Quantum Security of the Fujisaki-Okamoto and OAEP Transforms

Ehsan Ebrahimi Targhi, Dominique Unruh

University of Tartu, Estonia

December 18, 2015

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 attack.

1 Introduction

The interest in verifying the security of cryptosystems in the presence of a quantum adversary increased after the celebrated paper of Shor [Sho97]. Shor showed that any cryptosystem based on the factoring problem and the discrete logarithm problem is breakable in the existence of a quantum adversary. Also, many efficient classical cryptosystems are proved to be secure in the random oracle model [BR93] and many of them still lack equivalent proof in the quantum setting. Therefore to construct an efficient cryptosystem secure against quantum adversaries, even if we find a cryptographic primitive immune to quantum attacks, 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 [FO99] constructed a hybrid encryption scheme that is secure against chosen ciphertext attack (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 a classical adversary and it is not clear how one can fix their proof in the quantum setting. Following, we mention the parts of the classical proof that may not follow against quantum adversaries. The classical proof uses the record list of 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, there is no such a list. Also, the classical proof uses the fact that changing output of the random oracle on one random input does not make it distinguishable from the original random oracle and this may not occur in the quantum case as long as the adversary can query the random oracle in superposition of all inputs

and see all corresponding outputs in one query. Finally, the classical proof uses the fact that finding a collision for a function whose outputs have a high min-entropy is difficult with classical access to the function and polynomial number of queries. However, this may not happen when the adversary has quantum access to the function. Consequently, the quantum security of the scheme is left as an open problem in the related works of Boneh et al.[BDF+11] and Zhandry [Zha12].

In 1993, Bellar and Rogaway [BR93] introduced a hybrid encryption scheme secure against chosen ciphertext attack in the random oracle model provided that the trapdoor permutation used in the scheme is one-way. One year later, they proposed another method, named OAEP, for converting a trapdoor permutation into an encryption scheme [BR94]. It was believed that the OAEP-cryptosystem is provable secure in the random oracle model based on one-wayness of trapdoor permutation, but Shoup [Sho01] showed it is an unjustified belief. Later, Fujisaki et al. [FOPS04] proved IND-CCA security of the OAEP-cryptosystem based on a stronger assumption, namely, partial-domain one-wayness of the underlying permutation. Quantum security of the hybrid encryption scheme in [BR93] was proved by Boneh et al. [BDF+11] provided that the underlying injective trapdoor function is quantum-immune. However, they mentioned preimage awareness used in the security proof of the OAEP-cryptosystem as a classical technique that is not known to follow in the quantum setting. A quantum-immune candidate constructed in [PW08] based on lattices.

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 attack in the quantum random oracle model. For message m, the encryption algorithm of our scheme, Enc^{hy}_{pk}, works as follows:

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

where pk and sk are the public key and the secret key of the asymmetric encryption scheme. Enc^{asy}_{pk} and Enc^{sy}_{sk} are the asymmetric and symmetric encryption algorithms respectively. δ is a random element from message space of the asymmetric encryption scheme. H, G and H' are random oracles with proper domain and co-domain. 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 encryption of two messages when a fresh key is used for every encryption. In addition, the asymmetric encryption scheme is well-spread in which any message can lead to at least $2^{\omega(\log n)}$ potential ciphertexts. 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. We present a sketch of the security proof of OAEP-cryptosystem in the appendix since its proof is very similar to the security proof of the Fujisaki-Okamoto transform.

In what follows, we explain how we overcome the challenges that appear in the quantum security proof of the hybrid constructions. Similar to the idea used in [Unr15], we use extra hash H' and later in the security proof we replace it with a random polynomial to force the adversary to submit the input that has been used to obtain the cipertext. This can be done due to result by Zhandry [Zha12] 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 [Unr14a, Unr14b]. Unruh gives an upper bound for any adversary that is trying to distinguish two random oracles that have different output on only one random input. Finally, we use the existing result on

quantum query complexity of finding a collision for an unknown function f whose outputs are drawn according to a distribution with min-entropy k [ETTU15].

2 Preliminaries

Let KSP and MSP stand for the key space and the message space respectively. Notation $x \xleftarrow{\$} X$ shows that x is chosen uniformly at random from 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 $k \in \texttt{KSP}$ and message $m \in \texttt{MSP}$, outputs ciphertext c, $c \leftarrow Enc_k(m)$. The message space can be infinite and it depends 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 $m := Dec_k(c)$. It is required that decryption algorithm returns the original message, $Dec_k(Enc_k(m)) = m$, for every $k \in \texttt{KSP}$ and every $m \in \texttt{MSP}$.

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 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 message $m \in \texttt{MSP}$ and outputs the ciphertext c, $c \leftarrow Enc_{pk}(m)$. The message space, MSP, depends 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 $m := Dec_{sk}(c)$. It is required that decryption algorithm returns the original message, $Dec_{sk}(Enc_{pk}(m)) = m$, for every $(pk, sk) \leftarrow Gen(1^n)$ and every $m \in \texttt{MSP}$. 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 $h \in \texttt{COIN}$ where 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 [FO99])**.** *An asymmetric encryption scheme $\Pi = (Gen, Enc, Dec)$ is γ-spread if for every pk generated by $Gen(1^n)$ and every $x \in \texttt{MSP}$,*

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

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

Definition 2. *We say that function $f : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ 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 $\mathsf{neg(n)}$ be any non-negative function that is smaller than the inverse of any non-negative polynomial $p(n)$ for sufficiently large n. That is, $\lim_{n\to\infty} \mathsf{neg(n)}p(n) = 0$ for polynomial $p(n)$. In the following, we present the security definitions that are needed in this paper. Note that the definitions are the same with the security definitions in [FO99], except they have been represented in the presence of a **quantum** adversary in this paper. As the two following security definitions will be used in the security proof of our scheme, we differentiate between them using $\mathsf{neg}(n)^{sy}$ and $\mathsf{neg}(n)^{asy}$ in the definitions.

Definition 3 (One-Time secure)**.** *A symmetric encryption scheme $\Pi = (Enc, Dec)$ is indistinguishable in the presence of an eavesdropper (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 + $\mathsf{neg}(n)^{sy}$:*

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

***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 $PublK^{OW}_{\mathcal{A},\Pi}(n)$ game, except with probability at most $\mathsf{neg}(n)^{asy}$:*

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

***Challenge Query:** The challenger picks up uniformly at random x from the message space, $x \xleftarrow{\$} \mathtt{MSP}$, and encrypts it by the encryption algorithm Enc_{pk} to obtain the ciphertext y, $y \leftarrow Enc_{pk}(x)$ 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 where $\mathcal{A}$ can submit queries in superposition and the oracle H answers to the queries by 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 indistinguishable under chosen ciphertesxt attack in the quantum random oracle model (IND-CCA secure in QRO) if no efficient **quantum** adversary $\mathcal{A}$ can win in the $PublK^{CCA-QRO}_{\mathcal{A},\Pi}(n)$ game, except with probability at most 1/2 + $\mathsf{neg}(n)$:*

***$PublK^{CCA-QRO}_{\mathcal{A},\Pi}(n)$** 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^* *as 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 that we need to prove the security of our scheme.

Lemma 1 (One way to hiding (O2H) [Unr14b])**.** *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, 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) [Unr14a])**.** *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, 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 [ETTU15])**.** *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 message space $\mathtt{MSP}^{asy} = \{0,1\}^{n_1}$ and 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 followings.

The hybrid scheme $\Pi^{hy} = (Gen^{hy}, Enc^{hy}, Dec^{hy})$ is constructed as follows in which $\mathtt{MSP}^{hy}$ is 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 $c \leftarrow Enc_a^{sy}(m)$, where $a := G(\delta)$.
   - Compute $e := Enc_{pk}^{asy}(\delta; h)$, where $h := H(\delta \parallel c)$.
   - Finally, output (e, c, d) as $Enc_{pk}^{hy}(m; \delta)$, where $d := H'(\delta)$.
3. Dec^{hy}, the decryption algorithm, on input sk and ciphertext (e, c, d) does as follows:
   - Compute $\hat{\delta} := Dec_{sk}^{asy}(e)$.
   - Set $\hat{h} := H(\hat{\delta} \parallel c)$.
   - If $e \neq Enc_{pk}^{asy}(\hat{\delta}; \hat{h})$: abort and output $\perp$.
   - Else if $d = H'(\hat{\delta})$:
     - Compute $\hat{a} := G(\hat{\delta})$ and output $Dec_{\hat{a}}^{sy}(c)$.
   - Else abort and output $\perp$.

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

Theorem 4. *The hybrid scheme Π^{hy} constructed as 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 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 $q_{hy} := q_H + q_G + q_{H'} + q_{dec} + 1$, that is 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 $b \xleftarrow{\$} \{0,1\}$, $c^* \leftarrow Enc^{sy}_{G(\delta^*)}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \parallel c^*))$, $d^* := H'(\delta^*)$
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 $1/2 + \mathtt{neg}(n)$, we shall introduce a sequence of games and compute the difference between their success probability. For simplicity, we remove the definition of random variables that appear with the same probability 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 input sk and ciphertext (e, c, d) does as follows:

1. If e^* is defined and $e = e^*$: abort and return $\perp$.
2. Else do:
   - Compute $\hat{\delta} := Dec^{asy}_{sk}(e)$.
   - Set $\hat{h} := H(\hat{\delta} \parallel c)$.
   - If $e \neq Enc^{asy}_{pk}(\hat{\delta}; \hat{h})$: query $H'(\delta^* \oplus 1)$, 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 $\perp$.

Therefore, the Game 1 is:

Game 1:

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

We prove that the probability of success in Game 0 and Game 1 are in a 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 3.1.

Lemma 5. *If the asymmetric encryption scheme Π^{asy} is well-spread, then*

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

It is clear that $\ell(n)$ is a negligible function and as a result Game 0 and Game 1 are in a 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 $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c^*))$
let $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
return $[b = b']$

Now, one can prove that $\Pr[1 \leftarrow Game\ 2] = 1/2 + \mathsf{neg(n)}^{sy}$. This follows from the One-Time security assumption of the symmetric encryption scheme. We postpone the detailed proof of the following lemma to Section 3.1 in favor of not having a messy proof.

Lemma 6. *If the symmetric encryption scheme Π^{sy} is One-Time secure, then* $\Pr[1 \leftarrow Game\ 2] = 1/2 + \mathit{neg(n)}^{sy}$.

By using Lemma 6, we only need to show that the difference between the success probability of Game 1 and Game 2 is negligible. To achieve our goal, we use the O2H Lemma 1 to obtain an upper bound for $\Big|\Pr[1 \leftarrow Game\ 1] - \Pr[1 \leftarrow Game\ 2]\Big|$.

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 $H \xleftarrow{\$} \Omega_H, (pk, sk) \leftarrow Gen^{asy}(1^n)$, $b \xleftarrow{\$} \{0,1\}$
let $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
let $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c^*))$
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 $q_{o2h} := q_G + q_{H'} + 2q_{dec}$ number of queries to the random oracle $G \times H'$ in order to respond to the A_{hy} queries [1].

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 can say $P_A^1 = \Pr[1 \leftarrow Game\ 1]$ and $P_A^2 = \Pr[1 \leftarrow Game\ 2]$ 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 $P_C = \Pr[1 \leftarrow Game\ 3]$ where P_C is defined in O2H Lemma 1 for the adversary $C^{G\times H'}$. Thus by O2H Lemma 1:

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

[1] For example to respond to a query to the random oracle G on input $|\delta\rangle$, the adversary $A_{G\times H'}$ prepares three registers where first register contains the input, second register stores the output of G and finally third register stores the output of H'. Then, the adversary $A_{G\times H'}$ applies the unitary transformation $(I\otimes I\otimes H^{\otimes n_1})(U_{G\times H'})(I\otimes I\otimes H^{\otimes n_1})$ to the input $|\delta, y, 0^{n_1}\rangle$ to obtain the output $|\delta, G(\delta)\oplus y, 0^{n_1}\rangle$ and sends the first and second wire to the adversary A_{hy}. The same idea applies to answer to the queries submitted to the random oracle H'.

We define Game 3 as follows:

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 $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c^*))$
  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 $2(q_{H'} + q_{dec} + 1)$-wise independent function. Random polynomials of degree $2(q_{H'} + q_{dec} + 1) - 1$ over finite field $GF(2^{n_1})$ of size 2^{n_1} are $2(q_{H'} + q_{dec} + 1)$-wise independent. Let Ω_{wise} be the set of all such polynomials.

Game 4:

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 $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c^*))$
  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 [Zha12], a $2(q_{H'} + q_{dec} + 1)$-wise independent function H' is indistinguishable from a random function when the adversary makes at most $q_{H'} + q_{dec} + 1$ 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 where 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 $e = Enc_{pk}^{asy}\big(\hat{\delta}; H(\hat{\delta} \parallel c)\big)$:
     - query H' on input $\hat{\delta}$.
     - compute $\hat{a} := G(\hat{\delta})$ and return $Dec_{\hat{a}}^{sy}(c)$.
   - Else if $e = Enc_{pk}^{asy}\big(\delta^*; H(\delta^* \parallel c)\big)$:
     - 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 $\perp$.

We emphasis that finding roots of polynomial $H' - d$ is polynomial time computable [Ben81] and it does not involve query to the polynomial H'.

Game 5:

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 $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{**}}(pk)$
  let $c^* \leftarrow Enc_{a^*}^{sy}(m_b), e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c^*))$
  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 two decryption algorithms Dec^* and Dec^{**} return the same output. Also, we have to prove that the total number of queries submitted to the random oracles G and H' are equal in two algorithms and the number of queries with argument δ^* are equal and appear in the same time.

Suppose the adversary submits 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 $e \neq e^*$. Therefore, there are two subcases:

   (a) If $e \neq Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c))$, 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 $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$ 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 $\hat{\delta} = \delta^*$ 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 $e \neq Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c))$, 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 $\mathtt{MSP}^{asy} \setminus \{\delta^*\}$. Both decryption algorithms return $\perp$.

   (c) Else, both decryption algorithms query H' on input δ^* and output $\perp$.

4. 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 $e \neq Enc_{pk}^{asy}(\delta^*; H(\delta^* \parallel c))$, 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 $Dec_{G(\delta^*)}^{sy}$.

5. If $\hat{\delta} \neq \perp, \hat{\delta} \neq \delta^*$ and $H'(\hat{\delta}) = d$. Note that $\delta \neq \delta^*$ implies that $e \neq e^*$. Therefore, there are two subcases:

(a) If $e \neq Enc^{asy}_{pk}(\delta^*; H(\delta^* \parallel c))$, 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 $\texttt{MSP}^{asy} \setminus \{\delta^*\}$ and outputs $\perp$.

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

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

Note that Dec^{**} does not use the secret key of asymmetric encryption scheme to decrypt the ciphertext. Therefore, we replace the $H(\delta^* \parallel c^*)$ with a random element from $\texttt{COIN}^{asy}$, once we will be able to reduce the proof of security to the One-Way security of asymmetric encryption scheme.

Game 6:

let $H' \xleftarrow{\$} \Omega_{wise}$ $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \texttt{KSP}^{sy}$, $d^* \xleftarrow{\$} \texttt{MSP}^{asy}$, $i \xleftarrow{\$} \{1, \ldots, q_{o2h}\}$
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 before challenge query and $q_{1GH'}$ queries after challenge query to the random oracle $G \times H'$. In order to obtain an upper bound for $\Big|\Pr[1 \leftarrow Game\ 5] - \Pr[1 \leftarrow Game\ 6]\Big|$, 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^H_0:

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

Let A^H_1 be an adversary that has quantum access to the random oracle H and use the final sate of A^H_0. Therefore, he can use all the random variables that are chosen by A^H_0 and also he can use the output of A^H_0. The adversary A^H_1 on input (δ^*, h^*) does the following:

The adversary $A^H_1(\delta^*, h^*)$:

let $\delta^* \xleftarrow{\$} \texttt{MSP}^{asy}$
if $i > q_{0GH'}$ **then**
  run until $(i - q_{0GH'})$*-th query to oracle* $G \times H'$
    let $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; h^*)$
    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 $[\tilde{\delta} = \delta^*]$

Note that the adversary A^H_0 may be stopped before receiving the challenge query (or when $i \leq q_{0GH'}$), in this case the adversary A^H_1 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 continue running the adversary A_{hy} till $(i - q_{0GH'})$-th query to the random oracle $G \times H'$ and he 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 O2HA Lemma 2 for random oracle H.

Let 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 $A_1^H(\delta^*, h^*)$ until (just before) the j-th query to the random oracle H, measure the argument of the 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,

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

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 $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}, j \xleftarrow{\$} \{0, \ldots, q_1\}$
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 probability in Game 6 and Game 7 are negligible. We present the proof of the lemma in Section 3.1.

Lemma 7. *If the asymmetric scheme* Π^{asy} *is One-way secure then,*

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

With combining the bounds derived above we can conclude that,

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

□

3.1 Proof of Lemma

Lemma 5.

Proof. We list all the possibilities that the adversary can do to differentiate between two games. Suppose that the adversary sends 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, two decryption algorithms return $\perp$.

2. $(e = e^*, c \neq c^*, d = d^*)$. This means that $Enc^{asy}_{pk}(\delta^*; H(\delta^* \parallel c)) = Enc^{asy}_{pk}(\delta^*; H(\delta^* \parallel c^*))$ and it is a collision in the sense of Lemma 3 since δ^* is chosen randomly and the $Enc^{asy}_{pk}(\delta^*; H(\delta^* \parallel \cdot))$ has min-entropy $\omega(\log(n))$. Therefore, it happens 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 happen.

We can conclude that:

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

□

Lemma 6.

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

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

The adversary A^{sy} is given input 1^n works as following:

1. Runs $Gen^{asy}(1^n)$ to obtain (pk, sk).

2. Runs the adversary $A_{hy}(pk)$.

3. Use $2(q_H + q_{dec} + 1)$-wise independent function, $2(q_G + q_{dec} + 1)$-wise independent function, $2(q_{H'} + q_{dec} + 1)$-wise independent function to answer to the queries submitted to the random oracles H, G and H' respectively.

4. Whenever A_{hy} outputs challenge messages (m_0, m_1), does as follow:

   - Selects $b \xleftarrow{\$} \{0,1\}$, $r \xleftarrow{\$} \mathtt{COIN}^{sy}$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $a^* \leftarrow \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \{0,1\}^{n_1}$.
   - Sets $c^* = Enc^{sy}_{a^*}(m_b; r)$ and $e^* = Enc^{asy}_{pk}(\delta^*; H(\delta^*, c^*))$.
   - Sends (e^*, c^*, d^*) to the adversary A_{hy}.

5. Answer to the oracle queries and decryption queries as before.

6. When A_{hy} returns bit b', outputs the same b'.

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

Lemma 7.

As the proof for two games is similar we provide the instances for Game 7 in brackets wherever there is a difference.

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

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

The adversary A^{asy} is given input $(1^n, pk, y)$ works as following:

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

2. Use $2(q_H + q_{dec} + 1)$-wise independent function, $2(q_G + q_{dec} + 1)$-wise independent function, $2(q_{H'} + q_{dec} + 1)$-wise independent polynomial to answer to the queries submitted to random oracles H, G and H' respectively.

3. Answer to the decryption queries as Dec^{**}.

4. Whenever A_{hy} outputs challenge messages (m_0, m_1), does as follow:

   - Selects $b \xleftarrow{\$} \{0,1\}$, $r \xleftarrow{\$} \mathtt{COIN}^{sy}$, $a^* \leftarrow \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \{0,1\}^{n_1}$.
   - Sets $c^* := Enc^{sy}_{a^*}(m_b; r)$ and $e^* := y$.
   - Sends (e^*, c^*, d^*) to the adversary A_{hy}.

5. Answer to the oracle queries as before and to the decryption queries by algorithm 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 query output $\perp$).

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

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

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *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.

[Ben81] Michael 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.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria 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.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De 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.

[ETTU15] Ehsan Ebrahimi Targhi, Gelo Tabia, and Dominique Unruh. Quantum collision-resistance of non-uniformly distributed functions, 2015. Available at `http://www.cs.ut.ee/~unruh/collision.pdf`.

[FO99] Eiichiro Fujisaki and Tatsuaki 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.

[FOPS04] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. *J. Cryptology*, 17(2):81–104, 2004.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 187–196. ACM, 2008.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sho01] Victor Shoup. OAEP reconsidered. In Joe 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.

[Unr14a] Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario 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.

[Unr14b] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth 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.

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc 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.

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran 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.

A OAEP-cryptosystem

The following definitions are similar to the definitions presented in [FOPS04], except we define them in the presence of a **quantum** adversary.

Definition 6 (Quantum partial-domain one-way function)**.** *We say 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,*

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

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

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

where $r \xleftarrow{\$} \{0,1\}^{k_0}$, and outputs ciphertext $(c,d) := \Big(f(s,t), H'(s\|t)\Big)$.

3. **Dec***: Given ciphertext (c,d), the decryption algorithm does as follows:*
   - *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 output $\perp$.*
     (b) *If c^* is not defined, then query the random oracle H' on a random input and return $\perp$.*
   - *When $c \in \operatorname{Im} f$, the decryption algorithm extracts $(s,t) = f^{-1}(c)$. If $H'(s\|t) \neq d$ it outputs $\perp$, otherwise it does as follows:*
     (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.

Theorem 8. *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 $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $r \xleftarrow{\$} \{0,1\}^{k_0}$, $(pk, sk) \leftarrow Gen(1^n)$, $\alpha^* \xleftarrow{\$} \{0,1\}^{k-k_0}$, $i \xleftarrow{\$} \{1, \ldots, q_G + q_{dec} + 1\}$, $\beta^* \xleftarrow{\$} \{0,1\}^{k_0}$
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^* is 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 $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* := f(s^*, t^*)$, $d^* := H'(s^* \| t^*)$
    let $b' \leftarrow A^{H,G,H',Dec}(c^*, d^*)$
  measure the argument $\tilde{r}$ of the i-th query to oracle G
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 $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* := f(s^*, t^*)$,
    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 $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* = f(s^*, t^*)$
      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 [Zha12], 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 $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b||0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* = f(s^*, t^*)$
    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 [Zha12]:

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

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 $b \xleftarrow{\$} \{0,1\}$, $s^* := m_b || 0^{k_1} \oplus \alpha^*$, $t^* := r \oplus H(s^*)$, $c^* = f(s^*, t^*)$
    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 $(\hat{s}, \hat{t}) := f^{-1}(c)$. The decryption algorithm does as follows:
   - 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:
   - 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:

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

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

□