A Note on Non-Committing Encryption in the Quantum Random Oracle Model

Anish Banerjee
Department of Computer Science and Engineering
IIT Delhi
Delhi, India
anish.cse.iitd@gmail.com

Shankh Gupta
Department of Computer Science and Engineering
IIT Delhi
Delhi, India
shankhgupta.iitdelhi@gmail.com

Venkata Koppula
Department of Computer Science and Engineering
IIT Delhi
Delhi, India
kvenkata@iitd.ac.in

Mahesh Sreekumar Rajasree*
CISPA Helmholtz
Saarbrücken, Germany
srmahesh1994@gmail.com

***Abstract*—Nielsen (CRYPTO 2002) demonstrated that constructing a non-committing encryption (NCE) scheme for an unbounded number of messages is impossible in both the standard model and the non-programmable random oracle model, underscoring fundamental barriers to achieving NCE within conventional cryptographic frameworks. However, he circumvented this impossibility in the *programmable* random oracle model by presenting a construction based on trapdoor functions, exploiting the oracle's programmability to enable ciphertext equivocation. In this work, we analyze Nielsen's construction in the quantum random oracle model (QROM), where quantum adversaries can query the oracle in superposition, and examine whether its security remains intact in this significantly stronger setting.**

***Index Terms*—non-committing, public key encryption, quantum random oracle**

I. Introduction

Non-committing encryption (NCE), introduced by Canetti, Feige, Goldreich, and Naor [1], is a powerful cryptographic primitive that enables a sender to encrypt a message without committing to its content. This distinctive property ensures that even if the encryption process is later scrutinized, the sender retains the ability to convincingly explain the ciphertext as encrypting any chosen message. Such flexibility makes NCE an essential building block in cryptographic protocols requiring adaptability, particularly in secure multiparty computation [1]–[3], where parties must interact securely despite potential future compromises. Recently, NCE has also found applications in incompressible cryptography [4]–[10], helping design encryption schemes that remain secure even when adversaries have bounded long-term storage.

A typical NCE scheme comprises standard setup, encryption, and decryption algorithms, alongside a simulator. The simulator can generate a fake public key and dummy ciphertext that are computationally indistinguishable from genuine ones. Later, upon receiving a specific message, the simulator reveals the randomness used to generate both the fake public key and the dummy ciphertext, demonstrating that the ciphertext corresponds to the given message. Furthermore, by revealing the randomness used to generate the public key, one can derive the corresponding secret key, allowing verification of the ciphertext's validity through decryption.

However, constructing secure non-committing encryption schemes is inherently challenging. A fundamental barrier was established by Nielsen [11], who proved that no secure NCE scheme can exist in the standard model or even in the non-programmable random oracle model (ROM) [12]. Nielsen showed that in these models, it is impossible to generate an unbounded number of non-committing ciphertexts while retaining the capability to later disclose the necessary randomness. This impossibility result posed a significant obstacle for constructing NCE schemes beyond heuristic approaches.

This barrier was circumvented in the programmable random oracle model, where the random oracle serves as an idealized cryptographic primitive that behaves like a truly random function with one crucial difference: it can be adaptively reprogrammed during protocol execution. Leveraging this programmability, Nielsen [11] introduced the first secure construction of a non-committing encryption scheme. Notably, this also marked the first clear separation between the programmable ROM and its non-programmable counterpart, highlighting the strength and utility of oracle programmability in cryptographic constructions.

With rapid advances in quantum computing and the growing importance of quantum cryptography, revisiting the security of Nielsen's construction in the quantum setting has become essential. Quantum adversaries are significantly more powerful than their classical counterparts, as they can query random oracles in superposition [13]. Such capabilities potentially

*Funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

invalidate classical security proofs, especially those relying on rewinding or adaptive programming techniques. Over the past decade, significant efforts have been made to understand cryptographic security against quantum adversaries, leading to the development of new insights and innovative proof techniques [14]–[18] to handle quantum oracle access.

In this paper, we address this open question by proving that Nielsen's non-committing encryption construction remains secure against quantum adversaries in the programmable quantum random oracle model.

II. Preliminaries

A. Notation

Let PPT and QPT denote probabilistic and quantum polynomial time. We denote the set of all positive integers up to n as $[n] := \{1, \ldots, n\}$. For any finite set S, $x \leftarrow S$ denotes a uniformly random element x from the set S.

B. Trapdoor Functions

A trapdoor function is a tuple of algorithms $(\mathsf{TDF.Setup}, \mathsf{TDF.Eval}, \mathsf{TDF.Inv})$ where:

- $(\mathsf{tdf.k}, \mathsf{tdf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda)$ is an efficient generation algorithm that takes as input the security parameter 1^λ and outputs a public key $\mathsf{tdf.k}$ and a trapdoor $\mathsf{tdf.td}$.
- $y \leftarrow \mathsf{TDF.Eval}(\mathsf{tdf.k}, x)$ is an efficient evaluation algorithm which takes as input a public key $\mathsf{tdf.k}$ and an element x from the domain $\mathcal{X}$[1]. It outputs an element from the range $\mathcal{Y}$.
- $x \leftarrow \mathsf{TDF.Inv}(\mathsf{tdf.td}, y)$ is an efficient inversion algorithm that takes as input a trapdoor $\mathsf{tdf.td}$ and an element $y \in \mathcal{Y}$ and returns an element $x \in \mathcal{X}$ or $\perp$

We require the following properties of correctness and security:

- **Correctness:** For all $\lambda \in \mathbb{N}, x \in \mathcal{X}$,

  $$\Pr\left[x = \mathsf{TDF.Inv}(\mathsf{tdf.td}, y)\right] = 1$$

  where $(\mathsf{tdf.k}, \mathsf{tdf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda)$, $y \leftarrow \mathsf{TDF.Eval}(\mathsf{tdf.k}, x)$ and the probability is over the randomness used in the setup and evaluation algorithm. Observe that this requires the function to be injective.
- **Security:** For any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

  $$\Pr\left[x \leftarrow \mathcal{A}(\mathsf{tdf.k}, y)\right] = \mathrm{negl}(\lambda)$$

  where $(\mathsf{tdf.k}, \mathsf{tdf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda)$, $x \leftarrow \mathcal{X}$ and $y \leftarrow \mathsf{TDF.Eval}(\mathsf{tdf.k}, x)$

In this work, we also require the notion of $n-$security, that is, given $n = \mathsf{poly}(\lambda)$ evaluations of the trapdoor function, no PPT algorithm can find the preimage of any one of the evaluations. It is easy to show that the above notion of trapdoor security implies $n-$ security.

Definition 1 ($n-$security)**.** *A trapdoor function is $n-$secure if for any PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all $\lambda \in \mathbb{N}$,*

$$\Pr\left[X^* \cap \left\{x^{(i)}\right\}_{i\in[n]} \neq \emptyset \;\middle|\; \begin{array}{l} (\mathsf{tdf.k}, \mathsf{tdf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda), \\ x^{(i)} \leftarrow \mathcal{X},\ y^{(i)} \leftarrow \mathsf{TDF.Eval}(\mathsf{tdf.k}, x^{(i)}), \\ X^* \leftarrow \mathcal{A}\left(\mathsf{tdf.k}, \left\{y^{(i)}\right\}_{i\in[n]}\right) \end{array} \right] = \mathrm{negl}(\lambda)$$

Theorem 2 ([19])**.** *Assuming the hardness of* LWE*, there exits post-quantum secure trapdoor functions.*

C. Non-Committing Encryption (NCE)

A non-committing encryption scheme consists of the following algorithms.

- $\mathsf{Setup}(1^\lambda; r_{\mathsf{Setup}})$: The setup algorithm takes as input the security parameter 1^λ . Using the random coins r_{Setup}, it outputs the public key pk and secret key sk.
- $\mathsf{Enc}(\mathsf{pk}, m; r_{\mathsf{Enc}})$: The encryption algorithm takes as input a master public key pk, a message m and using random coins r_{Enc} outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: The decryption algorithm takes as input a secret key sk and a ciphertext ct and outputs either a message m or $\perp$.
- $\mathsf{Sim}_1(1^\lambda)$: The first simulator takes as input the security parameter 1^λ and outputs a public key pk and a state state_1.
- $\mathsf{Sim}_2(\mathsf{state})$: The second simulator takes as input a state state, and outputs ciphertexts ct_i and an updated state state'.
- $\mathsf{Sim}_3\left(\mathsf{state}_2, \{m_i\}_{i\in[n]}\right)$: The second simulator takes as input a state state_2 and n messages and outputs $\left(r_{\mathsf{Setup}}, \{r_{\mathsf{Enc},i}\}_{i\in[n]}\right)$.

a) Correctness.: For correctness, we require that for all $\lambda \in \mathbb{N}$ and $(\mathsf{pk}, \mathsf{sk})$ output by $\mathsf{Setup}(1^\lambda)$, any message m,

$$\Pr_r[\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = m \mid \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, m; r))] = 1 - \mathrm{negl}(\lambda)$$

Non-Committing Security. Consider the following two experiments with an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and let $n = \mathsf{poly}(\lambda)$.

Real World::

- **Initialization Phase:** The challenger computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda; r_{\mathsf{Setup}})$ and sends pk to $\mathcal{A}_0$.
- **Challenge Phase:** Initialize $\mathcal{R} = \emptyset$. The adversary $\mathcal{A}_0$ can perform the following n many times where $n = \mathsf{poly}(\lambda)$ for any adversarially chosen polynomial $\mathsf{poly}(\cdot)$:
  1) It sends m_i^* to the challenger.
  2) The challenger computes $\mathsf{ct}_i^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i^*; r_{\mathsf{Enc},i})$ and sends it to $\mathcal{A}_0$. Further it adds $r_{\mathsf{Enc},i}$ to $\mathcal{R}$.

  At the end of this phase, $\mathcal{A}_0$ sends some auxiliary information $|\mathsf{aux}_0\rangle$ to the challenger.
- **Response Phase:** The challenger returns $(|\mathsf{aux}_0\rangle, (r_{\mathsf{Setup}}, \mathcal{R}))$ to $\mathcal{A}_1$. Finally, $\mathcal{A}_1$ outputs b.

[1] We assume the existence of an efficient sampling algorithm that can generate an (almost) uniformly random element from $\mathcal{X}$.

Simulated World::

- **Initialization Phase:** The challenger computes $(\mathsf{pk}, \mathsf{state}) \leftarrow \mathsf{Sim}_1(1^\lambda)$ and sends pk to $\mathcal{A}_0$.
- **Challenge Phase:** Initialize $\mathcal{M} = \emptyset$ and $\mathsf{state}_0 = \mathsf{state}$. For $i = 1$ to n:
  1) $\mathcal{A}_0$ sends m_i^* to the challenger. The challenger adds it to $\mathcal{M}$.
  2) The challenger computes $(\mathsf{ct}_i^*, \mathsf{state}_i) \leftarrow \mathsf{Sim}_2(\mathsf{state}_{i-1})$ and sends it to $\mathcal{A}_0$.

  At the end of this phase, $\mathcal{A}_0$ send some auxiliary information $|\mathsf{aux}_0\rangle$ to the challenger.
- **Response Phase:** The challenger computes $(r_{\mathsf{Setup}}, \mathcal{R}) \leftarrow \mathsf{Sim}_3(\mathsf{state}_n, \mathcal{M})$, where $\mathcal{R} = \{r_{\mathsf{Enc},i}\}_{i\in[n]}$ and returns $(|\mathsf{aux}_0\rangle, (r_{\mathsf{Setup}}, \mathcal{R}))$ to $\mathcal{A}_1$. Finally, $\mathcal{A}_1$ outputs b.

Let p_{Real} and p_{Sim} be the probabilities with which $\mathcal{A}$ outputs 0 in the real world and simulated world, respectively.

Definition 3. *An NCE scheme is said to be secure against unbounded non-committing encryptions if for all* PPT *adversaries $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that, for all $\lambda \in \mathbb{N}, n = \mathsf{poly}(\lambda)$,*

$$|p_{\mathrm{Real}} - p_{\mathrm{Sim}}| = \mathrm{negl}(\lambda)$$

Theorem 4 ([11]). *There exists a secure NCE against unbounded non-committing encryptions in the programmable random oracle model.*

In the main section, we will show that the scheme presented in [11] is also quantum secure, provided there exists post-quantum secure trapdoor pseudorandom functions. Hence, we will present the construction in [11] below. Let $\mathsf{TDF} = (\mathsf{TDF.Setup}, \mathsf{TDF.Eval}, \mathsf{TDF.Inv})$ be a secure trapdoor function with domain $\mathcal{X}$ and range $\mathcal{Y}$ and $H : \mathcal{X} \to \{0,1\}$ be a random function modeled as a random oracle.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm takes as input the security parameter 1^λ. It generates an evaluation key and trapdoor $(\mathsf{prf.k}, \mathsf{prf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda; r_{\mathsf{Setup}})$ and returns the public $\mathsf{pk} := \mathsf{prf.k}$ and secret key $\mathsf{sk} := \mathsf{prf.td}$.
- $\mathsf{Enc}(\mathsf{pk}, m)$: The encryption algorithm takes as input a public key $\mathsf{pk} = \mathsf{prf.k}$ and a message m. It randomly samples x and computes $c_0 = \mathsf{TDF.Eval}(\mathsf{pk}, x)$. It generates $c_1 = H(x) \oplus m$ and returns $\mathsf{ct} = (c_0, c_1)$. The randomness used in the algorithm r_{Enc} is the randomness used to generate x.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: The decryption algorithm takes as input a secret key $\mathsf{sk} = \mathsf{prf.td}$ and a ciphertext $\mathsf{ct} = (c_0, c_1)$. It computes $x \leftarrow \mathsf{TDF.Inv}(\mathsf{sk}, c_0)$ and $m = H(x) \oplus c_1$. It returns m.
- $\mathsf{Sim}_1(1^\lambda)$: The first simulator takes as input the security parameter 1^λ. It generates an evaluation key and trapdoor $(\mathsf{prf.k}, \mathsf{prf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda; r_{\mathsf{Setup}})$ and returns the public $\mathsf{pk} := \mathsf{prf.k}$ and state $\mathsf{state}_1 := ((\mathsf{prf.k}, \mathsf{prf.td}, r_{\mathsf{Setup}}), \emptyset)$.
- $\mathsf{Sim}_2(\mathsf{state}_{i-1} = ((\mathsf{prf.k}, \mathsf{prf.td}, r_{\mathsf{Setup}}), \mathcal{S}_{i-1}))$: To generate a fake ciphertext, it randomly samples x_i and computes $c_{0,i} = \mathsf{TDF.Eval}(\mathsf{pk}, x_i)$. It samples $c_{1,i}$ uniformly at random and sets $\mathsf{ct}_i = (c_{0,i}, c_{1,i})$. Let $s_i = (r_{\mathsf{Enc},i}, c_{1,i})$ where $r_{\mathsf{Enc},i}$ is the random coins used to generate x_i. It updates $\mathsf{state}_i = ((\mathsf{prf.k}, \mathsf{prf.td}, r_{\mathsf{Setup}}), \mathcal{S}_{i-1} \cup s_i)$. Finally, it returns $(\mathsf{ct}_i, \mathsf{state}_i)$.
- $\mathsf{Sim}_3\left(\mathsf{state}_2 = ((\mathsf{prf.k}, \mathsf{prf.td}, r_{\mathsf{Setup}}), \mathcal{S}_n), \{m_i\}_{i\in[n]}\right)$: The third simulator uses $\{r_{\mathsf{Enc},i}\}_{i\in[n]}$ obtained from $\mathcal{S}_n$ to compute $\{x_i\}_{i\in[n]}$. It then reprograms the random oracle as follows – $H(x_i) := c_{1,i} \oplus m_i$. Finally, it returns $\left(r_{\mathsf{Setup}}, \{r_{\mathsf{Enc},i}\}_{i\in[n]}\right)$ where r_{Setup} is obtained from state_1 which is a part of state_2.

Theorem 5 ([11]). *There does not exists an NCE secure* under blackbox reductions *against unbounded non-committing encryptions in the standard model and unprogrammable random oracle model.*

D. Quantum Random Oracle Model

In the classical random oracle model, the adversary is given access to a uniformly random hash function O (of a certain domain and range) and it can only learn a value $O(x)$ by querying the oracle on the classical string x. In the real world, the oracle is replaced by a (sufficiently complicated) hash function, allowing an attacker with a quantum computer to evaluate it on quantum states. It thus becomes essential to consider the quantum (accessible) random oracle, which is essentially a unitary U_O implementing

$$|x\rangle |y\rangle \xrightarrow{U_O} |x\rangle |y \oplus O(x)\rangle$$

The quantum adversary is allowed to make superposition queries $|\psi\rangle = \sum_{x,y} \alpha_{x,y} |x\rangle |y\rangle$ to U_O.

Zhandry [16] showed that a random $2q$-degree function can be used to simulate a random oracle for a quantum oracle algorithm that makes at most q queries.

Theorem 6 ([16]). *For any (unbounded) quantum oracle algorithm $\mathcal{A}$ that make at most q many oracle queries,*

$$\Pr_H[\mathcal{A}^H() = 1] = \Pr_f[\mathcal{A}^f() = 1]$$

where H is a randomly chosen function over the entire domain of functions with domain $\mathcal{X}$ and range $\mathcal{Y}$ whereas f is a randomly chosen function with degree $2q$ with domain $\mathcal{X}$ and range $\mathcal{Y}$.

E. One-Way to Hiding

The One-Way to Hiding (O2H) theorem is a important tool for proving security in the quantum random oracle model. It bounds an adversary's advantage in distinguishing between two experiments – one where oracle evaluations are performed on a randomly chosen set and another where the oracle outputs are replaced with independent randomness. We will be using the following variant in our proof.

Theorem 7 (One-way to hiding (O2H), reprogramming [14], [20]). *Let $H : X \to Y$ be random function. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be a (stateful) quantum oracle algorithm such that the output of $\mathcal{A}_0$ is a (quantum) state which is given as input to $\mathcal{A}_1$.*

Let q be the total number of oracle queries made by $\mathcal{A}$. Let $\mathcal{B}^H$ be an oracle algorithm that on input (i, S) does the following: run $\mathcal{A}_0^H[S, U]$[2] *until the i^{th} query, measure all query input registers in the computational basis, output the set T of measurement outcomes. Let*

$$P_{\text{Real}} := \Pr\left[b = 1 \middle| \begin{array}{c} S \subseteq_R X, A_0^H[S, \{H(x)\}_{x\in S}], \\ b \leftarrow A_1^H[S, \{H(x)\}_{x\in S}] \end{array} \right]$$

$$P_{\text{Sim}} := \Pr\left[b = 1 \middle| \begin{array}{c} S \subseteq_R X, V \subseteq_R Y, |S| = |V|, \\ A_0^H[S, V], H(S) := V, b \leftarrow A_1^H[S, V] \end{array} \right]$$

$$P_{\text{Guess}} := \Pr\big[S \cap T \neq \emptyset | S \subseteq_R X, i \leftarrow [d], T \leftarrow B^H[i, S]\big]$$

where $H(S) := V$ denotes H being reprogramming at S with V. Then,

$$|P_{\text{Real}} - P_{\text{Sim}}| \leq 2q\sqrt{P_{\text{Guess}}}$$

III. NCE in Programmable QROM

In this section, we prove that Nielsen's construction is secure in the programmable quantum random oracle model.

Theorem 8. *The NCE construction in Theorem 4 is secure against unbounded non-committing encryptions in the programmable **quantum** random oracle model.*

Proof overview: Recall that a properly generated ciphertext for a message m is obtained by randomly sampling x, computing $\mathsf{TDF.Eval}(\mathsf{pk}, x)$ as the first part of the ciphertext, and setting the second part as $H(x) \oplus m$. Similarly, to generate a dummy ciphertext, the NCE simulator first samples a random x and computes $\mathsf{TDF.Eval}(\mathsf{pk}, x)$, using this as the first component of the ciphertext. However, the second component is a truly random value y. Later, to open this ciphertext to a message m, the simulator reprograms the random oracle at x to $y \oplus m$. Our objective is to prove that no efficient quantum adversary can distinguish whether the quantum random oracle has been programmed or remains unmodified. To achieve this, we apply the O2H theorem (see Theorem 7) to reduce the distinguishing problem to the security of the trapdoor function.

Proof. We show that if an adversary $\mathcal{A}$ can distinguish between the two games with non-negligible probability, then we can construct a reduction B which breaks the security of the trapdoor function.

Claim 9. *If $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ can distinguish between the real and the simulated world with non-negligible advantage, then there exists a reduction $\mathcal{B}$ which breaks the security of the trapdoor function with non-negligible probability.*

Proof. Let the advantage of $\mathcal{A}$ be ϵ. Consider the following reduction:

1) The reduction algorithm $\mathcal{B}$ acts as an adversary for the trapdoor function challenger $\mathcal{C}$ and uses $\mathcal{A}$ as a subroutine.
2) $\mathcal{C}$ computes $(\mathsf{tdf.k}, \mathsf{tdf.td}) \leftarrow \mathsf{TDF.Setup}(1^\lambda)$ and sets $\mathsf{pk} = \mathsf{tdf.k}$. It sends pk to the reduction $\mathcal{B}$. Since $\mathcal{A}$ is a QPT algorithm, there are upper bounds, denoted as n and q, on the number of messages $\mathcal{A}_0$ sends in the challenge phase and the number of queries made by $\mathcal{A}$.
3) The challenger also generates evaluations on n random values and sets $c^{(i)} \leftarrow \mathsf{TDF.Eval}(\mathsf{pk}, x^{(i)})$ for randomly sampled $x^{(i)}$, $\forall i \in [n]$ and sends the set $\left\{c^{(i)}\right\}_{i\in[n]}$ to $\mathcal{B}$.
4) $\mathcal{B}$ randomly samples $i \leftarrow [q]$ and a $2q$-degree polynomial $p(\cdot)$ over $\mathcal{X}$. It sets the random oracle $H := p$ and runs $\mathcal{A}_0^H$ on input pk and for any $m^{(i)}$ it receives from $\mathcal{A}_0^H$, $\mathcal{B}$ sends $\left\{\mathsf{ct}^{*(i)}\right\}_{i\in[n]}$ where $\mathsf{ct}^{*(i)} := \left(c^{(i)}, y^{(i)}\right)$, where $y^{(i)}$ is sampled randomly from $\{0,1\}^\ell$.
5) Just before the i^{th} query, $\mathcal{B}$ stops execution of $\mathcal{A}_0$ and performs a measurement on all query input registers of $\mathcal{A}_0$ to obtain T. It then returns T to $\mathcal{C}$.

Using Theorem 6, the simulation of the random oracle using a random $2q$-degree function is perfectly indistinguishable to $\mathcal{A}$. Using Theorem 7, we obtain that

$$|P_{\text{Real}} - P_{\text{Sim}}| \leq 2q\sqrt{P_{\text{Guess}}}$$

where P_{Real} and P_{Sim} are the probabilities that the adversary $\mathcal{A}$ outputs $b = 0$ in the real and simulated worlds respectively. Observe that P_{Guess} is the probability that $\mathcal{B}$ returns T such that it contains at-least one of the challenge preimages $x^{(i)}$. Thus the probability that $\mathcal{B}$ wins in the above reduction against the trapdoor function challenger is:

$$P_{\text{Guess}} \geq \left(\frac{\epsilon}{2q}\right)^2$$

which is non-negligible since q is polynomial in the security parameter. □

Since, TDF is a secure quantum trapdoor function, we have $P_{\text{Guess}} = \text{negl}(\lambda)$. This implies that

$$|P_{\text{Real}} - P_{\text{Sim}}| \leq 2q \cdot \text{negl}(\lambda) = \text{negl}'(\lambda)$$

where $\text{negl}'(\lambda)$ denotes a negligible function obtained by multiplying the polynomial factor $2q$ by the negligible function $\text{negl}(\lambda)$. □

Using Theorem 2 and Theorem 8, we get the following corollary.

Corollary 10. *Assuming the hardness of* LWE*, there exists non-committing encryption scheme that is secure against unbounded non-committing enryptions in the programmable quantum random oracle model.*

References

[1] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 639–648. 1

[2] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai, "Universally composable two-party and multi-party secure computation," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 494–503. 1

[3] J. A. Garay, D. Wichs, and H.-S. Zhou, "Somewhat non-committing encryption and efficient adaptively secure oblivious transfer," in *Annual International Cryptology Conference*. Springer, 2009, pp. 505–523. 1

[2] The notation $\mathcal{A}^H[S, U]$ indicates that during the experiment in which $\mathcal{A}$ interacts, if the experiment requires $H(x_i)$ for any $x_i \in S$, it will use u_i where $S = [x_1, \ldots, x_\ell]$ and $U = [u_1, \ldots, u_\ell]$ are ordered set.

[4] J. Guan, D. Wichs, and M. Zhandry, "Incompressible Cryptography," in *Advances in Cryptology – EUROCRYPT 2022*, ser. Lecture Notes in Computer Science, O. Dunkelman and S. Dziembowski, Eds. Cham: Springer International Publishing, 2022, pp. 700–730. 1

[5] P. Branco, N. Döttling, and J. Dujmović, "Rate-1 incompressible encryption from standard assumptions," in *Theory of Cryptography Conference*. Springer, 2022, pp. 33–69. 1

[6] J. Guan, D. Wichs, and M. Zhandry, "Multi-instance randomness extraction and security against bounded-storage mass surveillance," in *Theory of Cryptography Conference*. Springer, 2023, pp. 93–122. 1

[7] K. Bhushan, R. Goyal, V. Koppula, V. Narayanan, M. Prabhakaran, and M. S. Rajasree, "Leakage-resilient incompressible cryptography: Constructions and barriers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2025, pp. 201–234. 1

[8] R. Goyal, V. Koppula, M. S. Rajasree, and A. Verma, "Incompressible Functional Encryption," in *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), R. Meka, Ed., vol. 325. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, pp. 56:1–56:22. 1

[9] R. Goyal, F. Kitagawa, V. Koppula, R. Nishimaki, M. S. Rajasree, and T. Yamakawa, "Non-committing identity based encryption: Constructions and applications," in *Public-Key Cryptography – PKC 2025*. Springer International Publishing, 2025. 1

[10] N. Döttling, A. Koch, S. Maier, J. Mechler, A. Müller, J. Müller-Quade, and M. Tieplet, "The quantum decoherence model: Everlasting composable secure computation and more," *Cryptology ePrint Archive*, 2025. 1

[11] J. B. Nielsen, "Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case," in *Annual International Cryptology Conference*. Springer, 2002, pp. 111–126. 1, 3

[12] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 62–73. 1

[13] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *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 17*. Springer, 2011, pp. 41–69. 1

[14] D. Unruh, "Quantum position verification in the random oracle model," in *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II 34*. Springer, 2014, pp. 1–18. 2, 3

[15] ——, "Revocable quantum timed-release encryption," *Journal of the ACM (JACM)*, vol. 62, no. 6, pp. 1–76, 2015. 2

[16] M. Zhandry, "Secure identity-based encryption in the quantum random oracle model," in *Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 758–775. 2, 3

[17] ——, "How to record quantum queries, and applications to quantum indifferentiability," in *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer, 2019, pp. 239–268. 2

[18] ——, "How to construct quantum random functions," *Journal of the ACM (JACM)*, vol. 68, no. 5, pp. 1–43, 2021. 2

[19] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 2008, pp. 197–206. 2

[20] A. Ambainis, M. Hamburg, and D. Unruh, "Quantum security proofs using semi-classical oracles," in *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer, 2019, pp. 269–295. 3