scheme presented by Fujisaki and Okamoto using an extra hash function H'. We prove that our scheme is indistinguishable secure against chosen ciphertext attacks in the quantum random oracle model. For a message m, the encryption algorithm of our scheme, Enc^{hy}_{pk}, works as follows:

$$Enc^{hy}_{pk}(m;\delta) = \left(Enc^{asy}_{pk}\Big(\delta; H\big(\delta \| Enc^{sy}_{G(\delta)}(m)\big)\Big),\ Enc^{sy}_{G(\delta)}(m),\ H'(\delta) \right)$$

where pk and sk are the public key and the secret key of the asymmetric encryption scheme. Enc_{pk}^{asy} and Enc_{sk}^{sy} are the asymmetric and symmetric encryption algorithms respectively and δ is a random element from the message space of the asymmetric encryption scheme. H, G and H' are random oracles. The asymmetric encryption scheme is one-way secure, that is, the adversary can not decrypt the encryption of a random message. The symmetric encryption scheme is one-time secure, that is, the adversary can not distinguish between the encryptions of two messages when a fresh key is used for every encryption. In addition, the asymmetric encryption scheme is well-spread, i.e. any message can lead to at least $2^{\omega(\log n)}$ potential ciphertexts.

Note that our modification increases the ciphertext size by only a single hash value $H'(\delta)$ and is computationally inexpensive.

As already mentioned above, the added hash value $H'(\delta)$ solves problem (a) because given $H'(\delta)$, it is well-defined what δ is. This is because H' is chosen to have the same domain and range size, and hence is indistinguishable from a permutation [16]. However, in the formal proof, we do not directly use that fact, instead our proof goes along the following lines: We replace H' with a random polynomial to force the adversary to submit the input that has been used to obtain the ciphertext. This can be done due to a result by Zhandry [17] that shows a random oracle is indistinguishable from a $2q$-wise independent function where q is the number of queries that the adversary makes to the oracle function. In addition, we use the "one way to hiding" lemmas presented in [13, 14]. As soon H' is implemented as a polynomial, we can use the fact that roots of a polynomial can be found in polynomial-time; this allows us to efficiently get all candidates for δ given $H'(\delta)$.

Also, we modify OAEP-cryptosystem and prove its security in the quantum random oracle model based on the existence of a partial-domain one-way trapdoor injective function secure against quantum adversaries. This will remain theoretical until a candidate for a quantum secure partial-domain one-way trapdoor injective function is discovered. The proof follows similar lines as that of the Fujisaki-Okamoto transform.

A note on superposition queries. Following [6], we use the quantum random oracle model in which the adversary can make queries to the random oracle in superposition (that is, given a superposition of inputs, he can get a superposition of output values). This is necessary since a quantum adversary attacking a scheme based on a real hash function is necessarily able to evaluate that function in superposition. Hence the random oracle model must reflect that ability.

However, we do not model superposition queries to the encryption and decryption oracles. (As was done, for example, in [7].) We do strive to achieve security for the case where the encryption is used within a classical protocol (this is modeled by the fact that plaintexts and ciphertexts are classical, while the adversary is quantum), which is probably the most important use case for post-quantum secure encryption schemes.

In contrast, [7] considers security where an encryption scheme intended for classical plaintexts is used with a quantum superposition of plaintexts. And [1] considers the case where an encryption scheme intended for encrypting quantum data is used.

On the necessity of our modifications. We have slightly modified both the Fujisaki-Okamoto and the OAEP-cryptosystem by adding one additional hash to the ciphertexts. Although these additions are not very costly, it is a natural question whether they are necessary, especially in light of the question whether existing implementations are post-quantum secure. Although it is clear that our proof technique strongly relies on these additional hashes, this does not mean that the original schemes are insecure. However, we urge the reader not to assume that they are post-quantum secure just because they are classically secure. For example, in [2] it was shown that (at least relative to a specific oracle) the Fiat-Shamir transform is insecure in the quantum setting (using quantum random oracles). Their setting is similar to ours, so while there are no known attacks on Fujisaki-Okamoto or OAEP, we should not rely on their security until a security proof is found. We leave finding either an attack or a proof as a (highly non-trivial) open problem.

Organization. In Section 2, we present the required security definitions and other definitions, as well as various theorems related to random oracles that we import from the prior works. In Section 3, we define our variant of the Fujisaki-Okamoto transform and prove its security. In Section 5, we define our variant of OAEP. The security proof of our variant of OAEP is presented in Appendix **??**.

2 Preliminaries

Let $\mathtt{KSP}$ and $\mathtt{MSP}$ stand for the key space and the message space respectively. The notation $x \xleftarrow{\$} X$ means that x is chosen uniformly at random from the set X. A symmetric encryption scheme and an asymmetric encryption scheme are defined as follows:

A symmetric encryption scheme Π consists of two polynomial time (in the security parameter n) algorithms, $\Pi = (Enc, Dec)$, such that:

1. Enc, the encryption algorithm, is a probabilistic algorithm which takes as input a key $k \in \mathtt{KSP}$ and a message $m \in \mathtt{MSP}$ and outputs a ciphertext $c \leftarrow Enc_k(m)$. The message space can be infinite and may depend on the security parameter.
2. Dec, the decryption algorithm, is a deterministic algorithm that takes as input a key k and a ciphertext c and returns message the $m := Dec_k(c)$. It is required that decryption algorithm returns the original message, i.e., $Dec_k(Enc_k(m)) = m$, for every $k \in \mathtt{KSP}$ and every $m \in \mathtt{MSP}$.

An asymmetric encryption scheme Π consists of three polynomial time (in the security parameter n) algorithms, $\Pi = (Gen, Enc, Dec)$, such that:

1. Gen, the key generation algorithm, is a probabilistic algorithm which on input 1^n outputs a pair of keys, $(pk, sk) \leftarrow Gen(1^n)$, called the public key and the secret key for the encryption scheme, respectively.
2. Enc, the encryption algorithm, is a probabilistic algorithm which takes as input a public key pk and a message $m \in \mathtt{MSP}$ and outputs a ciphertext $c \leftarrow Enc_{pk}(m)$. The message space, $\mathtt{MSP}$, may depend on pk.
3. Dec, the decryption algorithm, is a deterministic algorithm that takes as input a secret key sk and a ciphertext c and returns message the $m := Dec_{sk}(c)$. It is required that the decryption algorithm returns the original message, i.e., $Dec_{sk}(Enc_{pk}(m)) = m$, for every $(pk, sk) \leftarrow Gen(1^n)$ and every $m \in \mathtt{MSP}$. The algorithm Dec returns $\perp$ if ciphertext c is not decryptable.

Let $y := Enc_{pk}(x; h)$ be the encryption of message x using the public key pk and the randomness $h \in \mathtt{COIN}$ where $\mathtt{COIN}$ stands for the coin space of the encryption scheme. $\Pr[P : G]$ is the probability that the predicate P holds true where free variables in P are assigned according to the program in G.

Definition 1 (γ-spread, Definition 5.2 [8]). *An asymmetric encryption scheme $\Pi = (Gen, Enc, Dec)$ is γ-spread if for every pk generated by $Gen(1^n)$ and every $x \in \mathtt{MSP}$,*

$$\max_{y \in \{0,1\}^*} \Pr[y = Enc_{pk}(x; h) : h \xleftarrow{\$} \mathtt{COIN}] \leq \frac{1}{2^\gamma}.$$

Particularly, we say that the encryption scheme Π is well-spread if $\gamma = \omega(\log(n))$.

Definition 2. *We say that a function $f : \{0,1\}^{n_1} \rightarrow \{0,1\}^{n_2}$ has min-entropy k if*

$$-\log \max_{y \in \{0,1\}^{n_2}} Pr[y = f(x) : x \xleftarrow{\$} \{0,1\}^{n_1}] = k.$$

2.1 Security Definitions

Let $\mathsf{negl(n)}$ be any non-negative function that is smaller than the inverse of any non-negative polynomial $p(n)$ for sufficiently large n. That is, $\lim_{n\rightarrow\infty} \mathsf{negl(n)}p(n) = 0$ for any polynomial $p(n)$. In the following, we present the security definitions that are needed in this paper. Note that the definitions are the same as the security definitions in [8], except they have been represented in the presence of a **quantum** adversary in this paper. As the following two security definitions will both be used in the security proof of our scheme, we differentiate between them by using $\mathsf{negl(n)}^{sy}$ and $\mathsf{negl(n)}^{asy}$ in the definitions.

Definition 3 (One-time secure). *A symmetric encryption scheme $\Pi = (Enc, Dec)$ is one-time secure if no **quantum** polynomial time adversary $\mathcal{A}$ can win in the $PrivK^{OT}_{\mathcal{A},\Pi}(n)$ game, except with probability at most 1/2 + $\mathsf{negl(n)}^{sy}$:*

$PrivK^{OT}_{\mathcal{A},\Pi}(n)$ game:
Key Gen: *The challenger picks up a key k from $\mathtt{KSP}$ uniformly at random, i.e.,*

$k \xleftarrow{\$} \mathtt{KSP}$.

Query: *The adversary $\mathcal{A}$ on input (1^n) chooses two messages m_0, m_1 of the same length and sends them to the challenger. The challenger chooses $b \xleftarrow{\$} \{0,1\}$ and responds with $c^* \leftarrow Enc_k(m_b)$.*

Guess: *The adversary $\mathcal{A}$ produces a bit b', and wins if $b = b'$.*

Definition 4 (One-way secure). *An asymmetric encryption scheme $\Pi = (Gen, Enc, Dec)$ is one-way secure if no **quantum** polynomial time adversary $\mathcal{A}$ can win in the $PubK^{OW}_{\mathcal{A},\Pi}(n)$ game, except with probability at most $\mathsf{negl}(n)^{asy}$:*

$PubK^{OW}_{\mathcal{A},\Pi}(n)$ game:
Key Gen: *The challenger runs $Gen(1^n)$ to obtain a pair of keys (pk, sk).*

Challenge Query: *The challenger picks a uniformly random x from the message space, i.e., $x \xleftarrow{\$} \mathtt{MSP}$, and encrypts it using the encryption algorithm Enc_{pk} to obtain the ciphertext $y \leftarrow Enc_{pk}(x)$, and sends y to the adversary $\mathcal{A}$.*

Guess: *The adversary $\mathcal{A}$ on input (pk, y) produces a bit string x', and wins if $x' = x$.*

In the next definition, we say that the quantum algorithm $\mathcal{A}$ has quantum access to the random oracle H if $\mathcal{A}$ can submit queries in superposition and the oracle H answers to these queries by applying a unitary transformation that maps $|x,\ y\rangle$ to $|x,\ y \oplus H(x)\rangle$.

Definition 5 (IND-CCA in the quantum random oracle model). *An asymmetric encryption scheme $\Pi^{asy} = (Gen, Enc, Dec)$ is IND-CCA secure if no **quantum** polynomial time adversary $\mathcal{A}$ can win in the $PubK^{CCA-QRO}_{\mathcal{A},\Pi}(n)$ game, except with probability at most 1/2 + $\mathsf{negl}(n)$:*

$PubK^{CCA-QRO}_{\mathcal{A},\Pi}(n)$ game:
Key Gen: *The challenger runs $Gen(1^n)$ to obtain a pair of keys (pk, sk) and chooses random oracles.*

Query: *The adversary $\mathcal{A}$ is given the public key pk and with **classical** oracle access to the decryption oracle and **quantum** access to the random oracles chooses two messages m_0, m_1 of the same length and sends them to the challenger. The challenger chooses $b \xleftarrow{\$} \{0,1\}$ and responds with $c^* \leftarrow Enc_{pk}(m_b)$.*

Guess: *The adversary $\mathcal{A}$ continues to query the decryption oracle and the random oracles, but may not query the ciphertext c^* in a decryption query. Finally, the*

adversary $\mathcal{A}$ produces a bit b', and wins if $b = b'$.

2.2 Quantum accessible random oracles

In this section, we present some existing results about random oracles that we need to prove the security of our scheme.

Lemma 1 (One way to hiding (O2H) [14]). *Let $H : \{0,1\}^n \to \{0,1\}^m$ be a random oracle. Consider an oracle algorithm A_1 that makes at most q_1 queries to H. Let C be an oracle algorithm that on input x does the following: pick $i \xleftarrow{\$} \{1,\ldots,q_1\}$ and $y \xleftarrow{\$} \{0,1\}^m$, run $A_1^H(x,y)$ until (just before) the i-th query, measure the argument of the query in the computational basis, and output the measurement outcome. (When A_1 makes less than i queries, C outputs $\perp \notin \{0,1\}^n$.)*
Let

$$P_A^1 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^m), x \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^H(x, H(x))]$$

$$P_A^2 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^m), x \xleftarrow{\$} \{0,1\}^n, y \xleftarrow{\$} \{0,1\}^m, b' \leftarrow A_1^H(x,y)]$$

$$P_C := \Pr[x' = x : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^m), x \xleftarrow{\$} \{0,1\}^n, x' \leftarrow C^H(x,i)]$$

Then

$$\left|P_A^1 - P_A^2\right| \leq 2q_1\sqrt{P_C}.$$

Lemma 2 (One way to hiding, adaptive (O2HA) [13]). *Let $H : \{0,1\}^* \to \{0,1\}^n$ be a random oracle. Consider an oracle algorithm A_0 that makes at most q_0 queries to H. Consider an oracle algorithm A_1 that uses the final state of A_0 and makes at most q_1 queries to H. Let C be an oracle algorithm that on input (j,B,x) does the following: run $A_1^H(x,B)$ until (just before) the j-th query, measure the argument of the query in the computational basis, and output the measurement outcome. (When A_1 makes less than j queries, C outputs $\perp \notin \{0,1\}^\ell$.)*
Let

$$P_A^1 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \leftarrow A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, b' \leftarrow A_1^H(x, H(x||m))]$$

$$P_A^2 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \leftarrow A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^H(x,B))]$$

$$P_C := \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \leftarrow A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} \{1,\cdots,q_1\}, x'||m' \leftarrow C^H(j,B,x)]$$

Then

$$\left|P_A^1 - P_A^2\right| \leq 2q_1\sqrt{P_C} + q_0 2^{-\ell/2+2}.$$

Lemma 3 (Corollary 6 of [12]). *Let $f : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ be a function with min-entropy k. Let $H : \{0,1\}^* \to \{0,1\}^{n_1}$ be a random oracle. Then any quantum algorithm A making q queries to H returns a collision for $f \circ H$ with probability at most $O\left(\frac{q^{9/5}}{2^{k/5}}\right)$.*

3 The hybrid scheme and its security

In this section, we combine an asymmetric encryption scheme with a symmetric encryption scheme by using three hash functions in order to gain an IND-CCA secure public encryption scheme $\Pi^{hy} = (Gen^{hy}, Enc^{hy}, Dec^{hy})$ in the quantum random oracle model.

Let $\Pi^{asy} = (Gen^{asy}, Enc^{asy}, Dec^{asy})$ be an asymmetric encryption scheme with the message space $\mathtt{MSP}^{asy} = \{0,1\}^{n_1}$ and the coin space $\mathtt{COIN}^{asy} = \{0,1\}^{n_2}$. Let $\Pi^{sy} = (Enc^{sy}, Dec^{sy})$ be a symmetric encryption scheme where $\mathtt{MSP}^{sy}$ and $\mathtt{KSP}^{sy} = \{0,1\}^m$ are its message space and key space, respectively. The parameters n_1, n_2 and m depend on the security parameter n. We define three hash functions:

$$G : \mathtt{MSP}^{asy} \to \mathtt{KSP}^{sy},\ H : \{0,1\}^* \to \mathtt{COIN}^{asy} \text{ and } H' : \mathtt{MSP}^{asy} \to \mathtt{MSP}^{asy}.$$

These hash functions will be modeled as random oracles in the following.

The hybrid scheme $\Pi^{hy} = (Gen^{hy}, Enc^{hy}, Dec^{hy})$ is constructed as follows, with $\mathtt{MSP}^{hy}$ as its message space:

1. Gen^{hy}, the key generation algorithm, on input 1^n runs Gen^{asy} to obtain a pair of keys (pk, sk).
2. Enc^{hy}, the encryption algorithm, on input pk and message $m \in \mathtt{MSP}^{hy} := \mathtt{MSP}^{sy}$ does the following:
   - Select $\delta \xleftarrow{\$} \mathtt{MSP}^{asy}$.
   - Compute $c \leftarrow Enc^{sy}_a(m)$, where $a := G(\delta)$.
   - Compute $e := Enc^{asy}_{pk}(\delta; h)$, where $h := H(\delta \| c)$.
   - Finally, output (e, c, d) as $Enc^{hy}_{pk}(m; \delta)$, where $d := H'(\delta)$.
3. Dec^{hy}, the decryption algorithm, on input sk and ciphertext (e, c, d) does the following:
   - Compute $\hat{\delta} := Dec^{asy}_{sk}(e)$.
   - If $\hat{\delta} = \perp$: abort and output $\perp$.
   - Otherwise set $\hat{h} := H(\hat{\delta} \| c)$.
   - If $e \neq Enc^{asy}_{pk}(\hat{\delta}; \hat{h})$: abort and output $\perp$.
   - Else if $d = H'(\hat{\delta})$:
     - Compute $\hat{a} := G(\hat{\delta})$ and output $Dec^{sy}_{\hat{a}}(c)$.
   - Else output $\perp$.

Note that our construction is the same as the Fujisaki-Okamoto construction, except that we use an extra random oracle H'. Consequently, the ciphertext has one more component, the encryption algorithm has an additional instruction to compute $H'(\delta)$ and the decryption algorithm has an additional check corresponding to H'.

Theorem 1. *The hybrid scheme Π^{hy} constructed above is IND-CCA secure in the quantum random oracle model if Π^{sy} is an one-time secure symmetric encryption scheme and Π^{asy} is a well-spread one-way secure asymmetric encryption scheme.*

Proof. Let A_{hy} be a quantum polynomial time adversary that attacks Π^{hy} in the sense of IND-CCA in the quantum random oracle model. Suppose that A_{hy} makes at most q_H, q_G and $q_{H'}$ quantum queries to the random oracles H, G and H', respectively, and q_{dec} classical decryption queries. Set $q_{hy} := q_H+q_G+q_{H'}+q_{dec}+1$, i.e., the total number of queries that the adversary A_{hy} may make, including the challenge query. Let Ω_H, Ω_G, $\Omega_{H'}$ be the set of all function $H : \{0,1\}^* \to \{0,1\}^{n_2}$, $G : \{0,1\}^{n_1} \to \{0,1\}^m$ and $H' : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$, respectively. The following game shows the chosen ciphertext attack by the adversary A_{hy} in the quantum setting where the adversary A_{hy} has quantum access to the random oracles H, G and H' and classical access to the decryption algorithm Dec^{hy}.

Game 0:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $(pk, sk) \leftarrow Gen^{asy}(1^n)$
let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{hy}}(pk)$
let $b \xleftarrow{\$} \{0,1\}$, $c^* \leftarrow Enc^{sy}_{G(\delta^*)}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$,
$d^* := H'(\delta^*)$
let $b' \leftarrow A_{hy}^{H,G,H',Dec^{hy}}(e^*, c^*, d^*)$
return $[b = b']$

In order to show that the success probability of Game 0 is at most $1/2+\mathsf{negl(n)}$, we shall introduce a sequence of games and compute the difference between their success probabilities. For simplicity, we omit the definitions of random variables that appear with the same distribution and without any changes in all of the following games. These random variables are: $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}, (pk, sk) \leftarrow Gen^{asy}(1^n)$, and $b \xleftarrow{\$} \{0,1\}$.

In the next game, we replace the decryption algorithm Dec^{hy} with Dec^* where Dec^* on (e, c, d) does the following:

1. If e^* is defined and $e = e^*$: abort and return $\perp$.
2. Else do:
   - Compute $\hat{\delta} := Dec^{asy}_{sk}(e)$.
   - If $\hat{\delta} = \perp$: query $H'(\delta^* \oplus 1)$,[1] abort and output $\perp$.
   - Otherwise set $\hat{h} := H(\hat{\delta} \| c)$.
   - If $e \neq Enc^{asy}_{pk}(\hat{\delta}; \hat{h})$: query $H'(\delta^* \oplus 1)$,[1] abort and output $\perp$.
   - Else if $d = H'(\hat{\delta})$: compute $\hat{a} := G(\hat{\delta})$ and output $Dec^{sy}_{\hat{a}}(c)$.
   - Else: output $\perp$.

[1] This extra query is needed later to prove that Game 4 and Game 5 are identical.

Therefore, Game 1 is as follows:

Game 1:

let $H' \xleftarrow{\$} \Omega_{H'}$
let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
let $c^* \leftarrow Enc^{sy}_{G(\delta^*)}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$
let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, H'(\delta^*))$
return $[b = b']$

We prove that the probabilities of success in Game 0 and Game 1 have negligible difference. We can conclude the result by the fact that the asymmetric encryption scheme is well-spread. We present the proof of the following lemma in Section 4.

Lemma 4. *If the asymmetric encryption scheme Π^{asy} is well-spread, then*

$$\Big|\Pr[1 \leftarrow Game\ 0] - \Pr[1 \leftarrow Game\ 1]\Big| \leq O\left(\frac{(q_H + q_{dec} + 1)^{9/5}}{2^{\omega(\log(n))/5}}\right) =: \ell(n).$$

It is clear that $\ell(n)$ is a negligible function and as a result Game 0 and Game 1 have negligible difference.

We replace $G(\delta^*)$ and $H'(\delta^*)$ with random elements in the next game.

Game 2:

let $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
let $c^* \leftarrow Enc^{sy}_{a^*}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$
let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
return $[b = b']$

Now, we can prove that $\Pr[1 \leftarrow Game\ 2] = 1/2 + \mathsf{negl}(\mathsf{n})^{sy}$. This follows from the one-time security assumption of the symmetric encryption scheme. We postpone the detailed proof of the following lemma to Section 4 in favor of having a simple proof.

Lemma 5. *If the symmetric encryption scheme Π^{sy} is one-time secure, then* $\Pr[1 \leftarrow Game\ 2] = 1/2 + \mathit{negl}(n)^{sy}$.

By using Lemma 5, we only need to show that the difference between the success probabilities of Game 1 and Game 2 is negligible.

Note that if we were in the classical random oracle setting, we could define the **bad** event to be querying G or H' on input δ^* and argue that the two games are indistinguishable until the bad event happens. However, there is no

well-defined concept for the bad event when the adversary A can query G and H' in superposition and each quantum query can contain δ^* in some sense. Therefore, we use the O2H Lemma 1 to obtain an upper bound for $\Big|\Pr[1 \leftarrow \mathit{Game}\ 1]$ - $\Pr[1 \leftarrow \mathit{Game}\ 2]\Big|$.

Let $A^{G\times H'}$ be an adversary that has quantum access to random oracle $G \times H'$ $\Big($where $(G \times H')(\delta) := \big(G(\delta), H'(\delta)\big)\Big)$. The adversary $A^{G\times H'}$ on input $\big(\delta^*, (a^*, d^*)\big)$ does the following:

The adversary $A^{G\times H'}\big(\delta^*, (a^*, d^*)\big)$:

let $H \xleftarrow{\$} \Omega_H, (pk, sk) \leftarrow \mathit{Gen}^{asy}(1^n)$, $b \xleftarrow{\$} \{0,1\}$
let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
let $c^* \leftarrow \mathit{Enc}_{a^*}^{sy}(m_b)$, $e^* \leftarrow \mathit{Enc}_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
return $[b = b']$

Note that the adversary $A^{G\times H'}$ makes at most $q_{o2h} := q_G + q_{H'} + 2q_{dec}$ queries to the random oracle $G \times H'$ in order to respond to the A_{hy}-queries.[2]

Let C be an oracle algorithm that on input δ^* does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$ and $(a^*, d^*) \xleftarrow{\$} \mathtt{KSP}^{sy} \times \mathtt{MSP}^{asy}$, run $A^{G\times H'}\big(\delta^*, (a^*, d^*)\big)$ until (just before) the i-th query, measure the argument of the $G \times H'$-query in the computational basis, output the measurement outcome (when $A^{G\times H'}$ makes less than i queries, C outputs $\perp \notin \{0,1\}^{n_1}$). Note that with this definition we have $P_A^1 = \Pr[1 \leftarrow \mathit{Game}\ 1]$ and $P_A^2 = \Pr[1 \leftarrow \mathit{Game}\ 2]$ where P_A^1 and P_A^2 are defined in O2H Lemma 1 for the adversary $A^{G\times H'}$. Therefore, we will define Game 3 such that $P_C = \Pr[1 \leftarrow \mathit{Game}\ 3]$ where P_C is defined in O2H Lemma 1 for the adversary $C^{G\times H'}$. Thus by O2H Lemma 1:

$$\Big|\Pr[1 \leftarrow \mathit{Game}\ 1] - \Pr[1 \leftarrow \mathit{Game}\ 2]\Big| \leq 2q_{o2h}\sqrt{Pr[1 \leftarrow \mathit{Game}\ 3]}.$$

We define Game 3 as follows:

[2]For example, to respond to a query to the random oracle G with input register I and output register O, the adversary $A^{G\times H'}$ prepares an additional register T (for the output of H') in state $|+\rangle^{n_1}$ and invokes $U_{G\times H'}$ on I, O, T. It is easy to verify that this leaves T unchanged and applies U_G to I, O. (This idea was already used in [18] to ignore part of the output of an oracle.)

Game 3:

let $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
  let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
return $[\tilde{\delta} = \delta^*]$

In the next game, we replace the random oracle H' with a $2(q_{H'} + q_{dec})$-wise independent function. Random polynomials of degree $2(q_{H'} + q_{dec}) - 1$ over finite field $GF(2^{n_1})$ are $2(q_{H'} + q_{dec})$-wise independent. Let Ω_{wise} be the set of all such polynomials.

Game 4:

let $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
  let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
return $[\tilde{\delta} = \delta^*]$

Due to a result by Zhandry [17], a $2(q_{H'} + q_{dec})$-wise independent function H' is perfectly indistinguishable from a random function when the adversary makes at most $q_{H'} + q_{dec}$ queries to H'. Therefore, Game 3 and Game 4 are identical.

We replace the decryption algorithm Dec^* with a new decryption algorithm Dec^{**} in Game 5. Dec^{**} has access to the description (as a polynomial) of H'. Dec^{**} on input (e, c, d) works as follows:

1. If e^* is defined and $e = e^*$: output $\perp$.
2. Else do:
   - Calculate all roots of the polynomial $H' - d$. Let S be the set of those roots.
   - If there exists $\hat{\delta} \in S \setminus \{\delta^*\}$ such that $e = Enc_{pk}^{asy}\big(\hat{\delta}; H(\hat{\delta} \| c)\big)$:
     - query H' on input $\hat{\delta}$.
     - compute $\hat{a} := G(\hat{\delta})$ and return $Dec_{\hat{a}}^{sy}(c)$.
   - Else if $e = Enc_{pk}^{asy}\big(\delta^*; H(\delta^* \| c)\big)$:
     - If $H'(\delta^*) = d$, then compute $\hat{a} := G(\delta^*)$ and return $Dec_{\hat{a}}^{sy}(c)$.
     - Else: return $\perp$.
   - Else: query H' on random input $\delta \xleftarrow{\$} (\mathtt{MSP}^{asy} \setminus \{\delta^*\})$, and output $\perp$.

Note that Dec^{**} depends on the randomness used in choosing H'. This is formally unproblematic (it is comparable to Dec^{**} implicitly depending on secret key) and appears only in intermediate game within the proof. We emphasise that finding roots of polynomial $H' - d$ is possible in polynomial time [5] and it does not involve query to the polynomial H'. (We need that Dec^{**} as well as all other parts of our games run in polynomial time because we want to use the one-way security of the asymmetric encryption scheme in Lemma 6 below.)

Game 5:

let $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{**}}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
  let $b' \leftarrow A_{hy}^{H,G,H',Dec^{**}}(e^*, c^*, d^*)$
measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
return $[\tilde{\delta} = \delta^*]$

In order to show that Game 4 and Game 5 are identical, we need to prove that the two decryption algorithms Dec^* and Dec^{**} return the same output. Also, note that Game 4 and Game 5 succeed if they measure a query containing the argument δ^*. Therefore, we have to prove that the total number of queries submitted to the random oracles G and H' are equal in two decryption algorithms and the number of queries with argument δ^* are equal and appear at the same time.

Suppose the adversary submits a decryption query (e, c, d). Let $\hat{\delta} := Dec_{sk}^{asy}(e)$. We consider the following cases:

1. If $\hat{\delta} = \perp$: In this case, both decryption algorithms return $\perp$ and query the random oracle H', but not on input δ^*.
2. If $\hat{\delta} \neq \perp$, $\hat{\delta} \neq \delta^*$ and $H'(\hat{\delta}) \neq d$: Note that $\hat{\delta} \neq \delta^*$ implies that $e \neq e^*$ and $e \neq Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c))$. Therefore, there are two subcases:
   (a) If $e \neq Enc_{pk}^{asy}(\hat{\delta}; H(\hat{\delta} \parallel c))$, then the decryption algorithm Dec^* queries the random oracle H' on input $\delta^* \oplus 1$ and the decryption algorithm Dec^{**} queries H' on a random element from $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$ since $\hat{\delta} \notin S$. Both algorithms return $\perp$.
   (b) Else, the decryption algorithm Dec^* queries random oracle H' on input $\hat{\delta}$ and the decryption algorithm Dec^{**} queries H' on a random element from $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$ since $\hat{\delta} \notin S$. Both algorithms return $\perp$.
3. If $\hat{\delta} \neq \perp, \hat{\delta} \neq \delta^*$ and $H'(\hat{\delta}) = d$: Note that $\hat{\delta} \neq \delta^*$ implies that $e \neq e^*$ and $e \neq Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c))$. Therefore, there are two subcases:
   (a) If $e \neq Enc_{pk}^{asy}(\hat{\delta}; H(\hat{\delta} \parallel c))$, then the decryption algorithm Dec^* queries the random oracle H' on input $\delta^* \oplus 1$ and outputs $\perp$, and the decryption algorithm Dec^{**} queries H' on a random element from $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$ and outputs $\perp$.

(b) Else, both decryption algorithms query random oracles G and H' on input $\hat{\delta}$ and output $Dec^{sy}_{G(\hat{\delta})}$.

4. If $\hat{\delta} = \delta^*$ and $H'(\hat{\delta}) \neq d$: There are three subcases:
   (a) If e^* is defined and $e = e^*$: Then both decryption algorithms return $\perp$ without any query to the random oracles G and H'.
   (b) Else if $e \neq Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c))$: Then the decryption algorithm Dec^* queries the random oracle H' on input $\delta^* \oplus 1$ and the decryption algorithm Dec^{**} queries H' on a random element from $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$. Both decryption algorithms return $\perp$.
   (c) Else, both decryption algorithms query H' on input δ^* and output $\perp$.
5. If $\hat{\delta} = \delta^*$ and $H'(\hat{\delta}) = d$: There are three subcases:
   (a) If e^* is defined and $e = e^*$: Then both decryption algorithms return $\perp$ without any query to the random oracles G and H'.
   (b) Else if $e \neq Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c))$: Then the decryption algorithm Dec^* queries the random oracle H' on input $\delta^* \oplus 1$ and decryption algorithm Dec^{**} queries H' on a random element from $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$. Both decryption algorithms return $\perp$.
   (c) Else, both decryption algorithms query random oracles G and H' on input δ^* and output $Dec^{sy}_{G(\delta^*)}$.

Hence, $\Pr[1 \leftarrow \mathit{Game}\ 4] = \Pr[1 \leftarrow \mathit{Game}\ 5]$.

Note that Dec^{**} does not use the secret key of the asymmetric encryption scheme to decrypt the ciphertext. This will allow us below to make use of the one-way security of Π^{asy} (This is only possible if the secret key is never used).

The next step is to replace the random coins $H(\delta^* \| c^*)$ of the asymmetric encryption scheme by truly random coins from $\mathtt{COIN}^{asy}$.

Game 6:

let $H' \xleftarrow{\$} \Omega_{wise}$ $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A^{H,G,H',Dec^{**}}_{hy}(pk)$
  let $c^* \leftarrow Enc^{sy}_{a^*}(m_b), e^* \leftarrow Enc^{asy}_{pk}(\delta^*$)
  let $b' \leftarrow A^{H,G,H',Dec^{**}}_{hy}(e^*, c^*, d^*)$
measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
return $[\tilde{\delta} = \delta^*]$

Suppose that adversary A_{hy} makes $q_{0GH'}$ queries to the random oracle $G \times H'$ before the challenge query and $q_{1GH'}$ queries after the challenge query. In order to obtain an upper bound for $\Big|\Pr[1 \leftarrow \mathit{Game}\ 5] - \Pr[1 \leftarrow \mathit{Game}\ 6]\Big|$, we use O2HA Lemma 2. Let A^H_0 be a quantum adversary that has oracle access to the random oracle H. The adversary A^H_0 does the following:

The adversary A_0^H:

let $G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{wise}$, $(pk, sk) \leftarrow Gen^{asy}(1^n)$, $b \xleftarrow{\$} \{0,1\}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{**}}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$
return c^*

Let A_1^H be an adversary that has quantum access to the random oracle H and can use the final state of A_0^H. Therefore, he can access all the random variables that are chosen by A_0^H and also he can use the output of A_0^H. The adversary A_1^H on input (δ^*, h^*) does the following:

The adversary $A_1^H(\delta^*, h^*)$:

let $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
if $i > q_{0GH'}$ **then**
  run until $(i - q_{0GH'})$*-th query to oracle* $G \times H'$
    let $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; h^*)$
    let $b' \leftarrow A_{hy}^{H,G,H',Dec^{**}}(e^*, c^*, d^*)$
measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
return $[\tilde{\delta} = \delta^*]$

Note that the adversary A_0^H may be stopped before receiving the challenge query (or when $i \leq q_{0GH'}$), in this case the adversary A_1^H measures the argument $\tilde{\delta}$ of i-th query to the random oracle $G \times H'$ and outputs $[\tilde{\delta} = \delta^*]$. If $i > q_{0GH'}$, then the adversary A_1^H continues to run the adversary A_{hy} till the $(i - q_{0GH'})$-th query to the random oracle $G \times H'$ and measures the argument $\tilde{\delta}$ of i-th query to the random oracle $G \times H'$ and outputs $[\tilde{\delta} = \delta^*]$. Note that with these definitions we have $P_A^1 = \Pr[1 \leftarrow Game\ 5]$ and $P_A^2 = \Pr[1 \leftarrow Game\ 6]$ where P_A^1 and P_A^2 are as in the O2HA Lemma 2 for the random oracle H.

A_0^H makes q_0 queries to the random oracle H, and A_1^H makes at most q_1 queries to the random oracle H. Let C be an oracle algorithm that on input δ^* does the following: pick $j \xleftarrow{\$} \{1, \ldots, q_1\}$ and $h^* \xleftarrow{\$} \{0,1\}^{n_2}$, run $A_1^H\left(\delta^*, h^*\right)$ until (just before) the j-th query to the random oracle H, measure the argument of that query in the computational basis, output the measurement outcome (when A_1^H makes less than j queries, C outputs $\perp \notin \{0,1\}^n$). Now, we can introduce Game 7 such that by O2HA Lemma 2,

$$\left|\Pr[1 \leftarrow Game\ 5] \text{ - } \Pr[1 \leftarrow Game\ 6]\right| \leq 2q_1\sqrt{Pr[1 \leftarrow Game\ 7]} + q_0 2^{-n_1/2+2}.$$

Game 7:

let $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
run until *i-th query to oracle* $G \times H'$
  let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{**}}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$
let $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $j \xleftarrow{\$} \{1, \ldots, q_1\}$
run until *j-th query to oracle* H
  if $i > q_{0GH'}$ **then**
    run until $(i - q_{0GH'})$*-th query to oracle* $G \times H'$
      let $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; h^*)$
      let $b' \leftarrow A_{hy}^{H,G,H',Dec^{**}}(e^*, c^*, d^*)$
  measure the argument $\tilde{\delta}$ of the i-th query to oracle $G \times H'$
measure the argument $\hat{\delta}||\hat{c}$ of the j-th query to oracle H
return $[\hat{\delta} = \delta^*] \wedge [\hat{c} = c^*]$

The next lemma shows that the success probabilities in Game 6 and Game 7 are negligible. We present the proof of the lemma in Section 4.

Lemma 6. *If the asymmetric scheme* Π^{asy} *is one-way secure then*

$$Pr[1 \leftarrow Game\ 6] \leq \mathsf{negl}(n)^{asy} \text{ and } Pr[1 \leftarrow Game\ 7] \leq \mathsf{negl}(n)^{asy}.$$

Combining this with the bounds derived above we can conclude that

$$Pr[1 \leftarrow Game\ 0] \leq \frac{1}{2} + \mathsf{negl}(n)^{sy} + O\left(\frac{(q_H + q_{dec} + 1)^{9/5}}{2^{\omega(\log(n))/5}}\right) + 2q_{o2h}\sqrt{\mathsf{negl}(n)^{asy} + 2q_1\sqrt{\mathsf{negl}(n)^{asy}}} + q_0 2^{-n_1/2+2}.$$

□

4 Deferred proofs

4.1 Proof of Lemma 4

Proof. We list all the possibilities that the adversary can do to differentiate between the two games. Suppose that the adversary sends the ciphertext (e, c, d). Note that if $e \neq e^*$ or e^* is not defined, then two decryption algorithms Dec^{hy} and Dec^* return the same output and nothing is left to show. Therefore we analyze the following cases where e^* is defined and $e = e^*$.

1. $(e = e^*, c = c^*, d \neq d^*)$ or $(e = e^*, c \neq c^*, d \neq d^*)$: In these two cases, the two decryption algorithms return $\perp$.

2. $(e = e^*, c \neq c^*, d = d^*)$: This means that $Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c)) = Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$. This is a collision in the sense of Lemma 3 since δ^* is chosen randomly and the $Enc_{pk}^{asy}(\delta^*; H(\delta^* \| \cdot))$ has min-entropy $\omega(\log(n))$. Therefore, it occurs with probability at most $O\left(\frac{(q_H+q_{dec}+1)^{9/5}}{2^{\omega(\log(n))/5}}\right)$.
3. $(e = e^*, c = c^*, d = d^*)$. This query never occurs.

We can conclude that:

$$\left|\Pr[1 \leftarrow Game\ 0] \text{ - } \Pr[1 \leftarrow Game\ 1]\right| \leq O\left(\frac{(q_H+q_{dec}+1)^{9/5}}{2^{\omega(\log(n))/5}}\right).$$

□

4.2 Proof of Lemma 5

Proof. Let $\varepsilon(n) := \Pr[1 \leftarrow Game\ 2]$. We construct the adversary A^{sy} such that:

$$\Pr[PriK_{A^{sy},\Pi^{sy}}^{OT} = 1] = \varepsilon(n).$$

The adversary A^{sy} on input 1^n does the following:

1. Run $Gen^{asy}(1^n)$ to obtain (pk, sk).
2. Run the adversary $A_{hy}(pk)$.
3. Use a $2(q_H + q_{dec} + 1)$-wise independent function, a $2(q_G + q_{dec})$-wise independent function, and a $2(q_{H'} + q_{dec})$-wise independent function to answer the queries submitted to the random oracles H, G and H', respectively.
4. Whenever A_{hy} outputs challenge messages (m_0, m_1), do the following:
   - Select $b \xleftarrow{\$} \{0,1\}$, $r \xleftarrow{\$} \mathtt{COIN}^{sy}$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $a^* \leftarrow \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \{0,1\}^{n_1}$.
   - Set $c^* := Enc_{a^*}^{sy}(m_b; r)$ and $e^* := Enc_{pk}^{asy}(\delta^*; H(\delta^*, c^*))$.
   - Send (e^*, c^*, d^*) to the adversary A_{hy}.
5. Answer the random oracle queries and decryption queries as before.
6. When A_{hy} returns bit b', output the same bit b'.

It is obvious that $\Pr[PriK_{A^{sy},\Pi^{sy}}^{OT} = 1] = \varepsilon(n)$. Therefore, $\varepsilon(n) \leq 1/2 + \mathsf{negl(n)}^{sy}$.

□

4.3 Proof of Lemma 6

As the proof for two games is similar we provide the instances for Game 7 in brackets $[\![\ldots]\!]$ wherever there is a difference.

Proof. Let $\varepsilon(n) := \Pr[1 \leftarrow Game\ 6]$ $[\![:= \Pr[1 \leftarrow Game\ 7]]\!]$. We construct an adversary A^{asy} such that:

$$\Pr[PubK_{A^{asy},\Pi^{asy}}^{OW} = 1] = \varepsilon(n).$$

The adversary A^{asy} on input $(1^n, pk, y)$ does the following:

1. Run the adversary $A_{hy}(pk)$.

2. Use a $2(q_H+q_{dec})$-wise independent function, a $2(q_G+q_{dec})$-wise independent function, and a polynomial of degree $2(q_{H'}+q_{dec})-1$ to answer the queries submitted to random oracles H, G and H', respectively.
3. Answer the decryption queries using Dec^{**}.
4. Whenever A_{hy} outputs challenge messages (m_0, m_1), do the following:
   - Select $b \xleftarrow{\$} \{0,1\}$, $r \xleftarrow{\$} \mathtt{COIN}^{sy}$, $a^* \leftarrow \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \{0,1\}^{n_1}$.
   - Set $c^* := Enc^{sy}_{a^*}(m_b; r)$ and $e^* := y$.
   - Send (e^*, c^*, d^*) to the adversary A_{hy}.
5. Answer the random oracle queries as before and to the decryption queries using Dec^{**}.
6. When A_{hy} returns bit b' and halts, A^{asy} selects $i \xleftarrow{\$} \{1, \cdots, q_{o2h}\}$ $[\![i \xleftarrow{\$} \{1, \cdots, q_1\}]\!]$ and measures the argument $\hat{\delta}$ of i-th $[\![(i+q_0)$-th $]\!]$ query to the random oracle $G \times H'$ $[\![H]\!]$ and outputs $\hat{\delta}$ (When A_{hy} makes less than i queries output $\perp$).

It is obvious that $\Pr[PubK^{OW}_{A^{asy}, \Pi^{asy}} = 1] = \varepsilon(n)$. Therefore, $\varepsilon(n) \leq \mathsf{negl(n)}^{asy}$. □

5 A variant of OAEP

The following definitions are similar to the definitions presented in [9], except we define them in the presence of a **quantum** adversary.

Definition 6 (Quantum partial-domain one-way function). *We say a function $f : \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \to \{0,1\}^m$ is partial-domain one-way if for any polynomial time quantum adversary A,*

$$\Pr[\tilde{s} = s : s \xleftarrow{\$} \{0,1\}^{n+k_1},\ t \xleftarrow{\$} \{0,1\}^{k_0},\ \tilde{s} \leftarrow A(f(s,t))] \leq \mathit{negl(n)}.$$

Definition 7. *Let $G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}$, $H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}$ and $H' : \{0,1\}^k \to \{0,1\}^k$ be random oracles. The Q-OAEP $= (Gen, Enc, Dec)$ encryption scheme is defined as:*

1. **Gen***: Specifies an instance of the injective function f and its inverse f^{-1}. Therefore, the public key and secret key are f and f^{-1} respectively.*
2. **Enc***: Given a message $m \in \{0,1\}^n$, the encryption algorithm computes*

$$s := m||0^{k_1} \oplus G(r) \quad \textit{and} \quad t := r \oplus H(s),$$

   where $r \xleftarrow{\$} \{0,1\}^{k_0}$, and outputs the ciphertext $(c,d) := \Big(f(s,t), H'(s\|t)\Big)$.
3. **Dec***: Given a ciphertext (c,d), the decryption algorithm does the following:*
   - *When $c \notin \operatorname{Im} f$:*
     (a) *If c^* is defined (where c^* is the challenge ciphertext), then query the random oracle H' on input $(s^*\|t^*) \oplus 1$ (where $f(s^*, t^*) = c^*$) and return $\perp$.*

(b) *If c^* is not defined, then query the random oracle H' on a random input and return $\perp$.*

– *When $c \in \mathrm{Im}\, f$, the decryption algorithm extracts $(s,t) = f^{-1}(c)$. If $H'(s\|t) \neq d$ it returns $\perp$, otherwise it does the following:*
  (a) *query the random oracle H on input s and compute $r := t \oplus H(s)$.*
  (b) *query the random oracle G on input r and compute $M := s \oplus G(r)$.*
  (c) *if the k_1 least significant bits of M are zero then return the n most significant bits of M, otherwise return $\perp$.*

Note that k_0 and k depend on the security parameter n.

Note that *Dec* contains several unnecessary oracle calls (after it already decided to output $\perp$). These obviously do not effect correctness or security, but make the proof a bit simple to formulate.

Theorem 2. *If the underlying injective function is quantum partial-domain one-way, then the Q-OAEP scheme is IND-CCA secure in the quantum random oracle model.*

Proof. Since the proof is similar and relatively easier compared to the proof of Fujisaki-Okamoto transform, we only present the main games in pseudocode and the intuition of the their negligibility. Let Ω_H, Ω_G, $\Omega_{H'}$ be the set of all function $H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}$, $G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}$ and $H' : \{0,1\}^{k} \to \{0,1\}^{k}$, respectively. Let A be a polynomial time quantum adversary that attacks the OAEP-cryptosystem in the sense of IND-CCA in the quantum random oracle model and makes at most q_H, q_G and $q_{H'}$ queries to the random oracles H, G and H' respectively and q_{dec} decryption queries.

Game 0:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$
let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus G(r)$, $t^* := r \oplus H(s^*)$, $c^* := f(s^*, t^*)$, $d^* := H'(s^*\|t^*)$
let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
return $[b = b']$

Game 1:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$, $\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$
let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
let $b \xleftarrow{\$} \{0,1\}$, $s^* = m_b||0^{k_1} \oplus \alpha^*$, $t^* = r \oplus H(s^*)$, $c^* = f(s^*, t^*)$, $d^* := H'(s^*\|t^*)$
let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
return $[b = b']$

The probability of success in Game 1 is $1/2$ for the reason that s^* is a random element and independent of the bit b.

Game 2:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$
run until *i-th query to oracle* G
  let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
  let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* := f(s^*, t^*)$,
  $d^* := H'(s^* \| t^*)$
  let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
measure the argument $\tilde{r}$ of the i-th query to oracle G
return $[\tilde{r} = r]$ (When A makes less than i queries return $\perp$)

By O2H Lemma 1,

$$|\Pr[1 \leftarrow Game\ 0] - \Pr[1 \leftarrow Game\ 1]| \leq 2(q_G + q_{dec})\sqrt{\Pr[1 \leftarrow Game\ 2]}.$$

Game 3:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$
run until *i-th query to oracle* G
  let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
  let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* := f(s^*, t^*)$,
  $d^* := H'(s^* \| t^*)$
  let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
measure the argument $\tilde{r}$ of the i-th query to oracle G
return $[\tilde{r} = r]$ (When A makes less than i queries return $\perp$)

Since t^* and s^* are random and independent of r, the probability of success in Game 3 is $\frac{1}{2^{k_0}}$.

Game 4:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$,
$j \xleftarrow{\$} \{1, \ldots, q_H + q_{dec}\}$
run until *j-th query to oracle H*
  run until *i-th query to oracle G*
    let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
    let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* := f(s^*, t^*)$,
     $d^* := H'(s^* \| t^*)$
    let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
  measure the argument $\tilde{r}$ of the i-th query to oracle G
measure the argument $\tilde{s}$ of the j-th query to oracle H
return $[\tilde{s} = s^*]$ (When A makes less than j queries return $\perp$)

By O2H Lemma 1,

$$|\Pr[1 \leftarrow Game\ 2] - \Pr[1 \leftarrow Game\ 3]| \leq 2(q_H + q_{dec})\sqrt{\Pr[1 \leftarrow Game\ 4]}.$$

Game 5:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$s^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$,
$j \xleftarrow{\$} \{1, \ldots, q_H + q_{dec}\}$, $d^* \xleftarrow{\$} \{0,1\}^k$
run until *j-th query to oracle H*
  run until *i-th query to oracle G*
    let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
    let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* := f(s^*, t^*)$,
    let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
  measure the argument $\tilde{r}$ of the i-th query to oracle G
measure the argument $\tilde{s}$ of the j-th query to oracle H
return $[\tilde{s} = s^*]$ (When A makes less than j queries return $\perp$)

Game 6:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$,
$j \xleftarrow{\$} \{1, \ldots, q_H + q_{dec}\}$, $d^* \xleftarrow{\$} \{0,1\}^k$, $\ell \xleftarrow{\$} \{1, \ldots, q_{H'} + q_{dec}\}$
run until *ℓ-th query to oracle H'*
  run until *j-th query to oracle H*
    run until *i-th query to oracle G*
      let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
      let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* = f(s^*, t^*)$
      let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
    measure the argument $\tilde{r}$ of the i-th query to oracle G
  measure the argument $\tilde{s}$ of the j-th query to oracle H
measure the argument $(\tilde{s}, \tilde{t})$ of the ℓ-th query to oracle H'
return $[\tilde{s} = s^*] \wedge [\tilde{t} = t^*]$ *(When A makes less than ℓ queries return $\perp$)*

By O2H Lemma 1,

$$|\Pr[1 \leftarrow Game\ 4] - \Pr[1 \leftarrow Game\ 5]| \leq 2(q_{H'} + q_{dec})\sqrt{\Pr[1 \leftarrow Game\ 6]}.$$

Therefore, we only need to prove that the probability of success in Game 5 and Game 6 are negligible. Since a $2q$-wise independent function is indistinguishable from a random oracle provided the adversary makes at most q queries [17], we replace H' in Game 5 and Game 6 with a random polynomials of the proper degree. Let Ω_{wise} be the set of all such polynomials.

Game 5.b:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{wise}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$,
$j \xleftarrow{\$} \{1, \ldots, q_H + q_{dec}\}$, $d^* \xleftarrow{\$} \{0,1\}^k$
run until *j-th query to oracle H*
  run until *i-th query to oracle G*
    let $m_0, m_1 \leftarrow A^{H,G,H',Dec}(pk)$
    let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* = f(s^*, t^*)$
    let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
  measure the argument $\tilde{r}$ of the i-th query to oracle G
measure the argument $\tilde{s}$ of the j-th query to oracle H
return $[\tilde{s} = s^*]$ (When A makes less than j queries return $\perp$)

By Zhandry's result [17]:

$$\Pr[1 \leftarrow Game\ 5] = \Pr[1 \leftarrow Game\ 5.b].$$

Now we define the decryption algorithm Dec^* that on input (c, d) does as follows:

1. It calculates the roots of polynomial $H' - d$. Let S be the set of all the roots.

2. If there exists $(s,t) \in S$ such that $f(s,t) = c$, then it outputs a message m using (s,t) and similar to the algorithm Dec. Otherwise it outputs $\perp$.

Game 5.c:

let $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{wise}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$,
$\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec}\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$,
$j \xleftarrow{\$} \{1, \ldots, q_H + q_{dec}\}$, $d^* \xleftarrow{\$} \{0,1\}^k$
run until *j-th query to oracle* H
  run until *i-th query to oracle* G
    let $m_0, m_1 \leftarrow A^{H,G,H',Dec^*}(pk)$
    let $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus \beta^*$, $c^* = f(s^*, t^*)$
    let $b' \leftarrow A^{H,G,H',Dec^*}(c^*, d^*)$
  measure the argument $\tilde{r}$ of the i-th query to oracle G
measure the argument $\tilde{s}$ of the j-th query to oracle H
return $[\tilde{s} = s^*]$ (When A makes less than j queries return $\perp$)

We show that two decryption algorithms Dec and Dec^* return the same output with the same number of queries to the random oracle H. For given ciphertext (c,d):

1. If $c \notin \operatorname{Im} f$, then both decryption algorithms return $\perp$ with no query to the random oracle H.
2. If $c \in \operatorname{Im} f$. Let $(\hat{s}, \hat{t}) := f^{-1}(c)$. There are two subcases:
   - If $H'(\hat{s} \| \hat{t}) \neq d$, then both algorithms return $\perp$ with no query to the random oracle H.
   - If $H'(\hat{s} \| \hat{t}) = d$, then both decryption algorithms return the same output and query H on input $\hat{s}$ for the reason that $(\hat{s}, \hat{t}) \in S$ and $f(\hat{s}, \hat{t}) = c$.

As a result:

$$\Pr[1 \leftarrow Game\ 5.b] = \Pr[1 \leftarrow Game\ 5.c].$$

Note that the decryption algorithm Dec^* does not use the secret key f^{-1}, therefore we can reduce the success probability of Game 5.c to the partial-domain one-wayness of function f.

We repeat a similar approach (define Game 6.b and Game 6.c as before) to prove the success probability of Game 6 is negligible. Note that the decryption algorithm Dec^{**} does as follows in the case of Game 6:

1. It calculates the roots of polynomial $H' - d$. Let S be the set of all the roots.
2. If there exists $(s,t) \in S$ such that $f(s,t) = c$, then it queries the random oracle H' on input $(s \| t)$ and outputs a message m using (s,t) and similar to the algorithm Dec.
3. Else:
   - If c^* is defined and $c = c^*$, then query H' on input $(s^* \| t^*)$ and return $\perp$.

– If c^* is defined and $c \neq c^*$, then query H' on input $(s^* \| t^*) \oplus 1$ and return $\perp$.
– If c^* is not defined then query H' on a random input and return $\perp$.

We show that two decryption algorithms Dec and Dec^{**} return the same output with the same number of queries to the random oracle H'. For given ciphertext (c, d):

1. If $c \notin \operatorname{Im} f$, then both decryption algorithms return $\perp$ and query the random oracle H' on a random input or on input $(s^* \| t^*) \oplus 1$.
2. If $c \in \operatorname{Im} f$ and c^* is defined. Let $(\hat{s}, \hat{t}) := f^{-1}(c)$. Then:
   – If $H'(\hat{s} \| \hat{t}) = d$, then both decryption algorithms return the same output and query H' on input $(\hat{s} \| \hat{t})$.
   – If $H'(\hat{s} \| \hat{t}) \neq d$ and $c \neq c^*$, then both algorithms return $\perp$ and query the random oracle H' on an input different from $(s^* \| t^*)$.
   – If $H'(\hat{s} \| \hat{t}) \neq d$ and $c = c^*$, then both algorithms return $\perp$ and query the random oracle H' on input $(s^* \| t^*)$.
3. If $c \in \operatorname{Im} f$ and c^* is not defined. Let $(\hat{s}, \hat{t}) := f^{-1}(c)$. Then:
   – If $H'(\hat{s} \| \hat{t}) \neq d$, then both algorithms return $\perp$ and query the random oracle H' on an input.
   – If $H'(\hat{s} \| \hat{t}) = d$, then both decryption algorithms return the same output and query H' on input $(\hat{s} \| \hat{t})$.

By combining all the inequalities from the proof, we can conclude that:

$$\Pr[1 \leftarrow \mathit{Game}\ 0] \leq 1/2 + \mathsf{negl}(\mathsf{n}).$$

Since our security proof does not depend on the bit padding, the message space can be extended to the set $\{0,1\}^{n+k_1}$.

□

Acknowledgments. This work was supported by the Estonian ICT program 2011-2015 (3.2.1201.13-0022), the European Union through the European Regional Development Fund through the sub-measure "Supporting the development of R&D of info and communication technology", by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa, by the Estonian Centre of Excellence in Computer Science, EXCS.

References

1. G. Alagic, A. Broadbent, B. Fefferman, T. Gagliardoni, C. Schaffner, and M. S. Jules. Computational security of quantum encryption. IACR ePrint 2016/424, April 2016.
2. A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems (the hardness of quantum rewinding). In *FOCS 2014*, pages 474–483. IEEE, October 2014.

3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73. ACM, 1993.
4. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.
5. M. Ben-Or. Probabilistic algorithms in finite fields. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 394–398. IEEE Computer Society, 1981.
6. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.
7. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Crypto 2013*, 2013. Full version at IACR ePrint 2013/088.
8. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 537–554, London, UK, UK, 1999. Springer-Verlag.
9. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. *J. Cryptology*, 17(2):81–104, 2004.
10. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
11. V. Shoup. OAEP reconsidered. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259. Springer, 2001.
12. E. E. Targhi, G. N. Tabia, and D. Unruh. Quantum collision-resistance of non-uniformly distributed functions. In T. Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, volume 9606 of *Lecture Notes in Computer Science*, pages 79–85. Springer, 2016.
13. D. Unruh. Quantum position verification in the random oracle model. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2014.
14. D. Unruh. Revocable quantum timed-release encryption. In P. Q. Nguyen and E. Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 129–146. Springer, 2014.

15. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015.
16. H. Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Information & Computation*, 14(13-14):1089–1097, 2014.
17. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012.
18. M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.