On quantum indistinguishability under chosen plaintext attack

Tore Vincent Carstens[1], Ehsan Ebrahimi[2], Gelo Tabia[3], and Dominique Unruh[4]

[1] University of Tartu, Estonia
[2]SnT, University of Luxembourg
[3]Center for Quantum Technology, National Tsing Hua University, Hsinchu 300, Taiwan
[4]University of Tartu, Estonia

May 20, 2020

Abstract. An encryption scheme is called indistinguishable under chosen plaintext attack (short IND-CPA), if an attacker cannot distinguish the encryptions of two messages of his choice. Alternatively there are other variants of this definition, that all turn out to be equivalent in the classical case. However in the quantum case, there is a lack of a comprehensive study of all quantum versions of IND-CPA security notion. We give an overview of these different variants of quantum IND-CPA for symmetric encryption schemes. In total, 57 different notions are valid and achievable. We investigate the relations between these notions and prove various equivalences, implications, non-equivalences, and non-implications between these variants. Some of non-implications are left as conjectures and need further research.

Keywords. Symmetric encryption, Quantum security, IND-CPA.

Contents

1 Introduction

Advances in quantum computing have continuously raised the interest in post-quantum secure cryptography. In order for a post-quantum secure scheme to be designed, as a first step a security definition has to be agreed on. There have been extensive research works toward proposing quantum counterparts of classical security definitions for different cryptographic primitives: encryption schemes [BZ13b, GHS16, CEV20], message authentication codes [BZ13a, AMRS18], hash functions [Zha15, Unr16], etc. For a classical cryptographic primitive to be quantum secure, besides the necessity of a quantum hardness assumption, we also need to consider how a quantum adversary will interact with a classical algorithm. In the research works mentioned above, the security notions have been defined in a setting where the quantum adversary is allowed to make superposition queries to such cyptographic primitives. In this paper, we focus on defining quantum security definitions for symmetric encryption schemes. Our aim is to define and relate possible quantum versions of indistinguishability under chosen plaintext attack.

Indistinguishability under chosen plaintext attack (IND-CPA) is a classical security notion for encryption schemes in which the adversary interacts with the encryption oracle in two phases: the learning phase and challenge phase. The learning phase (if it exists) is defined in a unique way: the adversary makes queries to the encryption oracle. In contrast, the challenge phase can be defined in different ways:
(a) The adversary chooses two messages m_0, m_1 and sends them to the challenger. The adversary will receive back the encryption of m_b for a random bit b.
(b) The adversary chooses two messages m_0, m_1 and sends them to the challenger. The adversary will receive back the encryption of m_b, $m_{\bar{b}}$ for a random bit b.
(c) The adversary chooses a messages m and sends it to the challenger. The challenger will send back either the encryption of m or a randomly chosen message depended on a random bit b.
At the end, the adversary tries to guess the bit b. In other words, the definition varies according to how the challenger responds to the adversary during the challenge phase. We call it the "return type". As summarized above, there are three different return types: a) the challenger returns one ciphertext. b) the challenger returns two ciphertexts. c) the challenger returns a real or random ciphertext. A comprehensive study of these notions has been done in [BDJR97].

There are some scattered attempts to translate IND-CPA notions above to the quantum case [BZ13b, GHS16, MS16] (see Section 1.1 for more details), however, it is not a complete list and we lack a study of how the existing security definitions relate to each other. In our paper, we present all possible quantum versions of IND-CPA. We compare them to have a comprehensive study as the classical case.

1.1 Previous work

Boneh-Zhandry definition. In ([BZ13b]), Boneh and Zhandry initiate developing a quantum security version of IND-CPA. They consider that the adversary has "standard oracle access" to the encryption oracle in the learning phase. The standard oracle access to the encryption oracle Enc is defined as the unitary operator $U_{\mathrm{Enc}} : |x, y\rangle \to |x, y \oplus \mathrm{Enc}(x)\rangle$ (see section 3). For the challenge phase, they attempt to translate the classical notion of one-ciphertext and two-ciphertexts return type (presented in item a and item b) to the quantum case using standard query model. However, they show that the natural translation cannot be achieved. So instead they consider classical challenge queries in their proposed definition. This inconsistency between the learning phase and the challenge phase resulted in further investigation of the quantum IND-CPA notion in [GHS16].

Quantum IND-CPA notions in [GHS16]. In [GHS16], the authors attempt to resolve the inconsistency of the learning and the challenge phase of the security definition proposed in [BZ13b]. They propose a "security tree" of possible security notions. In a nutshell, their security tree is built on four different perspectives on the interaction between the adversary and the challenger: 1) how the challenger is implemented: the oracle model or the challenger model; 2) how the adversary sends the challenge queries: the adversary sends quantum messages during the challenge phase or it sends classical description of quantum messages; 3) whether the challenger sends back the input registers to the adversary or keeps them; and 4) the query model: the adversary has standard oracle access to the challenger or it has "minimal oracle" access [KKVB02] (that is defined as $|x\rangle \to |\mathrm{Enc}(x)\rangle$). Even though in total there are $2^4 = 16$ possible security definitions, only two are meaningful, achievable, and novel. These two definitions are (according to their terminology briefed above): 1) the challenger model, quantum messages, not returning the input register and minimal oracle access [1]. 2) the challenger model, classical messages, not returning the input register and minimal oracle access. In our paper, we do not consider the case when the adversary can submit the classical description of quantum messages. Therefore, we only study the former security notion in our paper. However, we do not differentiate between the challenger model and the oracle model[2]. Instead, we consider a black-box access to the challenger in which this black-box access can be either standard oracle access, or minimal oracle access, or etc. In this paper, we refer to the minimal query model as the "erasing query model" (see section 3).

Quantum IND-CPA notion in [MS16]. In [MS16], Mossayebi and Schack focus on translating the real-or-random case (item c) to the quantum setting by considering an adversary that has standard oracle access to the encryption oracle. Their security definition consists of two experiments, called real and permutation. In the real experiment, the adversary's queries will be answered by the encryption oracle without any modification (access to U_{Enc}) whereas in the permutation game, in each query a random permutation will be applied to the adversary's message and the permuted message will be encrypted and returned to the adversary (access to $U_{\mathrm{Enc}\circ\pi}$ for a random π). The advantage of the adversary in distinguishing these two experiments should be negligible for a secure encryption scheme. This is a security notion without the learning phase and many challenge queries when the adversary has the standard oracle access to the challenger and the challenge phase is implemented by the real-or-random return type.

Therefore, in total there are 3 achievable proposals for quantum IND-CPA notion in the literature so far. In this paper, we study and relate 57 achievable proposals for quantum IND-CPA notion (including security definitions briefed above). (See section 6.)

1.2 Our contribution

In this paper, we define all possible quantum IND-CPA security notions. In order to have a comprehensive list of security definitions, we classify them according to several criteria:

(1) Number of queries that the adversary can make during the learning and challenge phase: zero, one or many queries. (Note that in the learning phase either there is no query or many queries, while in the challenge phase either there is one query or many queries.)

(2) Query model in which the adversary is interacting with the challenger: classical, standard, erasing, or "embedding query model" where the embedding query model is the same as the standard oracle model except that the adversary only provides the input register and the output register will be initiated with $|0\rangle$ by the challenger (see section 3).

(3) The return type of the challenge ciphertext: one-ciphertext (similar to item a), two-ciphertexts (similar to item b) and real-or-random (similar to item c))

[1] This security definition is equivalent to the indistinguishability notion proposed in [BJ15] for secret key encryption of quantum messages when restricted to a classical encryption function operating in the minimal query type.

[2] Note that in order to implement the minimal oracle, a decryption query is needed and it has to be done by the challenger in case of symmetric key encryption schemes. So in [GHS16] the authors introduce the challenger model in which the challenger implement the minimal query model using its secret key. Later in [GKS20], the authors extend the security notion in [GHS16] for public-key encryption schemes and they show that the adversary can implement the minimal oracle itself (using the randomness and without using the decryption) for some public-key encryption schemes.

There are 5 choices for the learning phase and 24 choices for the challenge phase. Therefore, all the combinations are 120 cases.

Not valid security notions. Note that we do not consider a security notion with different quantum query models in the learning phase and the challenge phase to be a valid notion. (However, the classical access can be combined with any of quantum query models!) Also, we do not consider a security notion with no learning queries and only one challenge query since this corresponds to IND-OT-CPA notion that will not be considered in this paper.

Impossible security notions. Any security notion with the standard query model and the return type of one-ciphertext or two-ciphertexts in the challenge phase is impossible to be achieved. Any query model with the embedding query type and the one-ciphertext return type in the challenge phase is impossible to be achieved (see section 5).

This leaves us with 57 notions that remain valid. Then, we compare these notions and put the equivalent notions in the same panel and this results in 14 panels Figure 1. We give an overview of the equivalent notions in each panel and relation between panels below.

Security notions that are equivalent (see section 6):
Panel 1. We show that all valid security notions with the erasing query model in the challenge phase are equivalent excluding when the return type is real-or-random (security notions in Panels 4 and 9, see below) and when the learning queries are classical and there is one challenge query of either the return type of one-ciphertext or two-ciphertexts (security notions in Panels 3 and 8, see below). This panel consists of 8 security notions. (Note that this panel includes the security notion in [GHS16].) We can conclude these equivalences by Theorem 1, Theorem 3 and Theorem 6.
Panel 2. We show that all valid security notions with the standard query model and the real-or-random return type in the challenge phase are equivalent excluding the security notion with the classical learning queries and one challenge query (security notion in Panel 12, see below). In other words, all valid security notions that have many challenge queries of the standard query model and real-or-random return type are equivalent. This panel consists of 4 security notions. (Note that this panel includes the security notion in [MS16].) We can conclude these equivalences by Theorem 1 and Theorem 4.
Panel 3. This panel contains only one security notion: classical learning queries, one challenge query of erasing model and two-ciphertexts return type.
Panel 4. We show that all valid security notions with the erasing query model and the real-or-random return type in the challenge phase are equivalent excluding the security notion with the classical learning queries and one challenge query (security notion in Panel 9, see below). This panel consists of 4 security notions. We can conclude these equivalences by Theorem 1 and Theorem 4.
Panel 5. We show that all valid security notions with the embedding query model in the challenge phase are equivalent excluding when the return type is one-ciphertext (security notions in Panel 11, see below) and when the learning queries are classical and there is one challenge query of either two-ciphertexts or real-or-random return type (security notions in Panels 7 and 13, see below). This panel consists of 5 security notions. We can conclude these equivalences by Theorem 1, Theorem 4, Theorem 3 and Theorem 8.
Panel 6. We show that all valid security notions with the standard query model in the learning phase and the classical access in the challenge phase are equivalent. This panel consists of 6 security notions. (Note that this panel includes the security notion in [BZ13b].) We can conclude these equivalences by Theorem 1 and Theorem 2.
Panel 7. This panel consists of the security notion with the classical learning queries and one challenge query of type embedding model with two-ciphertexts return type.
Panel 8. This panel consists of the security notion with the classical learning queries and one challenge query of the erasing model with one-ciphertext return type.
Panel 9. This panel consists of the security notion with the classical learning queries and one challenge query of type erasing with real-or-random return type.
Panel 10. We show that all security notions with the erasing learning queries and the classical access in the challenge phase are equivalent. This panel consists of 6 security notions. We can conclude these equivalences by Theorem 1 and Theorem 2.
Panel 11. We show that all security notions with the embedding learning queries and the classical access in the challenge phase are equivalent. This panel consists of 6 security notions. We can conclude

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
P1		$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow^{7}$	$\Rightarrow$	$\Rightarrow^{9}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow^{17}$	$\Rightarrow$	$\Rightarrow$
P2	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow^{3}$	$\overset{?}{\not\Rightarrow}$	$\not\Rightarrow^{14}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P3	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\overset{?}{\not\Rightarrow}$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P4	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\Rightarrow$	$\overset{?}{\not\Rightarrow}$	$\Rightarrow$	$\not\Rightarrow^{3}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P5	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P6	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow^{15}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{16}$	$\Rightarrow$
P7	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow^{8}$	$\Rightarrow$
P8	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\overset{?}{\not\Rightarrow}$		$\Rightarrow^{7}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P9	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P10	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{15}$	$\not\Rightarrow$	$\not\Rightarrow$		$\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{13}$	$\Rightarrow$
P11	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$
P12	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\overset{?}{\not\Rightarrow}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\overset{?}{\not\Rightarrow}$		$\Rightarrow$	$\Rightarrow$
P13	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\Rightarrow$
P14	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{12}$	$\not\Rightarrow$	$\not\Rightarrow$	

Table 1: Implications and separations between panels. Red non-implications are conjectures. Blue non-implications have been proven assuming that conjectures are correct. See section 7 for more details.

these equivalences by Theorem 1 and Theorem 2.

Panel 12. This panel consists of the security notion with the classical learning queries and one challenge query of the standard query model with real-or-random return type.

Panel 13. This panel consists of the security notion with the classical learning queries and one challenge query of the embedding query model with real-or-random return type.

Panel 14. All valid security notions with classical learning and challenge queries are equivalent. This panel consists of 9 security notions. We can conclude these equivalences by Theorem 1 and Theorem 2.

Implications and non implications (see section 6 and section 7). The implications and separation have been drawn in Table 1. We read the table from the left to the right and an arrow in the position PnPm indicates if Pn implies Pm or Pn does not imply Pm. The red arrows with a question mark on top ($\overset{?}{\not\Rightarrow}$) were left as conjectures. The blue arrows show that proof of non-implication relies on a conjecture. Therefore, if the corresponding conjecture necessary to prove a blue non-implication is rejected, the non-implication becomes an open question.

Conclusions of Table 1. Two panels P1 and P2 together imply all other security notions. However, they are not comparable to each other. This resolves an open question stated in [MS16, GKS20] for a comparison between these security notions. As opposite to the classical case that different IND-CPA notions with one-ciphertext, two-ciphertexts and real-or-random return type are equivalent, when the challenge query is quantum (standard, embedding or erasing) this is not the case. More specifically, 1) for the standard query model, only the real-or-random return type is achievable (and two others are impossible to be achieved). 2) for the embedding query model, the one-ciphertext return type is impossible to be achieved, however, two other cases are equivalent (see Panel 5 above). 3) for the erasing query model, the one-ciphertext and two-ciphertexs return type are equivalent (see Panel 1) and they are stronger than the real-or-random return type (Panel 1 implies Panel 4 but Panel 4 does not imply Panel 1.)

Decoherence Lemmas: We show that for a random sparse injective function if we measure the input register of a quantum query of erasing in the computational basis, this will not be noticed by the adversary. We prove similar result for the embedding query model and a random function (see section 4). These lemmas help to prove the security of some encryption schemes since the measurement applied to the input register effectively make the query classical.

A secure encryption scheme in all notions. Finally, we present an encryption scheme that is secure in all security notions (see section 8). From Table 1, we can see that Panel 1 and Panel 2 together imply all other panels. So we present an encryption scheme that is secure respected to security notions in Panel 1 and 2.

1.3 Organization of the paper

In section 2, we give some notations and preliminaries. The section 3 is dedicated to definitions that are needed in the paper. We present all possible security notions for IND-CPA in the quantum case in this section. In section 4, we prove some lemmas that are needed for security proofs. The section 5 is dedicated to rule out security notions that are impossible to be achieved for any encryption scheme. In section 6, we investigate implications between all security notions defined in section 3. We obtain 14 groups of equivalent security notions. Then, we prove all implications between these 14 panels. The section 7 is dedicated to verify the remaining relations that are non-implications between panels. Even though we show many non-implications, we leave some of them unresolved and as conjectures (see Conjecture 1.). Finally, we present an encryption scheme that is secure with respect to all security notions defined in the paper in section 8.

2 Preliminaries

We recall some basic of quantum information and computation needed for our paper below. Interested reader can refer to [NC16] for more informations. For two vectors $|\Psi\rangle = (\psi_1, \psi_2, \cdots, \psi_n)$ and $|\Phi\rangle = (\phi_1, \phi_2, \cdots, \phi_n)$ in $\mathbb{C}^n$, the inner product is defined as $\langle \Psi, \Phi\rangle = \sum_i \psi_i^* \phi_i$ where ψ_i^* is the complex conjugate of ψ_i. Norm of $|\Phi\rangle$ is defined as $\||\Phi\rangle\| = \sqrt{\langle \Phi, \Phi\rangle}$. The outer product is defined as $|\Psi\rangle\langle\Phi| : |\alpha\rangle \to \langle \Phi, \alpha\rangle|\Psi\rangle$. The n-dimensional Hilbert space $\mathcal{H}$ is the complex vector space $\mathbb{C}^n$ with the inner product defined above. A quantum system is a Hilbert space $\mathcal{H}$ and a quantum state $|\psi\rangle$ is a vector $|\psi\rangle$ in $\mathcal{H}$ with norm 1. An unitary operation over $\mathcal{H}$ is a transformation $\mathbf{U}$ such that $\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbb{I}$ where $\mathbf{U}^\dagger$ is the Hermitian transpose of $\mathbf{U}$ and $\mathbb{I}$ is the identity operator over $\mathcal{H}$. The computational basis for $\mathcal{H}$ consists of $\log n$ vectors $|b_i\rangle$ of length $\log n$ with 1 in the position i and 0 elsewhere. With this basis, the unitary CNOT is defined as CNOT: $|m_1, m_2\rangle \to |m_1, m_2 \oplus m_2\rangle$ where m_1, m_2 are bit strings. The Hadamard unitary is defined as H: $|b\rangle \to \frac{1}{\sqrt{2}}(|\bar{b}\rangle + (-1)^b|b\rangle)$ where $b \in \{0, 1\}$. An orthogonal projection $\mathbf{P}$ over $\mathcal{H}$ is a linear transformation such that $\mathbf{P}^2 = \mathbf{P} = \mathbf{P}^\dagger$. A measurement on a Hilbert space is defined with a family of orthogonal projectors that are pairwise orthogonal. An example of measurement is the computational basis measurement in which any projection is defined by a basis vector. The output of computational measurement on state $|\Psi\rangle$ is i with probability $\|\langle b_i, \Phi\rangle\|^2$ and the post measurement state is $|b_i\rangle$. The density operator is of the from $\rho = \sum_i p_i|\phi_i\rangle\langle\phi_i|$ where p_i are non-negative and add up to 1. This represent that the system will be in the state $|\phi_i\rangle$ with probability p_i. We denote the trace norm with $||\cdot||_1$, i.e. $||M||_1 = \mathrm{tr}(|M|) = \mathrm{tr}(\sqrt{M^\dagger \cdot M})$. For two density operators ρ_1 and ρ_2, the trace distance is defined as $\mathrm{TD}(\rho_1, \rho_2) = \frac{1}{2}||\rho_1 - \rho_2||_1$. For two quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the composition of them is defined by the tensor product and it is $\mathcal{H}_1 \otimes \mathcal{H}_2$. For two unitary U_1 and U_2 defined over $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively, $(U_1 \otimes U_2)(\mathcal{H}_1 \otimes \mathcal{H}_2) = U_1(\mathcal{H}_1) \otimes U_2(\mathcal{H}_2)$.

Often, when we write "rando" we mean "uniformly random". Many terms, which we are going to use throughout this paper, are actually a function of the implicit security parameter η, however in order to keep notations simple, we refuse in most cases to make the dependence of η explicit, and just omit η. Quantum registers are denoted by Q with possibly some index. We will use the notation of U_f, $\hat{U}^g$ for arbitrary f, arbitrary injective g where

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle \quad \text{and} \quad \hat{U}^g : |x\rangle \mapsto |g(x)\rangle.$$

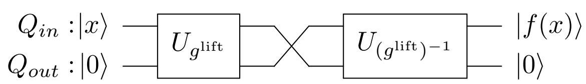
2.1 Realizability of $\hat{U}^g$ as a quantum circuit

The linear operator $\hat{U}^g$ is mathematically well defined however we have to argue that it can also be realized in a quantum computer efficiently whenever g is efficiently computable and reversible, classically. In order to do so we introduce a new concept, which we call the lifting of a classical injective function. Namely

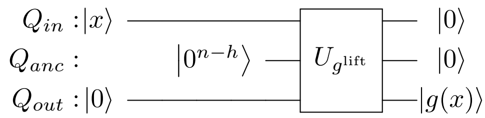
Definition 1. *For an arbitrary injective $g : \{0,1\}^h \hookrightarrow \{0,1\}^n$ we call $g^{\text{lift}} : \{0,1\}^h \xrightarrow{\approx} \{0,1\}^n$ some chosen (but in a fixed way) bijective function such that*

$$\forall x \in \{0,1\}^h : g^{\text{lift}}(x||0^{n-h}) = g(x)$$

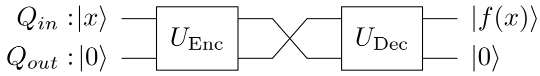
That is, g^{lift} as just an arbitrary extension of g with a bigger domain, so that g^{lift} is bijective. Now we implement $\hat{U}^g$ using its inverse.



where $U_{g^{\text{lift}}}$ is implemented as the following:



Note that for an injective function g if there exists an efficiently computable function g^{-1} such that $g^{-1}(g(x)) = x$, then we can implement *ER* type query without the ancillary register. For instance, this is the case for encryption scheme and its decryption:



2.2 Cryptographic Assumptions

We are using the random oracle model at different places, however many results can also be derived using only the existence of quantum one-way functions.

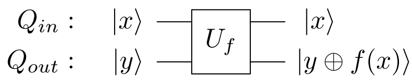
3 Definitions

One of the main points in this text is to compare different ways to model how a quantum-circuit can access a classical function (i.e., how to represent a classical function as a quantum gate). There are 3 query models that model this, here called *ST* (standard query model), *EM* (embedding query model) and *ER* (erasing query model). *EM* is in some sense the "weakest" in that it can be simulated by both *ST* and *ER*. Let

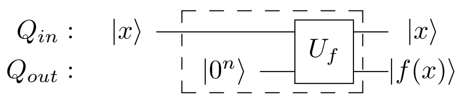
$$f : \{0,1\}^h \to \{0,1\}^n$$

be a deterministic function.

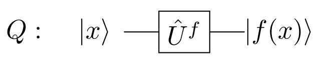
ST-query model: In this query model, an algorithm A that queries f provides two registers Q_{in}, Q_{out} of h and n q-bits, respectively. Then, the unitary $U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$ is applied to these registers and finally the registers Q_{in}, Q_{out} are passed back to A. We depict the quantum circuit corresponding to this query model as follows.



EM-query model: , The difference of the *EM*-query model with the *ST*-model is that the lower wire (called "output-wire") is forced to contain 0^n and is not part of the input to quantum circuit but produced locally. In other words, an algorithm A provides a register Q_{in} of h qubits and Q_{out} is initialized as 0^n and then the unitary U_f is applied to registers Q_{in}, Q_{out} and they are passed back to A. The following quantum circuit illustrates this query model.



ER-query model: This query model is only possible for functions f that are injective.



Note that the *ST* and *EM* oracles for a classical function f can be constructed in canonical way from a classical circuit that computes f [NC16] and the *ER* oracle constructed in 2.1.

Definition 2. *A triple* (KGen, Enc, Dec) *is called an* (h, n, n', t, t')*-encryption scheme (note that these parameters depend on* η*) iff*

$$\text{KGen} : \{0,1\}^{t'} \to \{0,1\}^h$$
$$\text{Enc} : \{0,1\}^h \times \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$$
$$\text{Dec} : \{0,1\}^h \times \{0,1\}^{n'} \to \{0,1\}^n \cup \{\perp\}$$

such that

$$\forall k \in \{0,1\}^h, m \in \{0,1\}^n, r \in \{0,1\}^t : \mathrm{Dec}_k(\mathrm{Enc}_k(m;r)) = m$$

(Note that an encryption scheme is by definition always entirely classical.) Here $\mathrm{Enc}_k(m;r)$ *is written instead of* $\mathrm{Enc}(k,m,r)$ *and* $\mathrm{Dec}_k(c)$ *instead of* $\mathrm{Dec}(k,c)$. Enc *is called the encryption function,* Dec *is called the decryption function and* KGen *is called the key generation function.* $\{0,1\}^h$ *is the key space,* $\{0,1\}^n$ *is the message (plaintext) space,* $\{0,1\}^{n'}$ *is the ciphertext space,* $\{0,1\}^t$ *is the randomness space and* $\{0,1\}^{t'}$ *is the key randomness space. The decryption is allowed but not required to output* $\perp$ *for an invalid ciphertext. The encryption algorithm samples an element of* $\{0,1\}^t$ *uniformly at random and then invokes the encryption function. The key generation algorithm samples an element of* $\{0,1\}^{t'}$ *uniformly at random and then invokes the key generation function.*

For simplicity, we allow ourselves to write $k \xleftarrow{\$} \mathrm{KGen}()$ *instead of* $kr \xleftarrow{\$} \{0,1\}^{t'}, k := \mathrm{KGen}(kr)$ *and* $c \xleftarrow{\$} \mathrm{Enc}_k(m)$ *instead of* $r \xleftarrow{\$} \{0,1\}^t, c := \mathrm{Enc}_k(m;r)$.

Definition 3. *For natural numbers h and n, two functions* $f_1 : \{0,1\}^h \to \{0,1\}^n$ *and* $f_2 : \{0,1\}^h \to \{0,1\}^n$ *are called c-indistinguishable (short for classically indistinguishable) iff there exists a negligible* ε *such that for all classical polynomial time oracle algorithms (adversaries)* $\hat{\mathcal{A}}$ *we have:*

$$|\mathrm{Prob}[1 \leftarrow \hat{\mathcal{A}}^{f_1}()] - \mathrm{Prob}[1 \leftarrow \hat{\mathcal{A}}^{f_2}()]| < \varepsilon,$$

(Note, that the definition of c-indistinguishability is never used in the paper, it is just mentioned for reference purposes) We call f_1, f_2 *s-indistinguishable (short for standard indistinguishable) or CL-q-indistinguishable iff there exists a negligible* ε *such that for all quantum polynomial time oracle algorithms (adversaries)* $\mathcal{A}$ *and all auxiliary quantum states* $|\psi\rangle$ *it holds:*

$$|\mathrm{Prob}[1 \leftarrow \mathcal{A}^{CL(f_1)}(|\psi\rangle)] - \mathrm{Prob}[1 \leftarrow \mathcal{A}^{CL(f_2)}(|\psi\rangle)]| < \varepsilon,$$

We call f_1, f_2 *qm-q-indistinguishable (short for (query model)-quantum-indistinguishable) for* $qm \in \{CL, ST, ER\}$ *(note that we are not considering EM) iff there exists a negligible* ε *such that for all quantum polynomial time oracle algorithms (adversaries)* $\mathcal{A}$ *making polynomial number of queries to its oracle in the query model qm and all auxiliary quantum states* $|\psi\rangle$ *it holds:*

$$|\mathrm{Prob}[1 \leftarrow \mathcal{A}^{qm(f_1)}(|\psi\rangle)] - \mathrm{Prob}[1 \leftarrow \mathcal{A}^{qm(f_2)}(|\psi\rangle)]| < \varepsilon.$$

Note that s-indistinguishability is the same as CL-q-indistinguishability.

We call a pseudorandom permutation π_s a vPRP for $v \in \{c, s, q\}$, iff it is v-indistiguishable from a truly random permutation.

That means, that cPRP (classically pseudorandom permutation), sPRP (standard pseudorandom permutation = quantum resistant pseudorandom permutation) and qPRP (quantum pseudorandom permutation) can be defined like this (Note that we mean strong PRP whenever we say PRP, we are not considering weak PRPs):

- With **cPRP** is meant a pseudorandom permutation π_s which is secure against a **classical** adversary with **classical** access to π_s and π_s^{-1}.
- With **sPRP** is meant a pseudorandom permutation π_s which is secure against a **quantum** adversary with **classical** access to π_s and π_s^{-1}.
- With **qPRP** is meant a pseudorandom permutation π_s which is secure against a **quantum** adversary with **superposition access** to π_s and π_s^{-1}.

Formally ST-qPRP and ER-qPRP have to be dinstinguished, but as shown below they are equivalent. More formally cPRP, sPRP, qPRP are defined by:

Definition 4. *A* (m,n)*-v-strong-PRP (also called block cipher) for* $v \in \{c, s, q\}$ *is a pair of two permutations (= bijective functions)* π *and* π^{-1} *with seed s:*

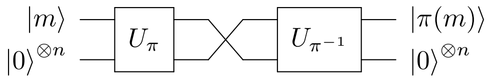
$$\pi_s, \pi_s^{-1} : \{0,1\}^n \to \{0,1\}^n, s \in \{0,1\}^m$$

sucht that the oracle $f_1(x) = \pi_s(x)$ *is v-indistinguishable from a truly random permutation* $f_2 : \{0,1\}^n \to \{0,1\}^n$.

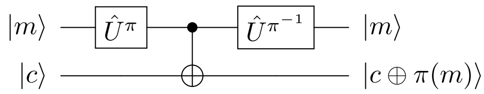
Remark 1. *Note that Zhandry showed in [Zha16] that a qPRP (ST-query-model) can be constructed from a one-way-function. Also we are not distinguishing qPRP in the ST-query-model and in the ER-query-model. The next lemma we will justify that by proving that ST-q-PRP-oracles and ER-q-PRP-oracles can be constructed out of each other by a simple construction.*

Lemma 1. *A bijection π is a strong ST-q-PRP iff it is a strong ER-q-PRP.*

Proof. The reason is, that ST and ER query models can be constructed out of each other if the oracle function is a permutation and with access to its inverse. The following circuit shows how a ER query can be simulated by ST queries to π and π^{-1}:



The following circuit shows how a ST query can be simulated by ER queries to π and π^{-1}:



□

Next we have to define what it means for an encryption scheme to fulfill a certain security notion. Namely we will define what it means to be $\mathfrak{l}$-$\mathfrak{c}$-IND-CPA-secure. Here $\mathfrak{l}$ and $\mathfrak{c}$ are just symbols which will be instantiated later. $\mathfrak{l}$ stands for learning query and $\mathfrak{c}$ stands for challenge query. Accordingly $\mathfrak{l}$ will be instantiated with some learning query model and $\mathfrak{c}$ will be instantiated with some challenge query model.

Definition 5. *We say the encryption scheme Enc* $= (\mathrm{KGen}, \mathrm{Enc}, \mathrm{Dec})$ *is $\mathfrak{l}$-$\mathfrak{c}$-IND-CPA-secure if any polynomial time quantum adversary $\mathcal{A}$ can win in the following game with probability at most $\frac{1}{2} + \epsilon$ for some negligible ϵ.*

<u>*The $\mathfrak{l}$-$\mathfrak{c}$-CPA game:*</u>
Key Gen: *The challenger runs* KGen *to obtain a key k, i.e, $k \xleftarrow{\$} \mathrm{KGen}()$.*
Learning Queries: *The challenger answers to the $\mathfrak{l}$-type queries of $\mathcal{A}$ using* Enc_k*. $\mathfrak{l}$ also specifies the number of times this step can be repeated.*
Challenge Queries: *The challenger picks a random bit b and answers to the $\mathfrak{c}$-type queries of $\mathcal{A}$ using* Enc_k *and the bit b. (Note that the adversary is allowed to submit some learning queries between the challenge queries as well.) $\mathfrak{c}$ also specifies the number of times this step can be repeated.*
Learning Queries: *The challenger answers to the $\mathfrak{l}$-type queries of $\mathcal{A}$ using* Enc_k*. $\mathfrak{l}$ also specifies the number of times this step can be repeated.*
Guess: *The adversary $\mathcal{A}$ returns a bit b', and wins if $b' = b$.*

In the two sections below, we define different types of the learning queries and the challenge queries and we specify which combination of them are considered for IND-CPA security of encryption schemes.

3.1 Syntax of $\mathfrak{l}$ - the learning queries

Note that in all of the following query models, we assume the challenger picks $k \xleftarrow{\$} \mathrm{KGen}()$. For simplicity, we omit it from our description.

3.1.1 Learning Query type CL

For any query on input message m, the challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and gives back $c \leftarrow \mathrm{Enc}_k(m;r)$ to the adversary.

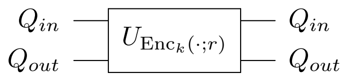
Input registers: None.
Classical input: $m \in \{0,1\}^n$.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t, c \leftarrow \mathrm{Enc}_k(m;r)$.
Quantum computation: None.

Output registers: None.
Classical output: $c \in \{0,1\}^{n'}$.

3.1.2 Learning Query type ST

For any query, the challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and applies the unitary U_{Enc_k} to the provided registers of the adversary, Q_{in}, Q_{out} registers, and gives them back to the adversary.

Input registers: Q_{in}, Q_{out}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t$.
Quantum computation:

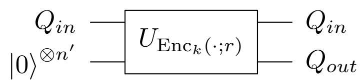


Output registers: Q_{in}, Q_{out}.
Classical output: None.

3.1.3 Learning Query type EM

Upon receiving the provided register of the adversary, say Q_{in} and Q_{out}, the challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and creates a register Q_{out} containing the state $|0\rangle^{\otimes n}$ and applies the unitary U_{Enc_k} to the registers Q_{in}, Q_{out}, and gives them back to the adversary.

Input registers: Q_{in}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t$.
Quantum computation:

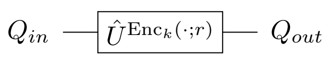


Output registers: Q_{in}, Q_{out}.
Classical output: None.

3.1.4 Learning Query type ER

Upon receiving the provided register of the adversary, say Q_{in}, the challenger picks $r \xleftarrow{\$} \{0,1\}^t$, applies the unitary $\hat{U}^{Enc_k(\cdot,r)}$ to the register Q_{in} and gives it back to the adversary.

Input registers: Q_{in}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t$.
Quantum computation:



Output registers: Q_{out}.
Classical output: None.
Note that $\hat{U}^{\mathrm{Enc}_k(\cdot;r)}$ is physically realizable because Enc_k is efficiently reversible for fixed r using Dec_k(see Section 2.1).

3.2 Syntax of $\mathfrak{c}$ - the challenge queries

First we give an informal overview over the different challenge query types, then we define each of them in a concise way:
Overview:

- chall$(\cdot, CL, 1\text{ct})$ $m_0, m_1 \mapsto \text{Enc}_k(m_b, r)$ classically
- chall$(\cdot, CL, 2\text{ct})$ $m_0, m_1 \mapsto \text{Enc}_k(m_b, r), \text{Enc}_k(m_{\bar{b}}, r)$ classically
- chall$(\cdot, CL, \text{ror})$ $m \mapsto \text{Enc}_k(m, r)$ or $\text{Enc}_k(r^*, r)$ classically
- chall$(\cdot, ST, 1\text{ct})$ $|m_0, m_1, c\rangle \mapsto |m_0, m_1, c \oplus \text{Enc}_k(m_b; r)\rangle$
- chall$(\cdot, ST, 2\text{ct})$ $|m_0, m_1, c_0, c_1\rangle \mapsto |m_0, m_1, c_0 \oplus \text{Enc}_k(m_b; r_b), c_1 \oplus \text{Enc}_k(m_{\bar{b}}; r_{\bar{b}})\rangle$
- chall$(\cdot, ST, \text{ror})$ $|m, c\rangle \mapsto \left|m, c \oplus \text{Enc}_k(\pi^b(m); r)\right\rangle$ for a random permuation π
- chall$(\cdot, EM, 1\text{ct})$ $|m_0, m_1, 0\rangle \mapsto |m_0, m_1, \text{Enc}_k(m_b; r)\rangle$
- chall$(\cdot, EM, 2\text{ct})$ $|m_0, m_1, 0, 0\rangle \mapsto |m_0, m_1, \text{Enc}_k(m_b; r_b), \text{Enc}_k(m_{\bar{b}}; r_{\bar{b}})\rangle$
- chall$(\cdot, EM, \text{ror})$ $|m, 0\rangle \mapsto \left|m, \text{Enc}_k(\pi^b(m); r)\right\rangle$ for a random permuation π
- chall$(\cdot, ER, 1\text{ct})$ $|m_0, m_1\rangle \mapsto |\text{Enc}_k(m_b; r)\rangle$ and trace out $|m_{\bar{b}}\rangle$
- chall$(\cdot, ER, 2\text{ct})$ $|m_0, m_1\rangle \mapsto |\text{Enc}_k(m_b; r), \text{Enc}_k(m_{\bar{b}}; r)\rangle$
- chall$(\cdot, ER, \text{ror})$ $|m\rangle \mapsto \left|\text{Enc}_k(\pi^b(m); r)\right\rangle$ for a random permuation π

Using a the permutation π in this way, is a general way of construction real-or-random-like quantum query models and first appeared in [MS16]. The idea behind it is that a random permutation π in some way replaces an plaintext m with a random bitstring, as this would be the case classically.

3.2.1 Challenge Query type chall$(\cdot, CL, 1\text{ct})$

(The notation 1ct stands for one-ciphertext.)
In this query model, the adversary picks two messages m_0, m_1 and sends them to the challenger. The challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and a random bit b and returns $Enc_k(m_b; r)$

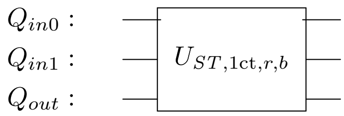
Input registers: None.
Classical input: $m_0, m_1 \in \{0,1\}^n$
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\},\ c \leftarrow \text{Enc}_k(m_b; r)$.
Quantum computation: None.
Output registers: None.
Classical output: $c \in \{0,1\}^{n'}$.

3.2.2 Challenge Query type chall$(\cdot, ST, 1\text{ct})$

In this query model, the adversary prepares two input registers Q_{in0}, Q_{in1}, one output register Q_{out} and sends them to the challenger. The challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the registers to the adversary.

$$U_{ST,1\text{ct},r,b} : |m_0, m_1, c\rangle \mapsto |m_0, m_1, c \oplus \text{Enc}_k(m_b; r)\rangle.$$

Input registers: $Q_{in0}, Q_{in1}, Q_{out}$
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$.
Quantum computation:



where

$$U_{ST,1\text{ct},r,b} : |m_0, m_1, c\rangle \mapsto |m_0, m_1, c \oplus \text{Enc}_k(m_b; r)\rangle$$

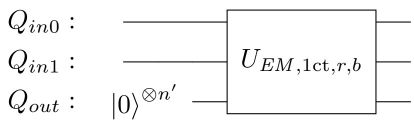
Output registers: $Q_{in0}, Q_{in1}, Q_{out}$.
Classical output: None.

3.2.3 Challenge Query type chall($\cdot$, $\boldsymbol{EM}$, 1ct)

In this query model, the adversary prepares two input registers Q_{in0}, Q_{in1}, and sends them to the challenger. The challenger prepares an output register Q_{out} containing $|0\rangle^{\otimes n'}$, picks $r \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the registers to the adversary.

$$U_{EM,1\text{ct},r,b} : |m_0, m_1, c\rangle \mapsto |m_0, m_1, c \oplus \text{Enc}_k(m_b; r)\rangle.$$

Input registers: Q_{in0}, Q_{in1}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$.
Quantum computation:



where

$$U_{ST,1\text{ct},r,b} : |m_0, m_1, c\rangle \mapsto |m_0, m_1, c \oplus \text{Enc}_k(m_b; r)\rangle$$

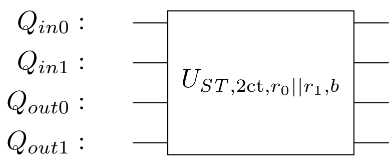
Output registers: $Q_{in0}, Q_{in1}, Q_{out}$.
Classical output: None.

3.2.4 Challenge Query type chall($\cdot$, $\boldsymbol{ST}$, 2ct)

(The notation 2ct stands for two-ciphertexts.)
In this query model, the adversary prepares two input registers Q_{in0}, Q_{in1}, two output registers Q_{out0}, Q_{out1} and sends them to the challenger. The challenger picks $r_0, r_1 \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the registers to the adversary.

$$U_{ST,2\text{ct},r_0||r_1,b} : |m_0, m_1, c_0, c_1\rangle \mapsto |m_0, m_1, c_0 \oplus \text{Enc}_k(m_b; r_0), c_1 \oplus \text{Enc}_k(m_{1-b}; r_1)\rangle.$$

Input registers: $Q_{in0}, Q_{in1}, Q_{out0}, Q_{out1}$.
Classical input: None.
Classical computation: $r_0 \xleftarrow{\$} \{0,1\}^t,\ r_1 \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$
Quantum computation:



where

$$U_{ST,2\mathrm{ct},r_0||r_1,b} : |m_0, m_1, c_0, c_1\rangle \mapsto |m_0, m_1, c_0 \oplus \mathrm{Enc}_k(m_b; r_0), c_1 \oplus \mathrm{Enc}_k(m_{1-b}; r_1)\rangle.$$

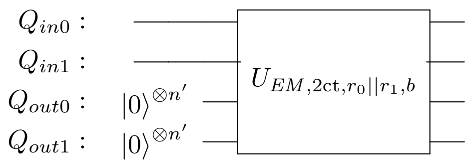
Output registers: $Q_{in0}, Q_{in1}, Q_{out0}, Q_{out1}$.
Classical output: None.

3.2.5 Challenge Query type chall(·, *EM*, 2ct)

In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} and sends them to the challenger. The challenger prepares two registers Q_{out0}, Q_{out1} containing $|0\rangle^{\otimes n'}$, picks $r_0, r_1 \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the registers to the adversary.

$$U_{EM,2\mathrm{ct},r_0||r_1,b} : |m_0, m_1, c_0, c_1\rangle \mapsto |m_0, m_1, c_0 \oplus \mathrm{Enc}_k(m_b; r_0), c_1 \oplus \mathrm{Enc}_k(m_{1-b}; r_1)\rangle.$$

Input registers: Q_{in0}, Q_{in1}.
Classical input: None.
Classical computation: $r_0 \xleftarrow{\$} \{0,1\}^t,\ r_1 \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$.
Quantum computation:



where

$$U_{ST,2\mathrm{ct},r_0||r_1,b} : |m_0, m_1, c_0, c_1\rangle \mapsto |m_0, m_1, c_0 \oplus \mathrm{Enc}_k(m_b; r_0), c_1 \oplus \mathrm{Enc}_k(m_{1-b}; r_1)\rangle.$$

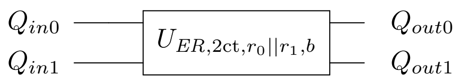
Output registers: $Q_{in0}, Q_{in1}, Q_{out0}, Q_{out1}$.
Classical output: None.

3.2.6 Challenge Query type chall(·, *ER*, 2ct)

In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} of h qubits and sends them to the challenger. The challenger picks $r_0, r_1 \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the outcome to the adversary.

$$U_{ER,2\mathrm{ct},r_0||r_1,b} : |m_0, m_1\rangle \mapsto |\mathrm{Enc}_k(m_b; r_0), \mathrm{Enc}_k(m_{1-b}; r_1)\rangle$$

Input registers: Q_{in0}, Q_{in1}.
Classical input: None.
Classical computation: $r_0 \xleftarrow{\$} \{0,1\}^t,\ r_1 \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$.
Quantum computation:

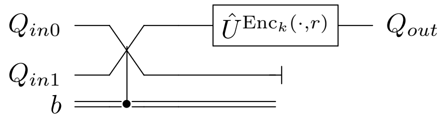


Output registers: Q_{out0}, Q_{out1}.
Classical output: None.

3.2.7 Challenge Query type chall(·, *ER*, 1ct)

In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} of h qubits and sends them to the challenger. The challenger picks $r \xleftarrow{\$} \{0,1\}^t$ and a random bit b, measures the register $Q_{in\bar{b}}$ (one of the provided registers by the adversary) and throws out the result, applies the unitary $\hat{U}^{Enc_k(\cdot,r)}$ to the registers Q_{inb}, and passes them back to the adversary.

Input registers: Q_{in0}, Q_{in1}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$.
Quantum computation:



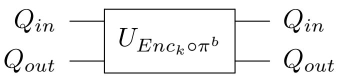
Output registers: Q_{out}.
Classical output: None.

3.2.8 Challenge Query type chall(·, *ST*, ror)

(The notation ror stands for "real or random".)

In this query model, the adversary provides two registers Q_{in}, Q_{out}. The challenger picks $r \xleftarrow{\$} \{0,1\}^t$, $b \xleftarrow{\$} \{0,1\}$, a random permutation π on $\{0,1\}^n$, applies the unitary $U_{Enc_k \circ \pi^b}$ to Q_{in}, Q_{out} and passes them back to the adversary.

Input registers: Q_{in}, Q_{out}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\},\ \pi \xleftarrow{\$} \{\pi | \pi : \{0,1\}^n \to \{0,1\}^n \text{ is a permutation}\}$
Quantum computation:

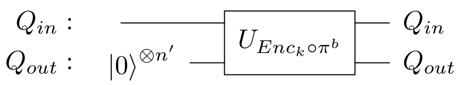


Output registers: Q_{in}, Q_{out}.
Classical output: None.

3.2.9 Challenge Query type chall(·, *EM*, ror)

In this query model, the adversary provides a register Q_{in}. The challenger prepares a register Q_{out} containing $|0\rangle^{\otimes n'}$, picks $r \xleftarrow{\$} \{0,1\}^t$, $b \xleftarrow{\$} \{0,1\}$, a random permutation π on $\{0,1\}^n$, applies the unitary $U_{Enc_k \circ \pi^b}$ to Q_{in}, Q_{out} and passes them back to the adversary.

Input registers: Q_{in}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\},\ \pi \xleftarrow{\$} \{\pi | \pi : \{0,1\}^n \to \{0,1\}^n \text{ is a permutation}\}$.
Quantum computation:



Output registers: Q_{in}, Q_{out}.
Classical output: None.

3.2.10 Challenge Query type chall($\cdot$, $\boldsymbol{ER}$, ror)

In this query model, the adversary prepares a register Q_{in} of h qubits and sends it to the challenger. The challenger picks $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\}$, a random permutation π on $\{0,1\}^n$, applies the following operation to the register Q_{in}, and passes it back to the adversary.

$$U_{ER,\mathrm{ror},r,b} : |m\rangle \mapsto \left|\mathrm{Enc}_k(\pi^b(m); r)\right\rangle$$

Input registers: Q_{in}.
Classical input: None.
Classical computation: $r \xleftarrow{\$} \{0,1\}^t,\ b \xleftarrow{\$} \{0,1\},\ \pi \xleftarrow{\$} \{\pi | \pi : \{0,1\}^h \to \{0,1\}^h$ is a permutation$\}$.
Quantum computation:

$$Q_{in} \;—\boxed{U_{ER,\mathrm{ror},r,b}}—\; Q_{out}$$

Output registers: Q_{out}.
Classical output: None.
Note that the circuit above is physically realizable because Enc_k and π are injective for fixed r. We give an alternative circuit for the above operation:

$$Q_{in} \;—\boxed{\hat{U}^{\pi^b}}—\boxed{\hat{U}^{\mathrm{Enc}(\cdot,r)}}—\; Q_{out}$$

3.3 Instantiation of learning and challenge query models

We define $\mathfrak{l} := \mathrm{learn}(\mathfrak{l}_{nb}, \mathfrak{l}_{qm})$ ("nb" stands for "number", "qm" stands for "query model") where $\mathfrak{l}_{nb}$ shows the number of the learning queries and $\mathfrak{l}_{qm}$ shows the type of the learning queries. Therefore, $\mathfrak{l} = \mathrm{learn}(\mathfrak{l}_{nb}, \mathfrak{l}_{qm})$ where $(\mathfrak{l}_{nb}, \mathfrak{l}_{qm}) \in (\{*\} \times \{CL, ST, EM, ER\}) \cup \{(0, -)\}$ where $*$ means arbitrary many queries and 0 means no learning queries. For the challenge queries, we define $\mathfrak{c} := \mathrm{chall}(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt})$ ("nb" stands for "number", "qm" stands for "query model", "rt" stands for "return type") where $\mathfrak{c}_{nb}$ shows the number of the challenge queries and $\mathfrak{c}_{qm},\ \mathfrak{c}_{rt}$ show the type of the challenge queries. Therefore, $\mathfrak{c} = \mathrm{chall}(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt})$ where $(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt}) \in \{1, *\} \times \{CL, ST, EM, ER\} \times \{1\mathrm{ct}, 2\mathrm{ct}, \mathrm{ror}\}$

Number of queries:

- 0: Zeros queries (only allowed for learning queries, otherwise the notion becomes trivial)
- 1: One query (only allowed for challenge queries)
- $*$: arbitrary many queries

Query models:

- CL: Classical queries
- ST: Standard quantum queries
- EM: Embedding quantum queries
- ER: Erasing quantum queries

Return types: (only relevant for challenge queries)

- 1ct: One-ciphertext, that is, the adversary sends two plaintexts m_0 and m_1, but only one of them, m_b is encrypted.
- 2ct: Two-ciphertexts, that is, the adversary sends two plaintexts m_0 and m_1 and both of them are encrypted and the adversary has to guess which ciphertext corresponds to which plaintext.
- ror: Real or random, that is, the adversary sends one plaintext m, and he gets either the encryption of m or of a $\pi(m)$ where π is a random permutation on the plaintext space.

3.4 The valid combinations of the learning and challenge queries

In Definition 5, we defined the security of an encryption scheme in the sense of $\mathfrak{l}$-$\mathfrak{c}$-IND-CPA. Now, we explicitly specify which combination of the learning queries, $\mathfrak{l}$, and the challenge queries, $\mathfrak{c}$, are considered in this paper.

The valid combinations. We consider only combinations where,

- $(\mathfrak{l}_{nb}, \mathfrak{c}_{nb}) \in \{(*, 1), (*, *), (0, *)\}$ i.e. $(\mathfrak{l}_{nb}, \mathfrak{c}_{nb}) \neq (0, 1)$. Which means we are not considering variants of IND-OT-CPA (which is security of encryption only used once)
- $(\mathfrak{l}_{qm}, \mathfrak{c}_{qm}) \in \{(CL, CL)\} \cup \{(CL, x), (x, CL), (x, x) | x \in \{ST, EM, ER\}\}$, i.e, if learning queries and challenge queries are both quantum they are not allowed to be from different query models. This is to keep the combinatorial explosion of different notions in check, and notions that combine two different notions of superposition queries strike as rather exotic.

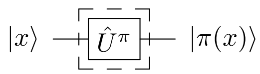
4 Decoherence lemmas

The informal idea of the following lemma is, that if you have one-time access to an *ER*-type oracle of a random permutation, you cannot distinguish whether this oracle "secretely" applies a projective measurement to your input, that measures whether your input is $|+\rangle^{\otimes m}$ and if not which computational state $|x\rangle$ it is.

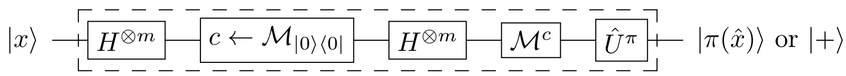
Lemma 2. *For a bijective function $\pi : \{0,1\}^m \to \{0,1\}^m$ let $\hat{U}^\pi$ be the unitary that performs the ER-type mapping $|x\rangle \mapsto |\pi(x)\rangle$. Let X be a quantum register with m qubits. Then the following two oracles can be distinguished in a single query with probability at most 2^{-m+1}:*

- *F_0: Pick a random permutation π and apply $\hat{U}^\pi$ on X,*
- *F_1: Pick a random permutation π, measure X as described later and then apply $\hat{U}^\pi$ to the result.*

The quantum circuit for F_0 is:

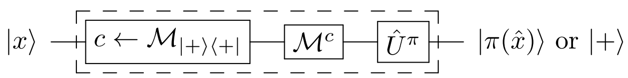


and for F_1 it is:



where $c \leftarrow \mathcal{M}_{|0\rangle\langle 0|}$ is a projective measurement, storing the result (0 or 1) in c, that projects to the spaces $\mathrm{span}(|0\rangle^{\otimes m})$ *(corresponding to* 0*) and its orthogonal space (corresponding to* 1*) and $\mathcal{M}^1$ is a measurement in the computational basis, whose outcome is denoted by $\hat{x}$ and $\mathcal{M}^0$ means no operation.*

Note, that if we write $\mathcal{M}_{|+\rangle\langle+|}$ for the projective measurement, that projects to the subspace $\mathrm{span}(|+\rangle^{\otimes m})$, we can write F_1 simply as:



Proof. Let $M := 2^m$ and $F := M!$ (the number of possible permutations π). A general strategy for distinguishing F_0 and F_1 can be described as follows: The adversary chooses some Hilbert space $\mathcal{H}$ and for each $x \in \{0,1\}^m$ picks $\hat{\alpha}_x \in \mathbb{C}$, normalized $|\phi_x\rangle \in \mathcal{H}$ such that $\sum_{x \in X} |\hat{\alpha}_x|^2 = 1$. The adversary then prepares the bipartite state

$$|\Psi\rangle_{AB} := \sum_{x \in \{0,1\}^m} \hat{\alpha}_x |\phi_x\rangle_A \otimes |x\rangle_B$$

and sends the B-part as the input of an oracle query to f. (We can assume this without loss of generality, because any state $|\Psi\rangle_{AB}$ can be written in this form.) Let ρ_0 be the density operator of the state after applying the oracle in F_0 to $|\Psi\rangle$. Let ρ_1 be the density operator of the state after applying the oracle in F_1 to $|\Psi\rangle$. Let ρ' be the density operator of the state in F_1 if the computational measurement $\mathcal{M}^c$ is omitted. Decompose $|\Psi\rangle$ as:

$$|\Psi\rangle = \gamma_{\text{yes}} |\psi_{\text{yes}}\rangle + \gamma_{\text{no}} |\psi_{\text{no}}\rangle$$

such that $|\psi_{\text{yes}}\rangle \in \mathcal{H} \otimes \text{span}\{|+\rangle^{\otimes m}\}$ and $|\psi_{\text{no}}\rangle \in \mathcal{H} \otimes \text{span}\{|+\rangle^{\otimes m}\}^{\perp}$. Now choose quantum states $|\Phi\rangle$ and $(|\psi_x\rangle)_x$ and scalars β and $(\alpha_x)_{x\in X}$ such that

$$\gamma_{\text{yes}}|\psi_{\text{yes}}\rangle = \beta|\Phi\rangle \otimes |+\rangle^{\otimes m}$$

and

$$\gamma_{\text{no}}|\psi_{\text{no}}\rangle = \sum_x (\alpha_x|\psi_x\rangle \otimes |x\rangle)$$

so then

$$|\Psi\rangle = \beta|\Phi\rangle \otimes |+\rangle^{\otimes m} + \sum_x (\alpha_x|\psi_x\rangle \otimes |x\rangle)$$

and such that $\mathcal{H} \otimes \text{span}\{|+\rangle^{\otimes m}\}$ is orthogonal to $\sum_x (\alpha_x|\psi\rangle \otimes |x\rangle^{\otimes m})$. To simplify computation choose quantum states $|\psi_{\text{yes}}\rangle$ and $|\psi_{\text{no}}\rangle$ and scalars γ_{yes} and γ_{no} ("yes" corresponds to measuring $c = 0$ and "no" corresponds to measuring $c = 1$).
Now we prove

Claim 1.

$$\sum_x \alpha_x|\psi_x\rangle = 0$$

Proof (of Claim):

$$\sum_x \alpha_x|\psi_x\rangle = \sum_{x,y} (I \otimes \langle y|)(\alpha_x|\psi_x\rangle \otimes |x\rangle)$$
$$= 2^{\frac{n}{2}} \left(I \otimes \langle +|^{\otimes m}\right) \sum_x (\alpha_x|\psi_x\rangle \otimes |x\rangle) = \; 2^{\frac{n}{2}} \left(I \otimes \langle +|^{\otimes m}\right)\gamma_{\text{no}}|\psi_{\text{no}}\rangle$$

But by the choice of $\gamma_{\text{no}}|\psi_{\text{no}}\rangle$ this is 0.
This proves the claim.

Now we show that $\rho_0 = \rho'$ and then we show that $\text{TD}(\rho_0, \rho_1)$ (which is equal to $\text{TD}(\rho', \rho_1)$) is negligible.

Claim 2.

$$\gamma_{\text{no}} \sum_\pi (I \otimes \hat{U}^\pi)|\psi_{\text{no}}\rangle = 0$$

Proof (of Claim):

$$\gamma_{\text{no}} \sum_\pi (I \otimes \hat{U}^\pi)|\psi_{\text{no}}\rangle = \sum_\pi (I \otimes \hat{U}^\pi) \sum_x (\alpha_x|\psi_x\rangle \otimes |x\rangle)$$
$$= \sum_\pi \sum_x (\alpha_x|\psi_x\rangle \otimes |\pi(x)\rangle) = \sum_\pi \sum_y (\alpha_y \left|\psi_{\pi^{-1}(y)}\right\rangle \otimes |y\rangle)$$
$$= \sum_y \sum_\pi (\alpha_y \left|\psi_{\pi^{-1}(y)}\right\rangle \otimes |y\rangle) = \sum_y \sum_x \sum_{\pi:\pi^{-1}(y)=x} (\alpha_x|\psi_x\rangle \otimes |y\rangle)$$
$$= \sum_y \sum_x \frac{M!}{M} \cdot (\alpha_x|\psi_x\rangle \otimes |y\rangle) = \frac{M!}{M} \cdot \sum_x |\psi_x\rangle \otimes \sum_y \alpha_y|y\rangle$$
$$\overset{(i)}{=} \frac{M!}{M} \cdot \sum_x |\psi_x\rangle \otimes 0 = \; 0$$

where (i) follows from Claim 1.
This proves the claim.

Claim 3.

$$(I \otimes \hat{U}^\pi)(\gamma_{\text{yes}}|\psi_{\text{yes}}\rangle) = \gamma_{\text{yes}}|\psi_{\text{yes}}\rangle$$

Proof (of Claim): This hold because $\gamma_{\text{yes}}|\psi_{\text{yes}}\rangle = \beta|\Phi\rangle \otimes |+\rangle^{\otimes m}$ and $\hat{U}^\pi|+\rangle^{\otimes m} = 2^{-\frac{n}{2}}\sum_x |\pi(x)\rangle = |+\rangle^{\otimes m}$.
This proves the claim.

Claim 4.

$$\rho_0 = \rho'$$

Proof (of Claim): This can be shown by proving that $\rho_0 - \rho' = 0$. We know that

$$\rho_0 = \frac{1}{M!}\sum_\pi (I \otimes \hat{U}^\pi)|\Psi\rangle\langle\Psi|(I \otimes \hat{U}^\pi)^\dagger$$

and

$$|\Psi\rangle = \gamma_{\text{yes}}|\psi_{\text{yes}}\rangle + \gamma_{\text{no}}|\psi_{\text{no}}\rangle$$

Defining the shorthand

$$\left|\psi'_{\text{yes}}\right\rangle := (I \otimes \hat{U}^\pi)\gamma_{\text{yes}}|\psi_{\text{yes}}\rangle = \gamma_{\text{yes}}|\psi_{\text{yes}}\rangle$$

and

$$\left|\psi'_{\text{no},\pi}\right\rangle := (I \otimes \hat{U}^\pi)\gamma_{\text{no}}|\psi_{\text{no}}\rangle$$

we can write

$$\rho_0 = \frac{1}{M!}\sum_\pi \left((\left|\psi'_{\text{yes}}\right\rangle + \left|\psi'_{\text{no},\pi}\right\rangle)(\left\langle\psi'_{\text{yes}}\right| + \left\langle\psi'_{\text{no},\pi}\right|)\right)$$

and

$$\rho' = \frac{1}{M!}\sum_\pi \left(\left|\psi'_{\text{yes}}\right\rangle\left\langle\psi'_{\text{yes}}\right| + \left|\psi'_{\text{no},\pi}\right\rangle\left\langle\psi'_{\text{no},\pi}\right|\right)$$

so that means that

$$\begin{aligned}\rho_0 - \rho' &= \frac{1}{M!}\sum_\pi(\left|\psi'_{\text{yes}}\right\rangle\left\langle\psi'_{\text{no},\pi}\right| + \left|\psi'_{\text{no},\pi}\right\rangle\left\langle\psi'_{\text{yes}}\right|)\\ &= \left|\psi'_{\text{yes}}\right\rangle(\sum_\pi \left\langle\psi'_{\text{no},\pi}\right|) + (\sum_\pi \left|\psi'_{\text{no},\pi}\right\rangle)\left\langle\psi'_{\text{yes}}\right|\end{aligned}$$

so this is 0 as Claim 2 implies $\sum_\pi \left|\psi'_{\text{no},\pi}\right\rangle = 0$.
This proves the claim.
Now move on to proving that $\text{TD}(\rho', \rho_1)$ is negligible. First observe that ρ_1 is the sum of two parts $\rho_1 = \rho_{\text{yes}} + \rho_{\text{no}}$ corresponding to the situations, ρ_{yes} where c was measured to be 0 and ρ_{no} where c was measured to be 1. And in the same way decompose $\rho' = \rho_{\text{yes}} + \rho'_{\text{no}}$ by defining:

$$\rho_{\text{yes}} = \gamma_{\text{yes}}|\psi_{\text{yes}}\rangle\langle\psi_{\text{yes}}|\gamma^*_{\text{yes}}$$

and

$$\rho'_{\text{no}} = \left(\frac{1}{M!}\sum_\pi\sum_{x,y}\alpha_x\alpha^*_y|\psi_x\rangle\langle\psi_y| \otimes |\pi(x)\rangle\langle\pi(y)|\right) = \frac{1}{M!}\sum_\pi \left|\psi'_{\text{no},\pi}\right\rangle\left\langle\psi'_{\text{no},\pi}\right|$$

and

$$\rho_{\text{no}} = \left(\frac{1}{M!}\sum_\pi\sum_x |\alpha_x|^2|\psi_x\rangle\langle\psi_x| \otimes |\pi(x)\rangle\langle\pi(x)|\right)$$

Now compute

$$\begin{aligned}\rho' - \rho_1 &= \left(\rho_{\text{yes}} + \frac{1}{M!}\sum_\pi \left|\psi'_{\text{no},\pi}\right\rangle\left\langle\psi'_{\text{no},\pi}\right|\right) - (\rho_{\text{yes}} + \rho_{\text{no}})\\ &= \frac{1}{M!}\sum_\pi \left|\psi'_{\text{no},\pi}\right\rangle\left\langle\psi'_{\text{no},\pi}\right| - \rho_{\text{no}}\\ &= \frac{1}{M!}\sum_\pi\sum_{x\neq y}\alpha_x\alpha^*_y|\psi_x\rangle\langle\psi_y| \otimes |\pi(x)\rangle\langle\pi(y)|\\ &= \left(\sum_{x\neq y}\alpha_x\alpha^*_y|\psi_x\rangle\langle\psi_y|\right) \otimes \left(\frac{1}{M^2}\sum_{u\neq w}|u\rangle\langle w|\right)\end{aligned}$$

So call

$$\sigma_1 := \sum_{x \neq y} \alpha_x \alpha_y^* |\psi_x\rangle\langle\psi_y|$$

and

$$\sigma_2 := \frac{1}{M^2} \sum_{u \neq w} |u\rangle\langle w|$$

Then

$$\rho_0 - \rho_1 = \sigma_1 \otimes \sigma_2$$

Now prove that $\|\sigma_2\|_1$ is sufficiently small, for this sake let $\rho_* = \frac{1}{M} I_M$:

$$\begin{aligned}
\|\sigma_2\|_1 &= \left\| \frac{1}{M^2} \left(\sum_{u \neq w} |u\rangle\langle w| \right) \right\|_1 = \left\| \frac{1}{M^2} \left(\sum_{u,w} |u\rangle\langle w| - \sum_z |z\rangle\langle z| \right) \right\|_1 \\
&= \frac{1}{M} \left\| \sum_{u,w} \frac{1}{M} |u\rangle\langle w| - \sum_z \frac{1}{M} |z\rangle\langle z| \right\|_1 \overset{(i)}{\leq} \frac{1}{M} \left(\left\| \frac{1}{M} \sum_{u,w} |u\rangle\langle w| \right\|_1 + \left\| \frac{1}{M} \sum_z |z\rangle\langle z| \right\|_1 \right) \\
&= \frac{1}{M} \left(\left\| |+\rangle^{\otimes m} \langle +|^{\otimes m} \right\|_1 + \|\rho_*\|_1 \right) \overset{(ii)}{=} \frac{1}{M}(1+1) = 2 \cdot 2^{-m}
\end{aligned}$$

where (i) uses the triangle inequality for the trace norm, and (ii) involves the following two facts: for any normalized pure state $|\psi\rangle$, $\||\psi\rangle\langle\psi|\|_1 = 1$ (here in particular we have $|+\rangle^{\otimes m} = \frac{1}{\sqrt{M}} \sum_x |x\rangle$) and for the maximally mixed state $\rho_* := \frac{1}{M} I_M$, $\|\rho_*\|_1 = 1$. So it follows that:

$$\|\sigma_2\|_1 \leq 2^{-m+1}$$

and we can compute

$$\begin{aligned}
\|\sigma_1\|_1 &= \left\| \sum_{x,y} \alpha_x \alpha_y^* |\psi_x\rangle\langle\psi_y| - \sum_x |\alpha_x|^2 |\psi_x\rangle\langle\psi_x| \right\|_1 \\
&= \left\| \Big(\sum_x \alpha_x |\psi_x\rangle \Big) \Big(\sum_y \alpha_y^* \langle\psi_y| \Big) - \sum_x |\alpha_x|^2 |\psi_x\rangle\langle\psi_x| \right\|_1 \\
&\leq 1 + \left\| \sum_x |\alpha_x|^2 |\psi_x\rangle\langle\psi_x| \right\|_1 \\
&\leq 1 + \sum_x |\alpha_x|^2 \, \||\psi_x\rangle\langle\psi_x|\|_1 \\
&= 1 + \sum_x |\alpha_x|^2 \cdot 1 \\
&= 1 + \|\gamma_{\text{no}} |\psi_{\text{no}}\rangle\|^2 \\
&= 1 + |\gamma_{\text{no}}|^2 \leq 2
\end{aligned}$$

So all in all

$$\|\rho_0 - \rho_1\|_1 = \|\sigma_1\|_1 \cdot \|\sigma_2\|_1 \leq 2 \cdot 2^{-m+1} = 2^{-m+2}$$

so

$$\text{TD}(\rho_0, \rho_1) = \frac{1}{2} \|\rho_0 - \rho_1\|_1 \leq 2^{-m+1}$$

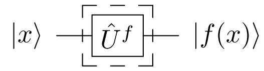
This implies that no adversary can distinguish the results of F_0 and F_1 with probability better than 2^{-m+1}. In particular if m is at least superlogarithmical, so for instance linear in the security parameter, then F_0 and F_1 are indistinguishable for one query. □

Lemma 3. *For numbers m and n and an injective function $f : \{0,1\}^m \to \{0,1\}^{m+n}$ let $\hat{U}^f$ be the isometry that performs the ER-type mapping $|x\rangle \mapsto |f(x)\rangle$. Let X be a quantum register containing m qubits. Then the following two oracles can be distinguished with probability at most $3 \cdot 2^{-n}$.*

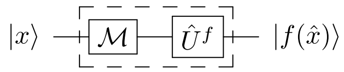
1. *F_0: Pick f uniformly at random and then apply $\hat{U}^f$ on X,*

2. F_1: *Pick f uniformly at random, measure X in the computational basis then apply $\hat{U}^f$ to the result.*

The quantum circuit for F_0 is:



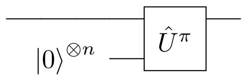
and for F_1 it is:



where $\mathcal{M}$ is a computational basis measurement (in the picture we denote the outcome of this measurement with $\hat{x}$).

Proof. Intuitively this follows from Lemma 2 because: Picking a random injection has the same distribution as composing concatenation of sufficiently many 0s with a random permutation. Formally, the equivalence is shown by a sequence of hybrid oracles where $G_0 = F_0$ and $G_4 = F_1$. In the definition of the hybrid games, π is always a random permutation $\pi : \{0,1\}^{m+n} \to \{0,1\}^{m+n}$.
G_0 is the same as F_0 and G_1 is the following oracle:

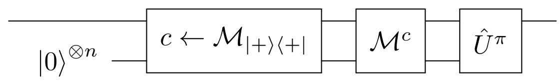
Oracle G_1 :



G_0 and G_1 are perfectly indistinguishable for any adversary, because the probability distributions of the observed functionality are exactly the same.

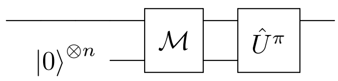
G_1 and G_2 can be distinguished with probability at most 2^{-m-n+1} by Lemma 2 where G_2 is the following oracle:

Oracle G_2 :



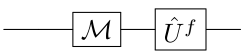
(Here we follow the same notation as above namely, that $c \leftarrow \mathcal{M}_{|+\rangle\langle+|}$ is a projective measurement, storing the result (0 or 1) in c, that projects to the spaces span$(|+\rangle^{\otimes m})$ (corresponding to 0) and its orthogonal space (corresponding to 1) and $\mathcal{M}^1$ is a measurement in the computational basis, whose outcome is denoted by $\hat{x}$ and $\mathcal{M}^0$ means no operation.)

Oracle G_3 :



G_2 and G_3 can be distinguished with probability at most 2^{-n} because the probability of measuring $|+\rangle$ is 2^{-n}. Or more formally because $(|\phi\rangle \otimes |0\rangle^{\otimes n})^\dagger |+\rangle \leq 2^{-\frac{n}{2}}$ for any $|\phi\rangle$.

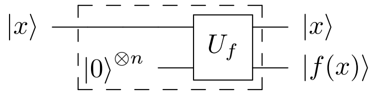
Oracle G_4 :



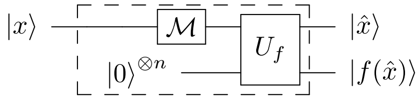
G_3 and G_4 are perfectly indistinguishable because the probability distributions are the same. And G_4 is the same as F_1. Thus F_0 and F_1 can be distinguished with probability at most $2^{-n} + 2^{-m-n+1} + 2^{-n}$ which is bounded by $3 \cdot 2^{-n}$ □

Lemma 4. *For a random function $f : \{0,1\}^m \to \{0,1\}^n$, an embedding query to f is indistinguishable from an embedding query to f preceded by a computational measurement on the input register. Let X be an m-qubit quantum register. Then for any input quantum register m, the following two oracles can be distinguished with probability at most 2^{-n}.*

1. F_0: *apply U_f to X and another register containing n zeros. The quantum circuit for F_0 is:*



2. F_1: *measure X in the computational basis and apply U_f to the result and another register containing zeros. The circuit for F_1 is:*



where $\mathcal{M}$ is a computational basis measurement whose outcome we denote by $\hat{x}$.

Proof. Let $M := 2^m$ and $N := 2^n$. A general strategy for distinguishing F_0 and F_1 can be described as follows: The adversary chooses some Hilbert space $\mathcal{H}_A$ and for each $x \in \{0,1\}^m$ picks $\alpha_x \in \mathbb{C}, |\phi_x\rangle \in \mathcal{H}_A$ such that $\sum_{x \in X} |\alpha_x|^2 = 1$. The adversary then prepares the bipartite state

$$|\Psi\rangle_{AM} := \sum_{x \in X} \alpha_x |\phi_x\rangle_A \otimes |x\rangle_M$$

and sends the B-part as the input of an oracle query to f. (We can assume this without loss of generality, because any state $|\Psi\rangle_{AB}$ can be written in this form.) Let ρ_0 be the density operator of the state after applying the oracle in F_0 to $|\Psi\rangle$. Let ρ_1 be the density operator of the state after applying the oracle in F_1 to $|\Psi\rangle$. Then holds

$$\rho_0 = \frac{1}{N^M} \sum_f \sum_{x,y} \alpha_x^* \alpha_y |\phi_x\rangle\langle\phi_y| \otimes |x\rangle\langle y| \otimes |f(x)\rangle\langle f(y)|$$

and

$$\begin{aligned}\rho_1 =& \frac{1}{N^M} \sum_f \sum_{x,y} \alpha_x^* \alpha_x |\phi_x\rangle\langle\phi_x| \otimes |x\rangle\langle x| \otimes |f(x)\rangle\langle f(x)| \\ =& \frac{1}{N^M} \sum_f \sum_{x,y} \delta_{x=y} \alpha_x^* \alpha_y |\phi_x\rangle\langle\phi_y| \otimes |x\rangle\langle y| \otimes |f(x)\rangle\langle f(y)|\end{aligned}$$

Compute

$$\begin{aligned}\rho_0 - \rho_1 =& \frac{1}{N^M} \sum_f \sum_{x,y} \delta_{x \neq y} \alpha_x^* \alpha_y |\phi_x\rangle\langle\phi_y| \otimes |x\rangle\langle y| \otimes |f(x)\rangle\langle f(y)| \\ =& \frac{1}{N^M} \sum_{x \neq y} \sum_{u,w} \sum_{f, f(x)=u, f(y)=w} \alpha_x^* \alpha_y |\phi_x\rangle\langle\phi_y| \otimes |x\rangle\langle y| \otimes |u\rangle\langle w| \\ =& \frac{1}{N^2} \big(\sum_{x \neq y} |x\rangle\langle y| \big) \otimes \big(\sum_{u,w} |u\rangle\langle w| \big) \\ =& \frac{1}{N} \big(\frac{1}{N} \sum_{x \neq y} |x\rangle\langle y| \big) \otimes \big(\frac{1}{N} \sum_{u,w} |u\rangle\langle w| \big) \\ =& \frac{1}{N} \big(|+\rangle\langle+| - \frac{1}{N} \mathrm{Id} \big) \otimes \big(|+\rangle\langle+| \big)\end{aligned}$$

where u and w run over $\{0,1\}^n$. This implies:

$$\|\rho_0 - \rho_1\|_1 = \| \frac{1}{N} \big(|+\rangle\langle+| - \frac{1}{N} \mathrm{Id} \big) \otimes \big(|+\rangle\langle+| \big) \|_1 = \frac{1}{N} \cdot 2 \cdot 1 = \frac{2}{N}$$

so

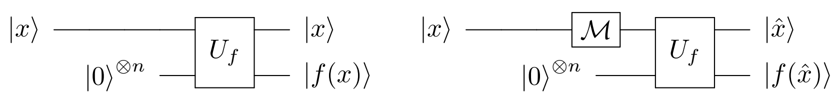
$$\mathrm{TD}(\rho_0, \rho_1) \leq \frac{1}{N} = 2^{-n}$$

and that's neglible hence ρ_0 and ρ_1 are indistinguishable. □

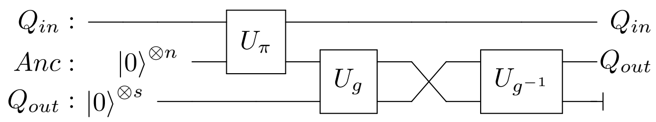
Corollary 1. *Assume $n \geq m$. For a random injective function $f : \{0,1\}^m \to \{0,1\}^n$ the oracles F_0 and F_1 in Lemma 4 are distinguishable with probability at most $1/2^n + C/2^n$ where C is a universal constant.*

Proof. This follows from Theorem 7 in [Zha15] that states any algorithm making q quantum queries cannot distinguish a random function from a random injective function, except with probability at most $Cq^3/2^n$. □

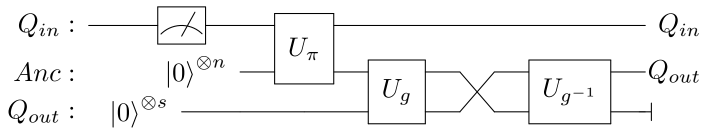
Corollary 2. *Let $R \subseteq \{0,1\}^s$ be a (fixed) set of size 2^n. Let $f : \{0,1\}^m \to \{0,1\}^s$ be a random injection with range R, that is, f is uniformly randomly chosen from the set of all injective functions $f : \{0,1\}^m \to \{0,1\}^s$ with $\operatorname{im} f \subseteq R$. An EM-query to f is distinguishable from an EM-query to f preceded with a computational basis measurement with probability at most $1/2^n + C/2^n$ where C is a universal constant. In other words, the following circuits are indistinguishable.*



Proof. We can write $f = g \circ \pi$ where $g : \{0,1\}^n \to \{0,1\}^s$ is a fixed injective function with range R and $\pi : \{0,1\}^m \to \{0,1\}^n$ is a random injective function. Let g^{-1} be a left inverse for the function g. An *EM* query to f can be implemented using functions g and π as follows (using an ancillary register *Anc*):



A simple calculation shows that the above circuit implements the isometry $U_f = U_{g\circ\pi}$. Now using Corollary 1, the circuit above is indistinguishable from the following circuit when one measures Q_{in} register at the beginning: (We stress that U_π is used only once as required by Corollary 1)



And this is a circuit that implements an *EM*-query to f preceded with a measurement. □

5 Impossible Security Notions

Proposition 1. *There is no* $\mathfrak{l}$-chall$(\mathfrak{c}_{\mathfrak{nb}}, ST, 1\text{ct})$*-IND-CPA-secure encryption scheme where the* $\mathfrak{l}$ *and* $\mathfrak{c}_{\mathfrak{nb}}$ *can be replaced by any of the possible parameters.*

Proof. This is formally proven in [BZ13b] as Theorem 4.2. For short the attack consists of inputting into the challenge query oracle the state

$$|0\rangle^{\otimes n} \otimes |\psi\rangle \otimes |0\rangle^{\otimes n'}$$

where $|\psi\rangle$ is some arbitrary "sufficiently non-classical" quantum state, for example $|+\rangle^{\otimes n}$. If $b = 0$ the state $|\psi\rangle$ is preserved and if $b = 1$ the state $|\psi\rangle$ is disturbed. So the adversary can distinguish by measuring the second register. □

Proposition 2. *There is no* $\mathfrak{l}$-chall$(\mathfrak{c}_{\mathfrak{nb}}, ST, 2\text{ct})$*-IND-CPA-secure encryption scheme where the* $\mathfrak{l}$ *and* $\mathfrak{c}_{\mathfrak{nb}}$ *can be replaced by any of the possible parameters.*

Proof. It is formally proven in [BZ13b] as Theorem 4.4. For short the attack consists of inputting into the challenge query oracle the state

$$|0\rangle^{\otimes n} \otimes |\psi\rangle \otimes |0\rangle^{\otimes n'} \otimes |+\rangle^{\otimes n'}$$

where $|\psi\rangle$ is some arbitrary "sufficiently non-classical" quantum state. If $b = 0$ the state $|\psi\rangle$ is preserved as its encryption is "absorbed" by $|+\rangle^{\otimes n'}$, but if $b = 1$ the state $|\psi\rangle$ is disturbed. So the adversary can distinguish by measuring the second register. □

Lemma 5. *A query with* $|0\rangle^{\otimes n'}$ *on the output register to an* ST*-oracle can be simulated by an* EM*-oracle.*

Proof. This follows immediately from the definition of ST- and EM-oracles. □

Proposition 3. *There is no* $\mathfrak{l}$-chall$(\mathfrak{c}_{\mathfrak{nb}}, EM, 1\text{ct})$*-IND-CPA-secure encryption scheme where the* $\mathfrak{l}$ *and* $\mathfrak{c}_{\mathfrak{nb}}$ *can be replaced by any of the possible parameters.*

Proof. The same proof as for Proposition 1 works as the attack is based on inputting $|0\rangle^{\otimes n'}$ on the output register, so Lemma 5 yields the result. More precisely, the adversary inputs $|0\rangle^{\otimes n} \otimes |\psi\rangle$ and gets exactly the same output as in the proof of Proposition 1 and then can do exactly the same measurement to distinguish. □

6 Implications

From the theoretically $(4+1)\times 2\times 4\times 3 = 120$ possible IND-CPA-notions, we excluded $1\times 1\times 4\times 3 = 12$ that correspond to IND-OT-CPA instead of IND-CPA, as there is no learning query and only 1 challenge. This leaves 108 notions. Next we excluded $2\times 2\times 3\times 3 = 36$ notations that we considered unreasonable, as they combine quantum learning queries with quantum challenge queries of different query models. This leaves 72 notions. Next we excluded 15 notions that are proven impossible. This leaves 57 notions.

Now we will relate the remaining IND-CPA-notions. The 57 notions can be grouped together in 14 Panels depicted in Figure 1, so that in each panel the notions are equivalent. In order to have a compact representation in Figure 1, for any $\mathfrak{qm} \in \{ST, EM, ER\}$ we define the set $T^*(\mathfrak{qm})$ as

$$T^*(\mathfrak{qm}) = \{(\text{learn}(0,-), *, \mathfrak{qm}), (\text{learn}(*, CL), *, \mathfrak{qm}), (\text{learn}(*, \mathfrak{qm}), 1, \mathfrak{qm}), (\text{learn}(*, \mathfrak{qm}), *, \mathfrak{qm})\}.$$

Note that $(\text{learn}(*, CL), 1, \mathfrak{qm})$ is not in $T^*(\mathfrak{qm})$. This set will only be used in Figure 1 to have a compact representation.

Inside each panel all the notions are equivalent and apart from that, there are the following 20 implications between the panels depicted in Figure 1 using black arrows. The full set of implications between all notions can be derived by taking the transitive closure of this graph. Every implication that is not in the transitive closure of the graph is being disproven in the section about separations section 7 or some of them have been stated as conjectures. The red dashed red arrows in Figure 1 show non-implications that have left as conjectures.

Note that Panel 6 corresponds to the quantum security definitions by Boneh and Zhandry [BZ13b]. Some implications follow from some theorem proven later and some are easy enough that say can be proven by a short argument. The arguments used are the following. In each case, we assign a short name in bold to that argument type.

- **more cqs:** i.e. more challenge queries. If two security notions just differ by the fact that one of them allows only one challenge query and the other allows polynomially many, then trivially the notion allowing polynomially many implies the notion allowing only one. For example:

$$\text{learn}(*, CL)\text{-chall}(*, ER, \text{ror}) \Rightarrow \text{learn}(*, CL)\text{-chall}(1, ER, \text{ror})$$

- **extra lq-oracle:** i.e. extra learning-query-oracle. If two security notions just differ by the fact, that one of them allows learning queries and the other doesn't, then trivially the notion allowing learning queries implies the notion allowing no learning queries. For example:

$$\text{learn}(*, CL)\text{-chall}(*, ER, \text{1ct}) \Rightarrow \text{learn}(0, -)\text{-chall}(*, ER, \text{1ct})$$

- **other ciphertext:** If two security just differ by the fact, that one of them allows $\text{chall}(\mathfrak{c}_{nb}, ER, \text{1ct})$ challenge queries and the other $\text{chall}(\mathfrak{c}_{nb}, ER, \text{2ct})$ challenge queries, then trivially the notions allowing $\text{chall}(\mathfrak{c}_{nb}, ER, \text{2ct})$ challenge queries implies the notion allowing $\text{chall}(\mathfrak{c}_{nb}, ER, \text{1ct})$ challenge queries (see subsubsection 3.2.6). For example:

$$\text{learn}(*, CL)\text{-chall}(1, ER, \text{2ct}) \Rightarrow \text{learn}(*, CL)\text{-chall}(1, ER, \text{1ct})$$

- **simulate classical:** Classical queries can be simulated with any quantum query type by measuring the result in the computational basis. For example:

$$\text{learn}(*, ER)\text{-chall}(*, ER, \text{ror}) \Rightarrow \text{learn}(*, CL)\text{-chall}(*, ER, \text{ror})$$

- **simulate le with ch:** When learning queries are classical, they can be simulated by the challenge queries in the case of 1ct and 2ct. In more details, on input m as a classical leaning query, we can query (m, m) as a challenge query and simulate the learning query. For instance:

$$\text{learn}(0, -)\text{-chall}(*, ER, \text{2ct}) \implies \text{learn}(*, CL)\text{-chall}(*, ER, \text{2ct})$$

- **EM simulation by ST**. The query type EM can be simulated by ST-type by putting $|0\rangle$ in the output register Q_{out}. For example,

$$\text{learn}(*, CL)\text{-chall}(*, ST, \text{ror}) \Rightarrow \text{learn}(*, CL)\text{-chall}(*, EM, \text{ror})$$

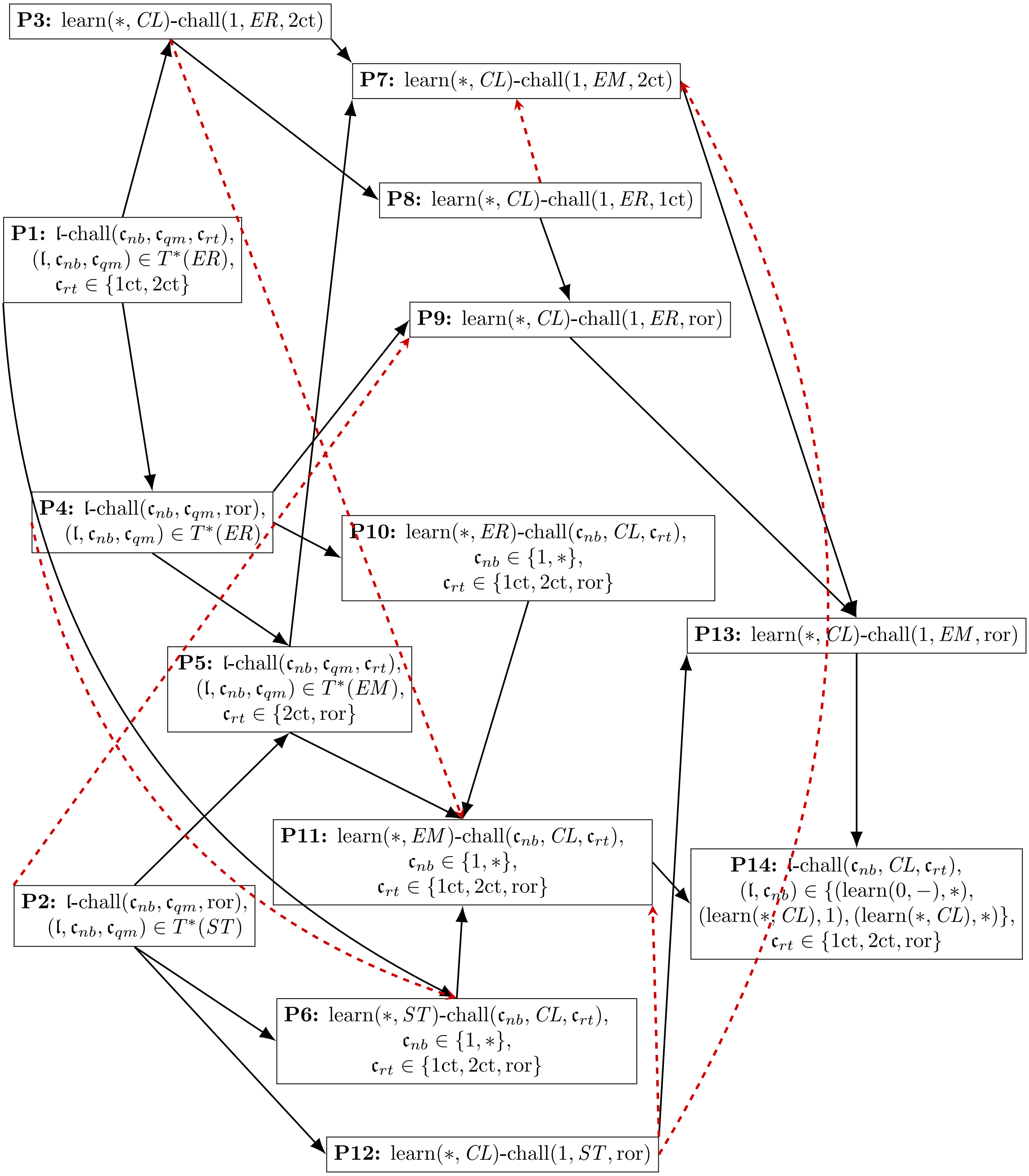
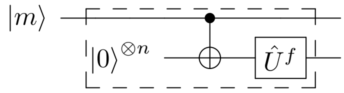


Figure 1: The 57 notions and equivalences and implications between them. The red dashed arrows show non-implications that have left as conjectures.

- **EM simulation by ER.** The query type EM can be simulated by ER-type queries. In the following, we present a circuit that depicts the simulation of EM-type queries to some function f using an ER-type query to f:



For example,

$$\text{learn}(*, ER)\text{-chall}(*, ER, \text{ror}) \Rightarrow \text{learn}(*, EM)\text{-chall}(*, EM, \text{ror})$$

For the panels with more than one notion, it has to be proven, that all the notations inside are equivalent:

Panel P1 (8 security notions):

learn($*$, CL)-chall($*$, ER, 2ct)	$\Longrightarrow$	learn($*$, CL)-chall($*$, ER, 1ct)	other ciphertext
learn($*$, CL)-chall($*$, ER, 1ct)	$\Longrightarrow$	learn(0, $-$)-chall($*$, ER, 1ct)	extra lq-oracle
learn(0, $-$)-chall($*$, ER, 1ct)	$\Longrightarrow$	learn($*$, ER)-chall($*$, ER, 1ct)	by Theorem 3
learn($*$, ER)-chall($*$, ER, 1ct)	$\Longrightarrow$	learn($*$, ER)-chall(1, ER, 1ct)	more cqs
learn($*$, ER)-chall(1, ER, 1ct)	$\Longrightarrow$	learn($*$, ER)-chall(1, ER, 2ct)	by Theorem 6
learn($*$, ER)-chall(1, ER, 2ct)	$\Longrightarrow$	learn($*$, ER)-chall($*$, ER, 2ct)	by Theorem 1
learn($*$, ER)-chall($*$, ER, 2ct)	$\Longrightarrow$	learn(0, $-$)-chall($*$, ER, 2ct)	extra lq-oracle
learn(0, $-$)-chall($*$, ER, 2ct)	$\Longrightarrow$	learn($*$, CL)-chall($*$, ER, 2ct)	simulate le with ch

Panel P2 (4 security notions):

learn($*$, ST)-chall($*$, ST, ror)	$\Longrightarrow$	learn($*$, CL)-chall($*$, ST, ror)	simulate classical
learn($*$, CL)-chall($*$, ST, ror)	$\Longrightarrow$	learn(0, $-$)-chall($*$, ST, ror)	extra lq-oracle
learn(0, $-$)-chall($*$, ST, ror)	$\Longrightarrow$	learn($*$, ST)-chall($*$, ST, ror)	by Theorem 4
learn($*$, ST)-chall($*$, ST, ror)	$\Longrightarrow$	learn($*$, ST)-chall(1, ST, ror)	more cqs
learn($*$, ST)-chall(1, ST, ror)	$\Longrightarrow$	learn($*$, ST)-chall($*$, ST, ror)	by Theorem 1

Panel P4 (4 security notions):

learn($*$, ER)-chall($*$, ER, ror)	$\Longrightarrow$	learn($*$, CL)-chall($*$, ER, ror)	simulate classical
learn($*$, CL)-chall($*$, ER, ror)	$\Longrightarrow$	learn(0, $-$)-chall($*$, ER, ror)	extra lq-oracle
learn(0, $-$)-chall($*$, ER, ror)	$\Longrightarrow$	learn($*$, ER)-chall($*$, ER, ror)	by Theorem 4
learn($*$, ER)-chall($*$, ER, ror)	$\Longrightarrow$	learn($*$, ER)-chall(1, ER, ror)	more cqs
learn($*$, ER)-chall(1, ER, ror)	$\Longrightarrow$	learn($*$, ER)-chall($*$, ER, ror)	by Theorem 1

Panel P5 (8 security notions):

learn($*$, EM)-chall($*$, EM, ror)	$\Longrightarrow$	learn($*$, CL)-chall($*$, EM, ror)	simulate classical
learn($*$, CL)-chall($*$, EM, ror)	$\Longrightarrow$	learn(0, $-$)-chall($*$, EM, ror)	extra lq-oracle
learn(0, $-$)-chall($*$, EM, ror)	$\Longrightarrow$	learn($*$, EM)-chall($*$, EM, ror)	by Theorem 4
learn($*$, EM)-chall($*$, EM, ror)	$\Longrightarrow$	learn($*$, EM)-chall(1, EM, ror)	more cqs
learn($*$, EM)-chall(1, EM, ror)	$\Longrightarrow$	learn($*$, EM)-chall(1, EM, 2ct)	by Theorem 1
learn($*$, EM)-chall(1, EM, 2ct)	$\Longrightarrow$	learn($*$, EM)-chall($*$, EM, 2ct)	by Theorem 1
learn($*$, EM)-chall($*$, EM, 2ct)	$\Longrightarrow$	learn($*$, CL)-chall($*$, EM, 2ct)	simulate classical
learn($*$, CL)-chall($*$, EM, 2ct)	$\Longrightarrow$	learn(0, $-$)-chall($*$, EM, 2ct)	extra lq-oracle
learn(0, $-$)-chall($*$, EM, 2ct)	$\Longrightarrow$	learn($*$, EM)-chall($*$, EM, 2ct)	by Theorem 3
learn($*$, EM)-chall($*$, EM, 2ct)	$\Longrightarrow$	learn($*$, EM)-chall($*$, EM, ror)	by Theorem 8

Panel P6 (6 security notions):

learn($*$, ST)-chall(1, CL, 1ct)	$\Longrightarrow$	learn($*$, ST)-chall($*$, CL, 1ct)	by Theorem 1
learn($*$, ST)-chall($*$, CL, 1ct)	$\Longrightarrow$	learn($*$, ST)-chall(1, CL, 1ct)	more cqs
The rest of equivalences			by Theorem 2

Panel P10 (6 security notions):

learn$(*, ER)$-chall$(1, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, ER)$-chall$(*, CL, 1\text{ct})$	by Theorem 1
learn$(*, ER)$-chall$(*, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, ER)$-chall$(1, CL, 1\text{ct})$	more cqs
The rest of equivalences			by Theorem 2

Panel P11 (6 security notions):

learn$(*, EM)$-chall$(1, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, EM)$-chall$(*, CL, 1\text{ct})$	by Theorem 1
learn$(*, EM)$-chall$(*, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, EM)$-chall$(1, CL, 1\text{ct})$	more cqs
The rest of equivalences			by Theorem 2

Panel P14 (9 security notions):

learn$(*, CL)$-chall$(1, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, CL)$-chall$(*, CL, 1\text{ct})$	by Theorem 1
learn$(*, CL)$-chall$(*, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, CL)$-chall$(1, CL, 1\text{ct})$	more cqs
learn$(*, CL)$-chall$(*, CL, 1\text{ct})$	$\Longrightarrow$	learn$(0, -)$-chall$(*, CL, 1\text{ct})$	extra lq-oracle
learn$(0, -)$-chall$(*, CL, 1\text{ct})$	$\Longrightarrow$	learn$(*, CL)$-chall$(*, CL, 1\text{ct})$	simulate le with ch
The rest of equivalences			by Theorem 2

The 20 arrows in detail:

- From panel 1 to panel 3
  precisely: learn$(*, CL)$-chall$(*, ER, 2\text{ct}) \Longrightarrow$ learn$(*, CL)$-chall$(1, ER, 2\text{ct})$
  argument: more cqs
- From panel 1 to panel 4
  precisely: learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \Longrightarrow$ learn$(*, ER)$-chall$(*, ER, \text{ror})$
  argument: Theorem 7
- From panel 1 to panel 6
  precisely: learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \Longrightarrow$ learn$(*, ST)$-chall$(*, CL, 1\text{ct})$
  argument: Theorem 9
- From panel 2 to panel 5
  precisely: learn$(*, ST)$-chall$(*, ST, \text{ror}) \Longrightarrow$ learn$(*, EM)$-chall$(*, EM, 2\text{ct})$
  argument: EM simulation by ST
- From panel 2 to panel 6
  precisely: learn$(*, ST)$-chall$(*, ST, \text{ror}) \Longrightarrow$ learn$(*, ST)$-chall$(*, CL, 1\text{ct})$
  argument: simulate classical
- From panel 4 to panel 5
  precisely: learn$(*, ER)$-chall$(*, ER, \text{ror}) \Longrightarrow$ learn$(*, EM)$-chall$(*, EM, \text{ror})$
  argument: *EM* simulation by *ER*
- From panel 2 to panel 12
  precisely: learn$(*, CL)$-chall$(*, ST, \text{ror}) \Longrightarrow$ learn$(*, CL)$-chall$(1, ST, \text{ror})$
  argument: more cqs
- From panel 3 to panel 7
  precisely: learn$(*, CL)$-chall$(*, ER, 2\text{ct}) \Longrightarrow$ learn$(*, CL)$-chall$(*, EM, 2\text{ct})$
  argument: EM simulation by ER.
- From panel 3 to panel 8
  precisely: learn$(*, CL)$-chall$(1, ER, 2\text{ct}) \Longrightarrow$ learn$(*, CL)$-chall$(1, ER, 1\text{ct})$
  argument: other ciphertext
- From panel 4 to panel 10
  precisely: learn$(*, ER)$-chall$(*, ER, \text{ror}) \Longrightarrow$ learn$(*, ER)$-chall$(*, CL, 1\text{ct})$
  argument: simulate classical

- From panel 4 to panel 9
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(*, \mathit{ER}, \text{ror}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{ER}, \text{ror})$
  argument: more cqs

- From panel 5 to panel 7
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(*, \mathit{EM}, \text{2ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{2ct})$
  argument: more cqs

- From panel 5 to panel 11
  precisely: $\text{learn}(*, \mathit{EM})\text{-chall}(*, \mathit{EM}, \text{2ct}) \implies \text{learn}(*, \mathit{EM})\text{-chall}(*, \mathit{CL}, \text{1ct})$
  argument: simulate classical

- From panel 6 to panel 11
  precisely: $\text{learn}(*, \mathit{ST})\text{-chall}(1, \mathit{CL}, \text{1ct}) \implies \text{learn}(*, \mathit{EM})\text{-chall}(1, \mathit{CL}, \text{1ct})$
  argument: EM simulation by ST

- From panel 8 to panel 9
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{ER}, \text{1ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{ER}, \text{1ct})$
  argument: Theorem 7

- From panel 10 to panel 11
  precisely: $\text{learn}(*, \mathit{ER})\text{-chall}(1, \mathit{CL}, \text{1ct}) \implies \text{learn}(*, \mathit{EM})\text{-chall}(1, \mathit{CL}, \text{1ct})$
  argument: EM simulation by ER

- From panel 7 to panel 13
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{2ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{ror})$
  argument: Theorem 8

- From panel 9 to panel 13
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{ER}, \text{1ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{ror})$
  argument: EM simulation by ER

- From panel 11 to panel 14
  precisely: $\text{learn}(*, \mathit{EM})\text{-chall}(1, \mathit{CL}, \text{1ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{CL}, \text{1ct})$
  argument: simulate classical

- From panel 12 to panel 13
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{ST}, \text{1ct}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{ror})$
  argument: EM simulation by ST.

- From panel 13 to panel 14
  precisely: $\text{learn}(*, \mathit{CL})\text{-chall}(1, \mathit{EM}, \text{ror}) \implies \text{learn}(*, \mathit{CL})\text{-chall}(*, \mathit{CL}, \text{1ct})$
  arguments: We can show the implication with the application of the following arguments respectively: simulate classical, Theorem 2 and Theorem 1

These are the implications. Now we prove the theorem mentioned in this list.

Theorem 1. *If a* $\text{chall}(1, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$*-challenge-query can be efficiently simulated with an* $\mathfrak{l}_{\text{qm}}$*-learning-query (when knowing the challenge bit b) then* $\text{learn}(*, \mathfrak{l}_{\text{qm}})\text{-chall}(1, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}}) \implies \text{learn}(*, \mathfrak{l}_{\text{qm}})\text{-chall}(*, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$.

Proof. Let $\mathcal{A}$ be an adversary that wins in the $\text{learn}(*, \mathfrak{l}_{\text{qm}})\text{-chall}(*, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$ game with non-negligible advantage $\epsilon(n)$. We assume that $\mathcal{A}$ makes q challenge queries. We construct an adversary $\mathcal{B}$ that attacks in the sense of $\text{learn}(*, \mathfrak{l}_{\text{qm}})\text{-chall}(1, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$. Let $\mathcal{B}$ be an adversary that chooses uniformly at random an element k from $\{0, \dots, q+1\}$, runs the adversary $\mathcal{A}$ and answers to the i-th challenge query made by $\mathcal{A}$ as follows:

1. When $i < k$, $\mathcal{B}$ simulates the i-th challenge query by a learning query assuming that $b = 0$.

2. For k-th challenge query, $\mathcal{B}$ uses a challenge query to answer.

3. When $i > k$, $\mathcal{B}$ simulates the i-th challenge query by a learning query assuming that $b = 1$.

At the end, $\mathcal{B}$ returns $\mathcal{A}$'s output. We define $q+2$ hybrid games $\mathcal{G}_k$ corresponding to the possible choice k of $\mathcal{B}$. So $\mathcal{G}_j$ has the same description as above when the random choice of $\mathcal{B}$ is j. Note that $|\Pr[1 \leftarrow \mathcal{G}_0] - \Pr[1 \leftarrow \mathcal{G}_{q+1}]| \geq \epsilon(n)$ and therefore there exists $\alpha \in \{0, \dots, q\}$ such that $|\Pr[1 \leftarrow \mathcal{G}_\alpha] - \Pr[1 \leftarrow \mathcal{G}_{\alpha+1}]| \geq \frac{\epsilon(n)}{q+1}$. Since $k = \alpha$ with probability $\frac{1}{q+2}$, $\mathcal{B}$ can be a distinguisher for games $\mathcal{G}_\alpha$ and $\mathcal{G}_{\alpha+1}$ with non-negligible probability. This is a contradiction with the security in the learn$(*, \mathfrak{l}_{\text{qm}})$-chall$(1, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$ sense. □

Theorem 2. *Let* $\mathfrak{L} = \{\text{learn}(0,-), \text{learn}(*, CL), \text{learn}(*, ST), \text{learn}(*, EM), \text{learn}(*, ER)\}$ *and* $\mathfrak{C}_{nb} = \{1, *\}$. *For all* $(\mathfrak{l}, \mathfrak{C}_{nb}) \in \mathfrak{L} \times \mathfrak{C}_{nb} \setminus \{(\text{learn}(0,-), 1)\}$, *the following security notions are equivalent for all encryption schemes: (Note that when* $\mathfrak{l} = \text{learn}(0,-)$ *and* $\mathfrak{c}_{nb} = 1$, *the security definition is IND-OT-CPA that we have excluded.)*

- $\mathcal{C}_{\text{1ct}} := \mathfrak{l}\text{-chall}(\mathfrak{c}_{nb}, CL, \text{1ct})$*-IND-CPA-security*
- $\mathcal{C}_{\text{2ct}} := \mathfrak{l}\text{-chall}(\mathfrak{c}_{nb}, CL, \text{2ct})$*-IND-CPA-security*
- $\mathcal{C}_{\text{ror}} := \mathfrak{l}\text{-chall}(\mathfrak{c}_{nb}, CL, \text{ror})$*-IND-CPA-security*

Proof. $\mathcal{C}_{\text{2ct}} \implies \mathcal{C}_{\text{1ct}}$: trivial.
$\mathcal{C}_{\text{1ct}} \implies \mathcal{C}_{\text{2ct}}$, case $\mathfrak{c}_{nb} = *$: A 2ct-challenge-query of the form

$$(m_0, m_1) \mapsto (\text{Enc}_k(m_b), \text{Enc}_k(m_{\bar{b}}))$$

can be simulated by two queries of the form $(m_0, m_1) \mapsto \text{Enc}_k(m_b)$, namely by querying

$$(m_0, m_1) \mapsto \text{Enc}_k(m_b)$$

to get $\text{Enc}_k(m_b)$ and then switching the inputs and querying

$$(m_1, m_0) \mapsto \text{Enc}_k(m_{\bar{b}})$$

to get $\text{Enc}_k(m_{\bar{b}})$. So the desired outcome $(\text{Enc}_k(m_b), \text{Enc}_k(m_{\bar{b}}))$ is simulated.
$\mathcal{C}_{\text{1ct}} \implies \mathcal{C}_{\text{2ct}}$ case $\mathfrak{c}_{nb} = 1$: We prove that

$$\mathfrak{l}\text{-chall}(1, CL, \text{1ct}) \implies \mathfrak{l}\text{-chall}(*, CL, \text{1ct}) \implies \mathfrak{l}\text{-chall}(*, CL, \text{2ct}) \implies \mathfrak{l}\text{-chall}(1, CL, \text{2ct})$$

(for simplicity we drop the IND-CPA-security from the notation above). The first implication follows from Theorem 1, the second implication was proven above and the third implication is trivial, because the only difference is that there are less challenge queries available on its right side.
$\mathcal{C}_{\text{1ct}} \implies \mathcal{C}_{\text{ror}}$: This follows from the fact that a ror-challenge-query can be simulated by a 1ct-challenge-query as follows. Let $\mathcal{A}$ be a successful adversary against $\mathfrak{l}\text{-chall}(\mathfrak{c}_{nb}, CL, \text{ror})$-IND-CPA-security, transform it into an adversary $\mathcal{B}^{\mathcal{A}}$ against $\mathfrak{l}\text{-chall}(\mathfrak{c}_{nb}, CL, \text{1ct})$-IND-CPA-security. (The adversary $\mathcal{B}$ runs $\mathcal{A}$ and plays the role of the challenger for $\mathcal{A}$.) The learning queries are simply forwarded. When $\mathcal{A}$ performs a challenge query with input m', then $\mathcal{B}$ samples a random value r and submits $(m_0, m_1) = (m', r)$ to the challenger. The challenger answers with $\text{Enc}_k(m_b)$ i.e. with $\text{Enc}_k(m')$ if $b = 0$ and with $\text{Enc}_k(r)$ if $b = 1$. This is exactly what $\mathcal{A}$ expects to get back, so $\mathcal{B}$ can simply pass it over to $\mathcal{A}$.
$\mathcal{C}_{\text{ror}} \implies \mathcal{C}_{\text{1ct}}$: We want to show that the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(m_0)$ is indistinguishable from the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(m_1)$. But since Enc is $\mathcal{C}_{\text{ror}}$-secure it follows that the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(m_0)$ is indistinguishable from the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(r)$ where r is random. And as well that the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(r)$ where r is random is indistinguishable from the game with challenge queries $(m_0, m_1) \mapsto \text{Enc}_k(m_1)$. So by transitivity of indistinguishability Enc is $\mathcal{C}_{\text{1ct}}$-secure. □

In the theorem below, we show that the security definition with no learning queries imply the security definition that performs *EM* and *ER* type learning queries. The idea of proof is to simulate learning queries with the challenge queries. Classically, we can simulate easily the learning queries using the challenge queries by making a copy of the message sent as a learning query and send two copy of messages as a challenge query. However, this approach is not straightforward in the quantum case because making a copy of a quantum message required an entanglement. This makes the output registers entangled in the $\text{chall}(*, EM, \text{2ct}), \text{chall}(*, ER, \text{2ct}), \text{chall}(*, ER, \text{1ct})$ type queries and discarding extra registers effects the other registers. Therefore, we define two intermediate games with learning queries that always return encryption of 0. Overall, we show that IND-CPA games and two intermediate games are indistinguishable.

Theorem 3. learn$(0,-)$-$\mathfrak{c}$ $\implies$ learn$(*,\mathfrak{l}_{\mathrm{qm}})$-$\mathfrak{c}$ *where* $\mathfrak{c} \in \{\text{chall}(*, EM, 2\text{ct}), \text{chall}(*, ER, 2\text{ct}), \text{chall}(*, ER, 1\text{ct})\}$ *and* $\mathfrak{l}_{\mathrm{qm}} \in \{EM, ER\}$.

Proof. Let Enc be some encryption scheme that is learn$(0,-)$-$\mathfrak{c}$-secure for $\mathfrak{c} \in \{\text{chall}(*, EM, 2\text{ct}), \text{chall}(*, ER, 2\text{ct}), \text{chall}(*, ER, 1\text{ct})\}$. We will show that Enc is learn$(*,\mathfrak{l}_{\mathrm{qm}})$-$\mathfrak{c}$-secure by defining a sequence of IND-CPA games that demonstrate that settings with challenge bit $b = 0$ and $b = 1$ are indistinguishable.

Define the learning query $\mathfrak{l}'$ to be as follows: For EM type learning queries, after receiving the quantum register Q_{in}, measure it in the computational basis to get a classical value x, compute Enc(0), and return $|x, \text{Enc}(0)\rangle$. For ER type learning queries, it returns $|\text{Enc}(0)\rangle$.

Let Game G_b be the IND-CPA game with $\mathfrak{c}$-challenge-queries and learn$(*,\mathfrak{l}_{\mathrm{qm}})$-learning-queries when the challenge bit is b. Let Game G'_b be the IND-CPA game with $\mathfrak{c}$-challenge-queries and $\mathfrak{l}'$-learning-queries when the challenge bit is b.

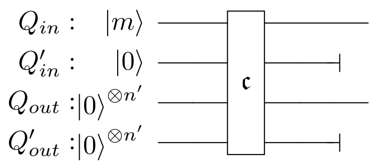
Now we shall show in sequence that these games are indistinguishable from one another:

$$G_0 \cong G'_1 \cong G'_0 \cong G_1.$$

To do this, we construct an adversary $\mathcal{B}$ that breaks learn$(0,-)$-$\mathfrak{c}$-security from an adversary $\mathcal{A}$ that distinguishes the two subsequent games in the relation above. Let b' denote the challenge bit of the adversary $\mathcal{B}$'s challenger. In all the cases, the adversary B answers the challenge queries made by $\mathcal{A}$ by forwarding them to its challenger. In the following, we show how the adversary $\mathcal{B}$ answers the learning queries made by $\mathcal{A}$ in each case.

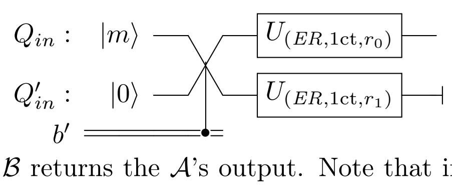
$\boldsymbol{G_0 \cong G'_1}$ **:** Upon receiving the quantum register Q_{in} as a learning query from the adversary $\mathcal{A}$, the adversary $\mathcal{B}$ prepares the quantum register Q'_{in} containing $|0\rangle$, performs the $\mathfrak{c}$-challenge query for Q_{in}, Q'_{in} registers and then does the following:

(i) When $\mathfrak{c} = \text{chall}(*, EM, 2\text{ct})$, $\mathcal{B}$ receives back four registers. $\mathcal{B}$ measures and discards the second and fourth registers and sends the first and third registers to $\mathcal{A}$.



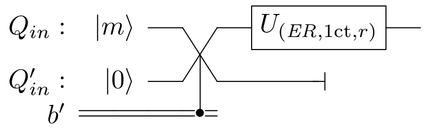
At the end, the adversary $\mathcal{B}$ returns the $\mathcal{A}$'s output. Note that if the challenge bit is $b' = 0$ then the adversary $\mathcal{B}$ returns $|m, \text{Enc}(m)\rangle$ to $\mathcal{A}$. This is a simulation of the EM type learning queries in game G_0. It is clear that the challenge queries made by $\mathcal{A}$ are simulated perfectly by $\mathcal{B}$. Therefore, the adversary $\mathcal{B}$ perfectly simulates game G_0 when $b' = 0$. When the challenge bit is $b' = 1$, the adversary $\mathcal{B}$ effectively measures Q_{in} by measuring Q'_{out} (which contains the encryption if Q_{in}) Thus, it returns $|m, \text{Enc}(0)\rangle$ (where m is the result of measuring Q_{in}) as an answer for a learning query. This is a simulation of the $\mathfrak{l}'$ learning queries in game G'_1. Therefore, the adversary $\mathcal{B}$ perfectly simulates game G'_1 when $b' = 1$. Since Enc is learn$(0,-)$-chall$(*, EM, 2\text{ct})$-secure, G_0 and G'_1 are indistinguishable.

(ii) When $\mathfrak{c} = \text{chall}(*, ER, 2\text{ct})$, $\mathcal{B}$ receives two registers. $\mathcal{B}$ measures and discards the second register and sends the first register to $\mathcal{A}$.



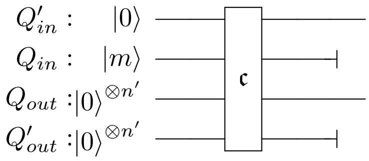
At the end, the adversary $\mathcal{B}$ returns the $\mathcal{A}$'s output. Note that if the challenge bit is $b' = 0$ then the adversary $\mathcal{B}$ returns $|\text{Enc}(m)\rangle$ to $\mathcal{A}$. This is a simulation of ER type learning queries in game G_0. It is clear that the challenge queries made by $\mathcal{A}$ are simulated perfectly by $\mathcal{B}$. Therefore, the adversary $\mathcal{B}$ perfectly simulates game G_0 when $b' = 0$. When the challenge bit is $b' = 1$ the adversary $\mathcal{B}$ returns $|\text{Enc}(0)\rangle$ as an answer for a learning query. This is a simulation of $\mathfrak{l}'$ learning queries in game G'_1. Therefore, the adversary $\mathcal{B}$ perfectly simulates game G'_1 when $b' = 1$. Since Enc is learn$(0,-)$-chall$(*, ER, 2\text{ct})$-secure, G_0 and G'_1 are indistinguishable.

(iii) When $\mathfrak{c} = \text{chall}(*, ER, 1\text{ct})$, $\mathcal{B}$ receives back one register and forwards it to $\mathcal{A}$.



At the end, the adversary $\mathcal{B}$ returns the $\mathcal{A}$'s output. Similar to the cases above, we can show that the adversary $\mathcal{B}$ simulates the game G_0 when the challenge bit is $b' = 0$ and it simulates the game G'_1 when the challenge bit is $b' = 1$. Since Enc is learn$(0, -)$-chall$(*, ER, 1\text{ct})$-secure, G_0 and G'_1 are indistinguishable.

$\boldsymbol{G'_0 \cong G_1}$: Similar to the cases above, we can show that G'_0 and G_1 are indistinguishable. In this case, the adversary $\mathcal{B}$ after receiving the quantum register Q_{in} as a learning query from the adversary $\mathcal{A}$, prepares the quantum register Q'_{in} containing $|0\rangle$, performs the $\mathfrak{c}$-challenge query for Q'_{in}, Q_{in} registers (the order of registers have been exchanged). Then it does exactly the same as above in each case. For instance in the case of $\mathfrak{c} = \text{chall}(*, EM, 2\text{ct})$, $\mathcal{B}$ receives back four registers, then measures and discards the second and fourth registers and sends the first and third registers to $\mathcal{A}$.



At the end, $\mathcal{B}$ returns the $\mathcal{A}$'s output. The other cases are similar.

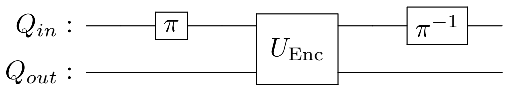
$\boldsymbol{G'_1 \cong G'_0}$ **:** It is clear that $\mathcal{B}$ can simulate $\mathfrak{l}'$ learning queries in both cases of EM and ER type queries by performing a $\mathfrak{c}$-challenge-query with input $|0\rangle \otimes |0\rangle$ to obtain Enc(0). Therefore, $\mathcal{B}$ can simulate the games G'_0 and G'_1 when $b' = 0$ and $b' = 1$, respectively. At the end, $\mathcal{B}$ returns the $\mathcal{A}$'s output. Two games are indistinguishable because Enc is learn$(0, -)$-$\mathfrak{c}$ secure. In summary, we showed that G_0 and G_1 are indistinguishable and therefore Enc is learn$(*, \mathfrak{l}_{\text{qm}})$-$\mathfrak{c}$-secure.

□

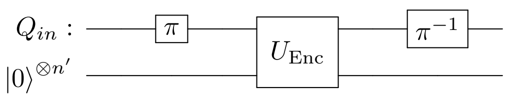
Theorem 4. learn$(0, -)$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror}) \implies$ learn$(*, \mathfrak{c}_{\text{qm}})$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$, $\quad \mathfrak{c}_{\text{qm}} \in \{ST, EM, ER\}$.

Proof. Let Enc be some encryption scheme that is learn$(0, -)$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$-secure for $\mathfrak{c}_{\text{qm}} \in \{ST, EM, ER\}$. We will show that Enc is learn$(*, \mathfrak{c}_{\text{qm}})$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$-secure by defining a sequence of IND-CPA games that demonstrate that the settings with the challenge bit $b = 0$ and $b = 1$ are indistinguishable. Let Game G_b be the IND-CPA game with chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$-challenge queries and learn$(*, \mathfrak{c}_{\text{qm}})$-learning queries when the challenge bit is b.

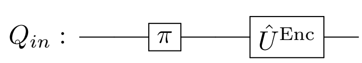
We define the game G' to be the IND-CPA game with chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$-challenge queries with the challenge bit $b = 1$ and learn$(*, \mathfrak{l}'_{\text{qm}})$-learning queries where the learning query model $\mathfrak{l}'_{\text{qm}}$ is as follows: For the query model qm $= ST$, after receiving the quantum registers Q_{in} and Q_{out}, apply a random permutation π on register Q_{in}, perform the query to Enc and finally apply π^{-1} on register Q_{in} afterwards. We draw the circuit below.



For the query model qm $= EM$, after receiving the quantum register Q_{in}, prepare a quantum register Q_{out} containing $|0\rangle^{\otimes n'}$, apply a random permutation π on register Q_{in}, perform the query to Enc and finally apply π^{-1} on register Q_{in} afterwards. We draw the circuit below.



For the query models ER, after receiving the quantum register Q_{in}, apply a random permutation π on register Q_{in}, perform the query to Enc (in the ER query model). The circuit for $\mathfrak{l}'_{\text{qm}}$ queries in this case is



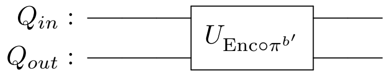
Next we will show the following indistinguishability relations.

$$G_0 \cong G' \cong G_1$$

In all cases, from an adversary that distinguishes two games we construct an adversary that breaks the learn$(0, -)$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$ security of Enc. Let $\mathcal{A}$ be an adversary that distinguishes two subsequent games in the relation above with non-negligible probability μ. We construct the adversary $\mathcal{B}$ that breaks the learn$(0, -)$-chall$(*, \mathfrak{c}_{\text{qm}}, \text{ror})$ IND-CPA security of Enc.

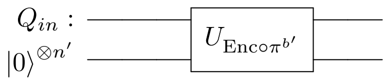
$\boldsymbol{G_0} \cong \boldsymbol{G'}$: In this case, the adversary $\mathcal{B}$ runs $\mathcal{A}$ and answers to $\mathcal{A}$'s learning queries by forwarding them as the challenge queries to the challenger. $\mathcal{B}$ will also directly forward the challenge queries made by $\mathcal{A}$ to the challenger. At the end, $\mathcal{B}$ returns the output of $\mathcal{A}$. We show that $\mathcal{B}$ simulates perfectly two games for the different type of queries separately:

1. When $\mathfrak{c}_{\text{qm}} = ST$: We recall the challenge query type (ST, ror) in the circuit below.

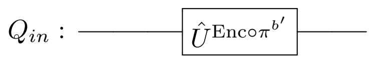
   Note that if the challenge bit $b' = 0$ then $\mathcal{B}$ simulates the learning and challenge queries in the game G_0 and if $b' = 1$ then $\mathcal{B}$ simulates the learning and challenge queries in the game G'. So the advantage of $\mathcal{B}$ in guessing the challenge bit b' is at least μ.

2. When $\mathfrak{c}_{\text{qm}} = EM$: We recall the challenge query type (EM, ror) in the circuit below.

   Note that if the challenge bit $b' = 0$ then $\mathcal{B}$ simulates the learning and challenge queries in the game G_0 and if $b' = 1$ then $\mathcal{B}$ simulates the learning and challenge queries in the game G'. So the advantage of $\mathcal{B}$ in guessing the challenge bit b' is at least μ.

3. When $\mathfrak{c}_{\text{qm}} = ER$: We recall the challenge query type (ER, ror) in the circuit below.

   It is clear that if the challenge bit b' is 0 then $\mathcal{B}$ simulates the learning queries in the game G_0 and if the challenge bit is 1 then $\mathcal{B}$ simulates the learning and challenge queries in the game G'. So the advantage of $\mathcal{B}$ in guessing the challenge bit b' is at least μ.

$\boldsymbol{G'} \cong \boldsymbol{G_1}$: We show these two games are indistinguishable for different query types:

1. When $\mathfrak{c}_{\text{qm}} = ST$. In this case, the adversary $\mathcal{B}$ answers to $\mathcal{A}$'s learning queries by forwarding them as the challenge queries to the challenger. To answer $\mathcal{A}$'s challenge queries, $\mathcal{B}$ applies a random permutation π on input register Q_{in} and sends Q_{in} and Q_{out} to the challenger. After getting the response from the challenger, it applies π^{-1} to the input register Q_{in} and sends them to the adversary $\mathcal{A}$. If the challenge bit $b' = 0$, then the adversary $\mathcal{B}$ simulates learning queries and challenge queries in the game G_1. If the challenge bit $b' = 1$, then the adversary $\mathcal{B}$ simulates learning queries and challenge queries in the game G'.

2. When $\mathfrak{c}_{\text{qm}} = EM$. The adversary $\mathcal{B}$ does the same as above except Q_{out} contains $|0\rangle^{\otimes n}$.

3. When $\mathfrak{c}_{\text{qm}} = ER$. In this case, the adversary $\mathcal{B}$ answers to $\mathcal{A}$'s learning queries by forwarding them as the challenge queries to the challenger. To answer the challenge queries, $\mathcal{B}$ applies a random permutation π on input register Q_{in} and sends it to the challenger. After getting the response from the challenger, it forwards it to the adversary $\mathcal{A}$. If the challenge bit $b' = 0$, then the adversary $\mathcal{B}$ simulates learning queries and challenge queries in the game G_1. If the challenge bit $b' = 1$, then the adversary $\mathcal{B}$ simulates learning queries and challenge queries in the game G'.

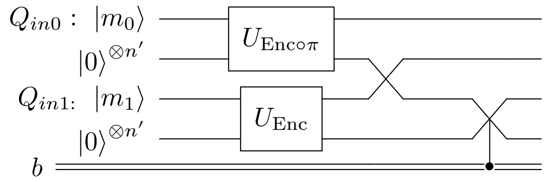
□

Theorem 5. learn$(*, EM)$-chall$(*, EM, \text{ror}) \implies$ learn$(*, EM)$-chall$(*, EM, \text{2ct})$

Proof. Let Enc be some encryption scheme that is learn($*, EM$)-chall($*, EM$, ror)-secure. We will show that Enc is learn($*, EM$)-chall($*, EM$, 2ct)-secure by showing that the settings with challenge bit $b = 0$ and $b = 1$ are indistinguishable. Since the learning queries are already the same, it is sufficient to define a sequence of games with indistinguishable challenge queries. (The learning queries are $(*, EM)$ in all cases)

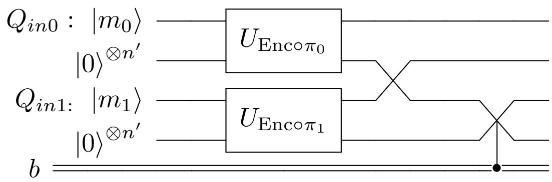
In the following we define $\mathfrak{c}^{(i)}$ challenge queries for $i = 1, 2, 3, 4$:

(i) $\mathfrak{c}^{(1)}$: On input registers Q_{in0} and Q_{in1} and the challenge bit b does the following:



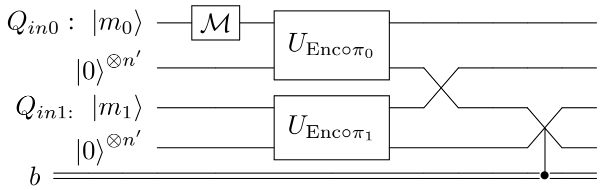
where π is a random permutation.

(ii) $\mathfrak{c}^{(2)}$: On input registers Q_{in0} and Q_{in1} and the challenge bit b does the following:



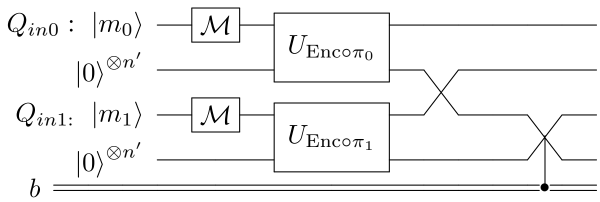
where π_0 and π_1 are random permutations.

(iii) $\mathfrak{c}^{(3)}$: On input registers Q_{in0} and Q_{in1} and the challenge bit b does the following:



where π_0 and π_1 are random permutations. The measurement outcome is discarded.

(iv) $\mathfrak{c}^{(4)}$: On input registers Q_{in0} and Q_{in1} and the challenge bit b does the following:



where π_0 and π_1 are random permutations. The measurement outcomes are discarded.

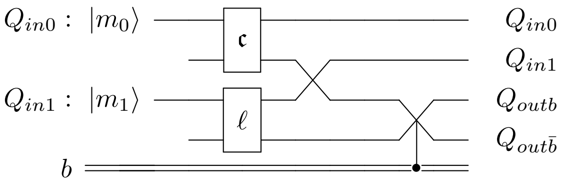
Let $\mathfrak{c}^{(0)} = \text{chall}(*, EM, \text{2ct})$. Let Game $G_b^{(i)}$ be the IND-CPA game with learn($*, EM$)-learning-queries and $\mathfrak{c}^{(i)}$-challenge-queries when the challenge bit is b, where $i \in \{0, 1, 2, 3, 4\}$.

We will show that the following sequence of games are indistinguishable from each other:

$$G_0^{(0)} \cong G_0^{(1)} \cong G_0^{(2)} \cong G_0^{(3)} \cong G_0^{(4)} \cong G_1^{(4)} \cong G_1^{(3)} \cong G_1^{(2)} \cong G_1^{(1)} \cong G_1^{(0)}$$

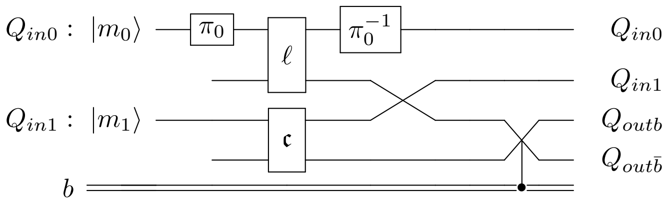
Let assume the adversary $\mathcal{A}_i$ distinguishes two games $G^{(i)}$ and $G^{(i+1)}$. In the circuits depicted below, ℓ refers to a unitary gate implementing $\ell = \text{learn}(*, EM)$ while $\mathfrak{c}$ refers to a unitary gate implementing $\mathfrak{c} = \text{chall}(*, EM, \text{ror})$.

$\boldsymbol{G_b^{(0)} \cong G_b^{(1)}}$: Let $\mathcal{A}_0$ be an adversary that distinguish $G_b^{(0)}$ and $G_b^{(1)}$. When $\mathcal{A}_0$ makes a learning query, $\mathcal{B}$ simply passes it through. When $\mathcal{A}_0$ makes a challenge query for input registers Q_{in0}, Q_{in1}, $\mathcal{B}$ simulates this by using a challenge query for the input register Q_{in0} and using a learning-query for Q_{in1}. Let Q_{out0} denote the output of the challenge query with Q_{in0} and Q_{out1} denote the output of the learning query with Q_{in1}. Then B gives the registers $Q_{in0}, Q_{in1}, Q_{outb}, Q_{out\bar{b}}$ to $\mathcal{A}_0$. We draw the circuit below in which uses a control-swap gate depends on the value of b. At the end, $\mathcal{B}$ makes the same guess b' as $\mathcal{A}_0$.



We analyse the case when $b = 0$. In this case if the challenge bit $b'' = 0$ then the adversary $\mathcal{B}$ simulates $(*, EM, 2ct)$ challenge queries and therefore it simulates game $\boldsymbol{G_0^{(0)}}$. When $b'' = 1$ then $\mathcal{B}$ returns $|m_0\rangle|m_1\rangle|\mathrm{Enc}(\pi(m_0))\rangle|\mathrm{Enc}(m_1)\rangle$ for a random permutation π. That is a $\mathfrak{c}^{(1)}$ type challenge query. In other words, $\mathcal{B}$ simulates the game $\boldsymbol{G_0^{(1)}}$. We can do the same analysis when $b = 1$.

$\boldsymbol{G_b^{(1)} \cong G_b^{(2)}}$: Let $\mathcal{A}_1$ be an adversary that distinguish $G_b^{(1)}$ and $G_b^{(2)}$. When $\mathcal{A}_1$ makes a learning query, $\mathcal{B}$ simply passes it through. When $\mathcal{A}_1$ makes a challenge query for input registers Q_{in0}, Q_{in1}, $\mathcal{B}$ simulates this by picking a random permutation π_0, applying to the register Q_{in0}, sending the result as a learning query, applying π_0^{-1} to Q_{in0} and using a challenge query for Q_{in1}. Let Q_{out0} denote the output of the learning query with Q_{in0} and Q_{out1} denote the output of the challenge query with Q_{in1}. Then B gives the registers $Q_{in0}, Q_{in1}, Q_{outb}, Q_{out\bar{b}}$ to $\mathcal{A}_0$. We draw the circuit below in which uses a control-swap gate depends on the value of b. At the end, $\mathcal{B}$ makes the same guess b' as $\mathcal{A}_0$.



We analyse when $b = 0$. In this case if the challenge bit $b'' = 0$, then the adversary $\mathcal{B}$ returns $(|m_0\rangle, |m_1\rangle, |\mathrm{Enc}(\pi_0(m_0))\rangle, |\mathrm{Enc}(m_1)\rangle)$. So $\mathcal{B}$ simulates the game $G_0^{(1)}$. If the challenge bit $b'' = 1$, then the adversary $\mathcal{B}$ returns $(|m_0\rangle, |m_1\rangle, |\mathrm{Enc}(\pi_0(m_0))\rangle, |\mathrm{Enc}(\pi_1(m_1))\rangle)$ for a random permutation π_1. That is $\mathcal{B}$ simulates the game $G_0^{(2)}$. We can do the same analysis when $b = 1$.

$\boldsymbol{G_b^{(2)} \cong G_b^{(3)}}$: These can be proven by direct application of Corollary 2. (with $f := \mathrm{Enc}_r \circ \pi_0$ and $R := \mathrm{im}\ \mathrm{Enc}_r$ for fixed randomness r.)

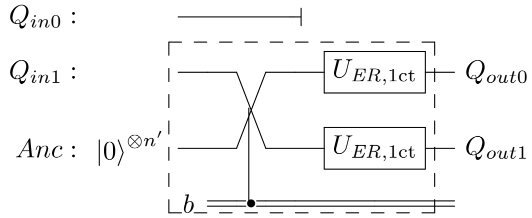
$\boldsymbol{G_b^{(3)} \cong G_b^{(4)}}$: These can be proven by direct application of Corollary 2.

$\boldsymbol{G_0^{(4)} \cong G_1^{(4)}}$: Since Enc is learn$(*, EM)$-chall$(*, EM, \mathrm{ror})$ secure, it is learn$(*, EM)$-chall$(*, CL, 2\mathrm{ct})$ secure by simulating classical queries by quantum queries (Panel 5 implies Panel 11 in Figure 1). Note that in the game $G_0^{(4)}$ the outcome of a challenge query will be $\big(m_0, m_1, \mathrm{Enc}(\pi_0(m_0)), \mathrm{Enc}(\pi_1(m_1))\big)$ and in the game $G_1^{(4)}$ it will be $\big(m_0, m_1, \mathrm{Enc}(\pi_1(m_1)), \mathrm{Enc}(\pi_0(m_0))\big)$. If there is an adversary $\mathcal{A}$ that distinguishes games $G_0^{(4)}$ and $G_1^{(4)}$, then one can construct an adversary $\mathcal{B}$ that breaks learn$(*, EM)$-chall$(*, CL, 2\mathrm{ct})$ security. The adversary $\mathcal{B}$ runs $\mathcal{A}$ and answers to its challenge queries as follows. Upon receiving the quantum registers Q_{in0} and Q_{in1} from the adversary $\mathcal{A}$, it measures the registers and gets two classical values m_0, m_1, applies random permutations π_0, π_1 to m_0, m_1, respectively, sends the result as a challenge query to its challenger, and finally forwards back the answer to $\mathcal{A}$. It is clear that when $b = 0$, the adversary $\mathcal{A}$ simulates the game $G_0^{(4)}$ and otherwise it simulates the game $G_1^{(4)}$. In conclusion, we have shown G_0 and G_1 are indistinguishable which implies Enc is learn$(*, EM)$-chall$(*, EM, 2\mathrm{ct})$ secure. □

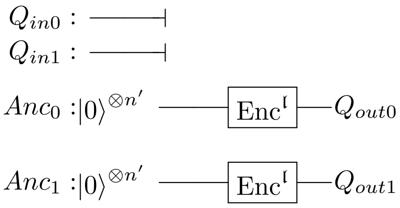
Theorem 6. learn$(*, ER)$-chall$(*, ER, 1\mathrm{ct}) \implies$ learn$(*, ER)$-chall$(*, ER, 2\mathrm{ct})$

Proof. Let Enc be some encryption scheme that is learn$(*, ER)$-chall$(*, ER, 1\mathrm{ct})$-secure. We will show that Enc is learn$(*, ER)$-chall$(*, ER, 2\mathrm{ct})$-secure by showing that the settings with challenge bit $b = 0$ and $b = 1$ are indistinguishable. The learning queries will be learn$(*, ER)$ in all games.

Define the challenge query $\mathfrak{c}'_b$ as follows: on input registers Q_{in0}, Q_{in1}, discard the register Q_{in0}, prepares an ancillary register Anc containing $|0\rangle^{\otimes n'}$ and use the chall($*, ER$, 1ct)-challenge-query (the dashed box below) for the registers Q_{in1}, Anc as follows:



where $U_{ER,1\text{ct}}$ is $\hat{U}^{\text{Enc}(\cdot,r_0)}/\hat{U}^{\text{Enc}(\cdot,r_1)}$. Define the challenge query $\mathfrak{c}''$ as follows: on input registers Q_{in0}, Q_{in1}, it discards Q_{in0}, Q_{in1}, prepares ancillary registers Anc_0 and Anc_1 containing $|0\rangle^{\otimes n'}$ and use learning queries for Anc_0, Anc_1. The quantum circuit for a $\mathfrak{c}''$ query is shown below.



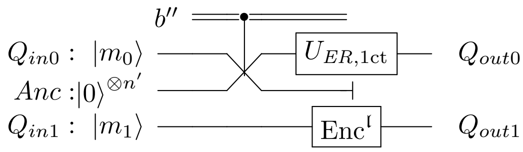
where $\text{Enc}^{\mathfrak{l}}$ is $\hat{U}^{\text{Enc}}$. Let Game G_b be the IND-CPA game with chall($*, ER$, 2ct)-challenge-queries when the challenge bit is b. Let Game G'_b be the IND-CPA game with $\mathfrak{c}'_b$-challenge queries. Let Game G'' be the IND-CPA game with $\mathfrak{c}''$-challenge queries.

We will show that the following sequence of games are indistinguishable from each other:

$$G_0 \cong G'_1 \cong G'' \cong G'_0 \cong G_1.$$

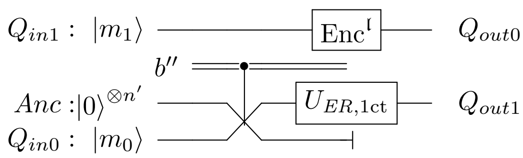
To do this, we construct an adversary $\mathcal{B}$ that breaks learn($*, ER$)-chall($*, ER$, 1ct)-security from an adversary $\mathcal{A}$ that distinguishes two consecutive games. Let b' denotes $\mathcal{A}$'s guess and b'' denotes the challenge bit of $\mathcal{B}$'s challenger.

$\boldsymbol{G_0 \cong G'_1}$ **:** When $\mathcal{A}$ submits the input registers Q_{in0}, Q_{in1} as a challenge query, $\mathcal{B}$ simulates this by using a learn($*, ER$)-learning query for Q_{in1} to get the second output register, prepares an ancillary register Anc containing $|0\rangle^{\otimes n'}$, and making a chall($*, ER$, 1ct)-challenge-query for Q_{in0}, Anc to get the first output register. At the end $\mathcal{B}$ makes the same guess as $\mathcal{A}$.



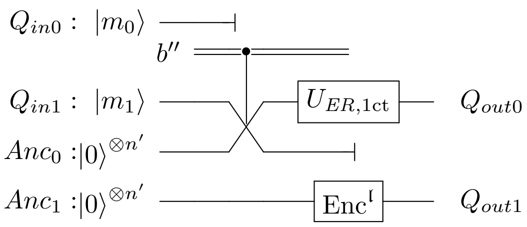
If the challenger bit $b'' = 0$ the adversary $\mathcal{B}$ will receive $|\text{Enc}(m_0)\rangle$ back from its challenger and sends $(|\text{Enc}(m_0)\rangle, |\text{Enc}(m_1)\rangle)$ to $\mathcal{A}$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G_0 when $b'' = 0$. If the challenger bit $b'' = 1$ the adversary $\mathcal{B}$ will receive $|\text{Enc}(0)\rangle$ back from its challenger and sends $(|\text{Enc}(0)\rangle, |\text{Enc}(m_1)\rangle)$ to $\mathcal{A}$. Note that this is an $\mathfrak{c}'_1$ type challenge query, therefore, $\mathcal{B}$ simulates the challenge queries in game G'_1 when $b'' = 1$.

$\boldsymbol{G_1 \cong G'_0}$ **:** When $\mathcal{A}$ submits the input registers Q_{in0}, Q_{in1} as a challenge query, $\mathcal{B}$ simulates this by using a learn($*, ER$)-learning query for Q_{in1} to get the first output register, prepares an ancillary register Anc containing $|0\rangle^{\otimes n'}$, and making a chall($*, ER$, 1ct)-challenge-query for Anc, Q_{in0} to get the second output register. At the end, $\mathcal{B}$ returns $\mathcal{A}$'s guess.



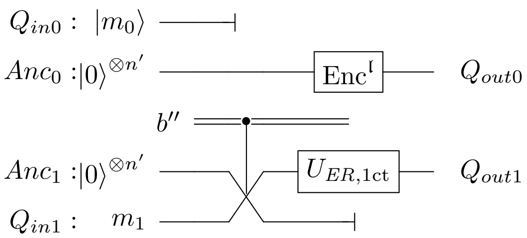
If the challenger bit $b'' = 0$ the adversary $\mathcal{B}$ will receive $|\text{Enc}(0)\rangle$ back from its challenger and sends $(|\text{Enc}(m_1)\rangle, |\text{Enc}(0)\rangle)$ to $\mathcal{A}$. Note that this is an $\mathfrak{c}'_0$ type challenge query, therefore, $\mathcal{B}$ simulates the challenge queries in game G'_0 when $b'' = 0$. If the challenger bit $b'' = 1$ the adversary $\mathcal{B}$ will receive $|\text{Enc}(m_0)\rangle$ back from its challenger and sends $(|\text{Enc}(m_1)\rangle, |\text{Enc}(m_0)\rangle)$ to $\mathcal{A}$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G_1 when $b'' = 1$.

$\boldsymbol{G'_0 \cong G''}$ **:** When $\mathcal{A}$ makes a challenge query by submitting the input registers Q_{in0} and Q_{in1}, $\mathcal{B}$ answers this by discarding the register Q_{in0}, preparing ancillary registers Anc_0, Anc_1 containing $|0\rangle^{\otimes n'}$, making a chall$(*, ER, \text{1ct})$-challenge-query for Q_{in1}, Anc_0 to get output register Q_{out0}, and using a learning query for Anc_1 to get the output register Q_{out1}. At the end, $\mathcal{B}$ makes the same guess b' as $\mathcal{A}$ where $b' = 1$ means $\mathcal{A}$ interacts in game G''.



When the challenge bit $b'' = 0$, $\mathcal{B}$ will receive back $|\text{Enc}(m_1)\rangle$ and sends $(|\text{Enc}(m_1)\rangle, |\text{Enc}(0)\rangle)$ to $\mathcal{A}$. Hence, $\mathcal{B}$ simulates the challenge queries in game G'_0. When the challenge bit $b'' = 1$ it will receive back $|\text{Enc}(0)\rangle$ and sends $(|\text{Enc}(0)\rangle, |\text{Enc}(0)\rangle)$ to $\mathcal{A}$. Hence, $\mathcal{B}$ simulates the challenge queries in game G'' in this case.

$\boldsymbol{G'_1 \cong G''}$ **:** When $\mathcal{A}$ makes a challenge query by submitting the input registers Q_{in0} and Q_{in1}, $\mathcal{B}$ answers this by discarding the register Q_{in0}, preparing ancillary registers Anc_0, Anc_1 containing $|0\rangle^{\otimes n'}$, making a chall$(*, ER, \text{1ct})$-challenge-query for Anc_1, Q_{in1} to get the output register Q_{out1}, and using a learning query for Anc_0 to get the output register Q_{out0}. At the end, $\mathcal{B}$ makes the same guess b' as $\mathcal{A}$ where $b' = 0$ means $\mathcal{A}$ interacts in game G''.



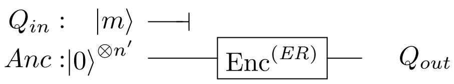
When the challenge bit $b'' = 0$, $\mathcal{B}$ will receive back $|\text{Enc}(0)\rangle$ and sends $(|\text{Enc}(0)\rangle, |\text{Enc}(0)\rangle)$ to $\mathcal{A}$. Hence, $\mathcal{B}$ simulates the challenge queries in game G''. When the challenge bit $b'' = 1$ it will receive back $|\text{Enc}(m_1)\rangle$ and sends $(|\text{Enc}(0)\rangle, |\text{Enc}(m_1)\rangle)$ to $\mathcal{A}$. Hence, $\mathcal{B}$ simulates the challenge queries in game G'_1 in this case. □

Theorem 7. *The following implications hold:*

- learn$(*, CL)$, chall$(1, ER, \text{1ct}) \implies$ learn$(*, CL)$-chall$(1, ER, \text{ror})$.
- learn$(*, ER)$-chall$(*, ER, \text{1ct}) \implies$ learn$(*, ER)$-chall$(*, ER, \text{ror})$

Proof. We prove the second implication and the first one can be proven analogously. Let Enc be an encryption scheme that is learn$(*, ER)$-chall$(*, ER, \text{1ct})$-secure. We will show that Enc is learn$(*, ER)$-chall$(*, ER, \text{ror})$-secure by showing that the settings with challenge bit $b = 0$ and $b = 1$ are indistinguishable. Since the learning queries are already the same, it is sufficient to define a sequence of games with indistinguishable challenge queries.

Define the challenge query $\mathfrak{c}'$ as follows: Upon receiving the input register Q_{in}, discard it and instead make a ER learning query for $|0\rangle$.



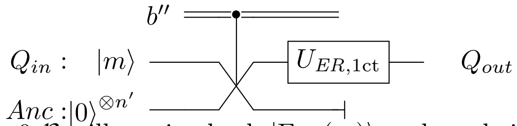
Let Game G_b be the IND-CPA game with chall$(*, ER, \text{ror})$-challenge-queries and CL-learning-queries when the challenge bit is b. Let Game G' be the IND-CPA game with $\mathfrak{c}'$-challenge-queries and CL-learning-queries.

Next we will show in sequence that these games are indistinguishable from one another:

$$G_0 \cong G' \cong G_1$$

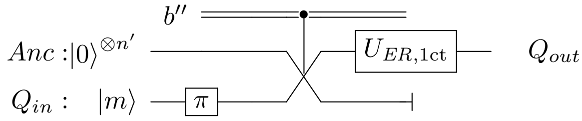
To do this, we construct an adversary $\mathcal{B}$ that breaks learn($*$, CL)-chall($*$, ER, 1ct)-security from an adversary $\mathcal{A}_b$ that distinguishes the game G_b from G'. Let b'' denotes the $\mathcal{B}$'s challenge bit.

$\boldsymbol{G_0 \cong G'}$ **:** When $\mathcal{A}_0$ makes a challenge query by submitting the input register Q_{in}, $\mathcal{B}$ answers this by preparing an ancillary register Anc containing $|0\rangle^{\otimes n'}$, and then sending the registers Q_{in}, Anc to its challenger and forwards back the result to $\mathcal{A}_0$.



If the challenge bit $b'' = 0$ $\mathcal{B}$ will receive back $|\text{Enc}(m)\rangle$ and sends it to $\mathcal{A}_0$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G_0. If the challenge bit $b'' = 0$ $\mathcal{B}$ will receive back $|\text{Enc}(0)\rangle$ and sends it to $\mathcal{A}_0$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G'.

$\boldsymbol{G_1 \cong G'}$ **:** When $\mathcal{A}_1$ makes a challenge query by submitting the input register Q_{in}, $\mathcal{B}$ answers this by preparing an ancillary register Anc containing $|0\rangle^{\otimes n'}$, picking a qPRP π and applying it to the register Q_{in}, then sending the registers Anc, Q_{in} to its challenger and forwarding back the result to $\mathcal{A}_1$.



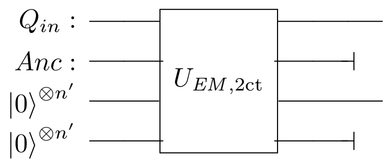
If the challenge bit $b'' = 0$ $\mathcal{B}$ will receive back $|\text{Enc}(0)\rangle$ and sends it to $\mathcal{A}_1$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G'. If the challenge bit $b'' = 0$, $\mathcal{B}$ will receive back $|\text{Enc}(\pi(m))\rangle$ and sends it to $\mathcal{A}_1$. Therefore, $\mathcal{B}$ simulates the challenge queries in game G_1.

□

Theorem 8. *The following implications hold:*

- learn($*$, CL)-chall($1, EM, 2\text{ct}$) $\implies$ learn($*$, CL)-chall($1, EM, \text{ror}$)
- learn($*$, EM)-chall($*, EM, 2\text{ct}$) $\implies$ learn($*$, EM)-chall($*, EM, \text{ror}$)

Proof. We prove the first implication and the second one can be proven analogously. Let Enc be an encryption scheme that is learn($*$, CL)-chall($1, EM, 2\text{ct}$) secure. Consider an adversary $\mathcal{A}$ that is successful in attacking Enc in the sense of learn($*$, CL)-chall($1, EM, \text{ror}$)-queries. Let G_b be the IND-CPA game against learn($*$, CL)-chall($1, EM, \text{ror}$)-queries when the challenge bit is b. By Corollary 2, if we measure the input register in the game G_1 this can not be detected by the adversary $\mathcal{A}$. (Note that each query uses a different random permutation π and uses it only once.) We define G_1' to be similar to the game G_1 except with a measurement in the computational basis on the input register submitted as a challenge query. The games G_1 and G_1' are indistinguishable by Corollary 2. We define G_1'' to be similar to the game G_1' except for each challenge query the input register will be initiated with a random classical value. It is clear that G_1'' and G_1' are indistinguishable. Since G_1 and G_1'' are indistinguishable, $\mathcal{A}$ can distinguish the games G_1'' and G_0. We define an adversary $\mathcal{B}$ against learn($*$, CL)-chall($1, EM, 2\text{ct}$), which uses $\mathcal{A}$ as follows: when $\mathcal{A}$ makes a chall($1, EM, \text{ror}$)-challenge-query by submitting the input register Q_{in}, $\mathcal{B}$ prepares an ancillary register Anc containing a random classical value and sends Q_{in}, Anc as a challenge query to its challenger. Upon receiving the response from the challenger, it discards the second and the fourth register and forwards the first and the third register to $\mathcal{A}$. At the end, $\mathcal{B}$ returns $\mathcal{A}$'s guess.



Let b' be the $\mathcal{B}$'s challenger bit. If $b' = 0$, $\mathcal{B}$ simulates the response in the game G_0 but if $b' = 1$, $\mathcal{B}$ simulates the response in the game G_1''. Therefore B can breaks the security of Enc against learn($*$, CL) -chall($1, EM, 2\text{ct}$). □

Theorem 9. learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(*, ST)$-chall$(*, CL, 2\text{ct})$. *This shows that P1 $\implies$ P6.*

Proof. This has been proven in [GHS16] using multiple implications. Refer to Figure 2 in [GHS16] such that "gqIND-qCPA" in the figure is learn$(*, ER)$-chall$(*, ER, 1\text{ct})$ in our notation and "'IND-qCPA" in the figure is learn$(*, ST)$-chall$(*, CL, 2\text{ct})$ in our notation. ☐

7 Separations

In this section all possible implications between different notions of IND-CPA security that are not shown in Figure 1 or do not follow from it by transitivity are disproven here, apart from the nonimplications stated in Conjecture 1, which we leave as a conjucture. First we give an overview of results in this section.

7.1 Overview of results

In the following, we use two rules to show non-implications:

- if $A \not\Rightarrow B$ and $C \implies B$ then we can deduce $A \not\Rightarrow C$.
- if $A \not\Rightarrow B$ and $A \implies C$ then we can conclude $C \not\Rightarrow B$.

Panel 1: From the Figure 1, we can conclude that $P1 \implies P3, P4, P5, P6, P7, P8, P9, P10, P11, P13, P14$. So it is only left to show the relation between $P1$ and $P2, P12$. From Theorem 17 $P1 \not\Rightarrow P12$ and as a corollary $P1 \not\Rightarrow P2$ because $P2 \implies P12$. Therefore

$$P1 \not\Rightarrow P2, P12.$$

This finishes all of implication and non-implications from $P1$.

Panel 2: From the Figure 1, we can conclude that $P2$ implies $P5, P6, P7, P11, P12, P13, P14$. We show in Corollary 3, $P2 \not\Rightarrow P8$ and since $P1, P3 \implies P8$ then $P2 \not\Rightarrow P1, P3$. In Theorem 14 we show $P2 \not\Rightarrow P10$ and since $P4 \implies P10$ then $P2 \not\Rightarrow P4$. From Conjecture 1, $P2 \not\Rightarrow P9$. Therefore,

$$P2 \not\Rightarrow P1, P3, P4, P8, P9, P10$$

This finishes all of implication and non-implications from $P2$

Panel 3: From the Figure 1, we can conclude that $P3 \implies P7, P8, P9, P13, P14$. Since $P1 \implies P3$ and $P1 \not\Rightarrow P2, P12$, we can deduce $P3 \not\Rightarrow P2, P12$. From Conjecture 1 $P3 \not\Rightarrow P11$ and since $P1, P4, P5, P6, P10 \implies P11$, we can deduce $P3 \not\Rightarrow P1, P4, P5, P6, P10$ Therefore,

$$P3 \not\Rightarrow P1, P2, P4, P5, P6, P10, P11, P12.$$

This finishes all of implication and non-implications from $P3$.

Panel 4: From the Figure 1, we can conclude that $P4 \implies P5, P7, P9, P10, P11, P13, P14$. From Corollary 3, $P4 \not\Rightarrow P8$ and since $P1, P3 \implies P8$ then $P4 \not\Rightarrow P1, P3$. Since $P1 \not\Rightarrow P2, P12$ and $P1 \implies P4$ then we can deduce $P4 \not\Rightarrow P2, P12$. From Conjecture 1, $P4 \not\Rightarrow P6$. Therefore,

$$P4 \not\Rightarrow P1, P3, P2, P6, P8, P12.$$

Panel 5: From the Figure 1, we can conclude that $P5 \implies P7, P11, P13, P14$. Since $P1 \implies P5$ and $P1 \not\Rightarrow P2, P12$, then $P5 \not\Rightarrow P2, P12$. Since $P2 \implies P5$ and $P2 \not\Rightarrow P1, P3, P4, P8, P9, P10$, we can deduce $P5 \not\Rightarrow P1, P3, P4, P8, P9, P10$ (note that $P5 \not\Rightarrow P9$ is based on conjecture $P2 \not\Rightarrow P9$). Since $P4 \implies P5$ and $P4 \not\Rightarrow P6$ (from Conjecture 1) then $P5 \not\Rightarrow P6$ (Note that this non-implication is based on a conjecture). Therefore,

$$P5 \not\Rightarrow P1, P2, P3, P4, P6, P8, P9, P10, P12.$$

Panel 6: From the Figure 1, $P6 \implies P11, P14$. Since $P2 \implies P6$ and $P2 \not\Rightarrow P1, P3, P4, P8, P10, P12$, then we can conclude that $P6 \not\Rightarrow P1, P3, P4, P8, P10, P12$. We show in Theorem 15 that $P6 \not\Rightarrow P7$ and since $P2, P5 \implies P7$ then we can deduce $P6 \not\Rightarrow P2, P5$. From Theorem 16, $P6 \not\Rightarrow P13$ and since $P9 \implies P13$, then $P6 \not\Rightarrow P9$. Therefore

$$P6 \not\Rightarrow P1, P2, P3, P4, P5, P7, P8, P9, P10, P12, P13.$$

We cover all implication and non-implications from $P6$.

Panel 7: From the Figure 1, $P7 \implies P13, P14$. Since $P1 \not\Rightarrow P2, P12$ and $P1 \implies P7$, then $P7 \not\Rightarrow P2, P12$. Since $P2 \not\Rightarrow P1, P3, P4, P8, P9, P10$ and $P2 \implies P7$, then $P7 \not\Rightarrow P1, P3, P4, P8, P9, P10$ (note that $P7 \not\Rightarrow P9$ is based on conjecture $P2 \not\Rightarrow P9$). Since $P3 \not\Rightarrow P5, P6, P11$ and $P3 \implies P7$, then $P7 \not\Rightarrow P5, P6, P12$. (Note that these non-implications are based on conjectures $P3 \not\Rightarrow P11$.) Therefore,

$$P7 \not\Rightarrow P1, P2, P3, P4, P5, P6, P8, P9, P10, P11, P12.$$

This covers all the cases.

Panel 8: From the Figure 1, $P8 \implies P9, P13, P14$. Since $P1 \not\Rightarrow P2, P12$ and $P1 \implies P8$, then $P8 \not\Rightarrow P2, P12$. From Conjecture 1, $P8 \not\Rightarrow P7$ and since $P1, P3, P4, P5 \implies P7$ then $P8 \not\Rightarrow P1, P3, P4, P5$. Since $P3 \not\Rightarrow P6, P10, P11$ and $P3 \implies P8$, then $P8 \not\Rightarrow P6, P10, P11$. (Note that these non-implications are based on conjecture $P3 \not\Rightarrow P11$.) Therefore,

$$P8 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P10, P11, P12.$$

This covers all the cases.

Panel 9: From the Figure 1, $P9 \implies P13, P14$. Since $P4 \not\Rightarrow P1, P2, P3, P8, P12$ and $P4 \implies P9$, then $P9 \not\Rightarrow P1, P2, P3, P8, P12$. Since $P3 \not\Rightarrow P4, P5, P6, P10, P11$ and $P3 \implies P9$, then $P9 \not\Rightarrow P4, P5, P6, P10, P11$. (Note that these non-implications are based on conjectures $P3 \not\Rightarrow P11$.) Since $P8 \not\Rightarrow P7$ from Conjecture 1 and $P8 \implies P9$ then $P9 \not\Rightarrow P7$. Therefore,

$$P9 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P10, P11, P12$$

Panel 10: From the Figure 1, $P10 \implies P11, P14$. Since $P4 \not\Rightarrow P1, P2, P3, P12$ and $P4 \implies P10$ then $P10 \not\Rightarrow P1, P2, P3, P12$. We show in Theorem 15 $P10 \not\Rightarrow P7$ and since $P4, P5 \implies P7$ then $P10 \not\Rightarrow P4, P5$. From Theorem 13, $P10 \not\Rightarrow P13$ and since $P8, P9 \implies P13$ then $P10 \not\Rightarrow P8, P9$. From Conjecture 1, $P4 \not\Rightarrow P6$ and since $P4 \implies P10$ $P10 \not\Rightarrow P6$. Therefore,

$$P10 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P9, P12, P13.$$

Panel 11: From the Figure 1, $P11 \implies P14$. Since $P6 \not\Rightarrow P1, P2, P3, P4, P5, P7, P8, P9, P10, P12, P13$ and $P6 \implies P11$ then $P11 \not\Rightarrow P1, P2, P3, P4, P5, P7, P8, P9, P10, P12, P13$. From Conjecture 1 $P4 \not\Rightarrow P6$ and since $P4 \implies P11$, then $P11 \not\Rightarrow P6$. Therefore,

$$P11 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P12, P13.$$

This covers all the cases.

Panel 12: From the Figure 1, $P12 \implies P13, P14$. Since $P2 \not\Rightarrow P1, P3, P4, P8, P9, P10$ and $P2 \implies P12$ then $P12 \not\Rightarrow P1, P3, P4, P8, P9, P10$ (note that $P12 \not\Rightarrow P9$ is based on conjecture $P2 \not\Rightarrow P9$). From Conjecture 1, $P12 \not\Rightarrow P7$ and since $P2, P5 \implies P7$ then $P12 \not\Rightarrow P2, P5$. From Conjecture 1, $P12 \not\Rightarrow P11$ and since $P6 \implies P11$ then $P12 \not\Rightarrow P6$. Therefore,

$$P12 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12$$

Panel 13: From the Figure 1, $P13 \implies P14$. Since $P1 \not\Rightarrow P2, P12$ and $P1 \implies P13$, then $P13 \not\Rightarrow P2, P12$. Since $P2 \not\Rightarrow P1, P3, P4, P8, P9, P10$ and $P2 \implies P13$, then $P13 \not\Rightarrow P1, P3, P4, P8, P9, P10$ (note that $P13 \not\Rightarrow P9$ is based on conjecture $P2 \not\Rightarrow P9$.) Since $P3 \not\Rightarrow P5, P6, P11$ and $P3 \implies P13$, then $P13 \not\Rightarrow P5, P6, P11$. (Note that these non-implications are based on conjectures $P3 \not\Rightarrow P11$.) Since $P8 \not\Rightarrow P7$ from Conjecture 1 and $P8 \implies P13$ then $P13 \not\Rightarrow P7$. Therefore,

$$P13 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12.$$

Panel 14: Since $P6 \not\Rightarrow P1, P2, P3, P4, P5, P7, P8, P9, P10, P12, P13$ and $P6 \implies P14$ then $P14 \not\Rightarrow P1, P2, P3, P4, P5, P7, P8, P9, P10, P12, P13$. Form Theorem 12, $P14 \not\Rightarrow P11$ and since $P6 \implies P11$, then $P14 \not\Rightarrow P6$. Since $P14 \not\Rightarrow P13$ and $P7 \implies P13$ then $P14 \not\Rightarrow P7$. Therefore,

$$P14 \not\Rightarrow P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13$$

7.2 Separations by Quasi-Length-Preserving Encryptions

The notion of a core function and quasi-length-preserving encryption schemes was first formally introduced in [GHS16]. Intuitively, the definition splits the ciphertext into a message-independent part and a message-dependent part that has the same length as the plaintext. We define a variant of a quasi-length-preserving encryption scheme below.

Definition 6 (Core function)**.** *A function $g : \{0,1\}^h \times \{0,1\}^t \times \{0,1\}^n$ is called the core function of an encryption scheme* (KGen, Enc, Dec) *if*

1. *for all $k \in \{0,1\}^h, m \in \{0,1\}^n, r \in \{0,1\}^t$,*

$$\mathrm{Enc}_k(m; r) = f(k, r) || g(k, m, r)$$

where f is an arbitrary function independent of the message.

2. *there exists a function f' such that for all $k \in \{0,1\}^h, m \in \{0,1\}^n, r \in \{0,1\}^t$ we have $f'(k, f(k, r), g(k, m, r)) = m$.*

Definition 7 (Quasi-Length-Preserving)**.** *An encryption scheme with core function g is said to be* ***quasi-length-preserving*** *if for all $k \in \{0,1\}^h, m \in \{0,1\}^n, r \in \{0,1\}^t$,*

$$|g(k, m, r)| = |m|,$$

that is, the output of the core function has the same length as the message.

In the theorem below we show that any quasi-length-preserving encryption scheme is insecure for the query model in Panel 8. And as a corollary any quasi-length-preserving encryption scheme is insecure for any query models in Panel 1 and Panel 3 because they imply Panel 8 Figure 1. (This corollary can be derived directly from the proof of the theorem below since the attack does not use learning queries.)

Theorem 10. *Any quasi-length-preserving encryption scheme is insecure for the query model* learn($*$, CL) -chall(1, ER, 1ct)*. This shows that any quasi-length-preserving encryption scheme is insecure for the query model in Panel 8.*

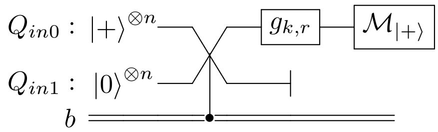
Proof. Suppose the function Enc is quasi-length-preserving, i.e., we can write

$$\mathrm{Enc}_k(m; r) = f(k, r) || g(k, m, r)$$

for some functions f and g such that

$$|g(k, m, r)| = |m|.$$

Since the encryption function is decryptable and quasi-length-preserving then g is essentially a permutation for fixed k, r. Now in the challenge query, the adversary prepares two input registers Q_{in0}, Q_{in1} containing the uniform superposition of all messages and $|0\rangle^{\otimes n}$, respectively. After getting the outcome, the adversary performs the projective measurement $\mathcal{M}_{|+\rangle}$ on the output register to determine whether it is in the state $|+\rangle^{\otimes n}$ or not. We draw the circuit below. For simplicity, we omit the classical values of $f(k, r)$ from the circuits.



When $b = 0$ the measurement $\mathcal{M}_{|+\rangle}$ succeeds with probability 1, but when $b = 1$, this happens only with negligible probability. ☐

In the theorem below we choose two query models from Panel 2 and Panel 4 and we propose a quasi-length-preserving encryption function that is secure in those two security notions. Then we can conclude that there is a quasi-length-preserving encryption function that is secure for any query models in Panel 2 and Panel 4 because query models inside of panels are equivalent. (This can be concluded directly from the proof of the theorem below as well.)

Theorem 11. *If there exists a quantum secure one-way function then for query models*

$$\text{learn}(*, \mathfrak{q}_{\text{qm}})\text{-chall}(1, \mathfrak{q}_{\text{qm}}, \text{ror}) \text{ when } \mathfrak{q}_{\text{qm}} \in \{ST, ER\}$$

there is a quasi-length-preserving encryption function that is secure. This shows that there is a quasi-length-preserving encryption function that is secure for any query models in Panels 2,4

Proof. Let

$$\text{Enc}_k(m; r) = \text{sPRF}_k(r) || \text{qPRP}_r(m)$$

where qPRP is a strong quantum-secure pseudo-random permutation [Zha16] and sPRF is a standard-secure pseudo-random function [Zha12]. Because fresh randomness is used in each learning and challenge query and sPRF_k is indistinguishable from a truly random function, we can replace $\text{sPRP}_k(r)$ with a random value in each (learning and challenge) query. This makes the second part of ciphertext independent of the first part in each query. Therefore in each query we have that qPRP_r is indistinguishable from a fresh truly random permutation σ. Therefore, with ror-type challenge queries, the adversary cannot distinguish an encryption of m from an encryption of $\pi(m)$ for a truly random permutation π because σ and $\sigma \circ \pi$ are indistinguishable. ☐

Corollary 3. *The security notions mentioned in Theorem 11 do not imply the security notions mentioned in Theorem 10. Specifically, P2, P4* $\not\Longrightarrow$ *P8.*

7.3 Separations by Simon's Algorithm

Roughly speaking, in this section we construct a couple of separating examples making use of the fact that Simon's algorithm (see [Sim97]) can only be executed by an quantum adversary with superposition access to the black box function, but not by a quantum adversary with classical access to the black box function.

The idea is to define a function $F_{s,\sigma}$ (s being a random bitstring) that is supposed to leak some bitstring σ to an adversary with superposition access to $F_{s,\sigma}$ but not to an adversary who has only classical access to $F_{s,\sigma}$. Namely the adversary with superposition access uses Simon's algorithm to retrieve σ. Roughly speaking $F_{s,\sigma}$ is composed of many small block functions $f_{s,\sigma,i}$, $i = 1, \ldots, \hat{n}$ and each of them leaking about one bit. It is proven in [Sim97] that $\hat{n} = O(|\sigma|)$ suffice to recover σ (see later).

The function $F_{s,\sigma}$ is first defined and then it is used several times in this subsection as a building block to construct separating examples for diverse IND-CPA-notions.

Definition 8. *Let* $s = s_1 || \ldots || s_{\hat{n}} || r_1 || \ldots || r_{\hat{n}}$ *be a random bitstring. Let* P_{s_i} *be a quantum secure pseudorandom permutation*[3] *(qPRP) with the seed* s_i *and input/output length of* $n/2$. *Let*

$$g_{s,\sigma,i}(y) = P_{s_i}(y) \oplus P_{s_i}(y \oplus \sigma) \quad \text{and} \quad f_{s,\sigma,i}(y) = g_{s,\sigma,i}(y) || (y \oplus r_i).$$

Note that when ignoring second part of $f_{s,\sigma,i}$ *it is* σ*-periodic. The second part makes* $f_{s,\sigma,i}$ *injective. Note that the inverse of* $f_{s,\sigma,i}$ *is easy to compute. Let*

$$F_{s,\sigma}(x) = f_{s,\sigma,1}(x_1) || \ldots || f_{s,\sigma,\hat{n}}(x_{\hat{n}})$$

where x_i *is i-th block of* x. *Note that* $F_{s,\sigma}$ *will be decryptable using* s *since each of* $f_{s,\sigma,i}$ *is decryptaable.*

[3]Quantum secure pseudorandom permutation can be constructed from a quantum secure one-way function [Zha16].

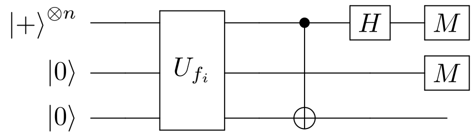
Lemma 6. *On the assumption of existing a quantum secure one-way function and for a random secret s and known $\sigma \neq 0$, $F_{s,\sigma}$ is classically one-query-indistinguishable from a truly random function.*

Proof. We show that for every i and y, $f_{s,\sigma,i}(y)$ is indistinguishable from a random bitstring. Since $y \oplus r_i$ is indistinguishable from a random bitstring (for random r_i), it is left to show $g_{s,\sigma,i}(y) = P_{s_i}(y) \oplus P_{s_i}(y \oplus \sigma)$ is indistinguishable from a random bitstring. The result follows because P_{s_i} is a pseudorandom permutation.

□

Lemma 7. *An adversary having one-query-EM-type quantum access to $F_{s,\sigma}$ can guess σ with high probability. (The reason we are looking at the embedding query model is because it is the weakest, the same statements for the standard and the erasing query model follow automatically.)*

Proof. The attack is a variation of Simon's attack [Sim97]. Remember that $F_{s,\sigma}$ consists of $\hat{n}$-many block function $f_{s,\sigma,i}$. In the analysis below, we shorten $f_{s,\sigma,i}$ to f_i and $g_{s,\sigma,i}$ to g_i. In the attack the same operation is done with each of the f_i. Namely the attack on one of the f_i happens according to the following quantum circuit:



The evolution of the quantum state right after CNOT gate is

$$2^{-\frac{n}{2}} \sum_m |m, 0, 0\rangle \mapsto 2^{-\frac{n}{2}} \sum_m |m, g_i(m), m \oplus r_i\rangle \mapsto 2^{-\frac{n}{2}} \sum_m |m, g_i(m), r_i\rangle$$

The last register contains a classical value and therefore it does not interfere the analysis of Simon's algorithm for the function g_i. So the measurement returns a random m such that $m \cdot \sigma = 0$.

Hence it yields a linear equation about σ. As this happens for every block, the adversary gets $\hat{n}$ linear equations about σ, so by the choice of $\hat{n}$ (i.e. $\hat{n} = 2|\sigma|$) the adversary is able to retrieve σ with high probability. □

Theorem 12. *If there exists a quantum secure one-way function then* learn($*$, *CL*)-chall($*$, *CL*, 1ct) $\not\Longrightarrow$ learn($*$, *EM*)-chall(1, *CL*, 1ct). *This shows that Panel 14 $\not\Longrightarrow$ Panel 11.*

Proof. Consider

$$\mathrm{Enc}_{k,k'}(m, m'; r||r') = F_{r,k}(m)||\mathrm{PRF}_{k'}(r)||(\mathrm{PRF}_k(r') \oplus m')||r',$$

where PRF_k and $\mathrm{PRF}_{k'}$ are standard secure pseudorandom functions with the key k, k' respectively. $\mathrm{Enc}_{k,k'}$ is decryptable because using the secret key k and the last part of ciphertext (r') we can obtain m' and using the secret key k' we can obtain the randomness r and then decrypt $F_{r,k}$. We prove Enc is learn($*$, *CL*)-chall($*$, *CL*, 1ct)-secure. In every query, since r is fresh randomness and $\mathrm{PRF}_{k'}$ is a pseudorandom function, we can replace $\mathrm{PRF}_{k'}(r)$ with a random bitstring. Now we can use Lemma 6 to replace $F_{r,k}(m)$ with a random bitstring. Finally, since r' is a fresh randomness in each query and PRF_k is a pseudorandom function we can replace $\mathrm{PRF}_k(r') \oplus m'$ with a random bitstring. Therefore, in each query the encryption scheme just returns a random looking bitstring, which obviously hides b. This proves the learn($*$, *CL*)-chall($*$, *CL*, 1ct)-security. We show the learn($*$, *EM*)-chall(1, *CL*, 1ct)-insecurity. In the attack, the adversary uses one learning query to retrieve k, according to Lemma 7 and then the challenge query can be trivially distinguished by decrypting the third part of the challenge ciphertext (adversary knows k, r' and can decrypt $\mathrm{PRF}_k(r') \oplus m'$.) □

Theorem 13. *If there exists a quantum secure one-way function then the following nonimplication holds:*

learn($*$, *ER*)-chall($*$, *CL*, 1ct) $\not\Longrightarrow$ learn($*$, *CL*)-chall(1, *EM*, ror).

This means that P10 $\not\Longrightarrow$ P13.

Proof. The idea of the proof is like in the last theorem to open up a backdoor that only a quantum adversary can use. We define Enc as follows.

$$\mathrm{Enc}_k(z||x;l||s) = \mathrm{sPRP}_k(l||s)||\mathrm{qPRP}_l(z)||F_{s,l}(x)$$

where $F_{s,l}$ is defined in Definition 8. Enc_k is decryptable since we can obtain l, s from $\mathrm{sPRP}_k(l||s)$ and then decrypt $\mathrm{qPRP}_l(z)$ using l and decrypt $F_{s,l}(x)$ using s, l. Now we show that Enc is insecure in the learn($*$, CL)-chall($*$, EM, ror)-sense. The attack works as follows: $\mathcal{A}$ chooses $z = 0^n$ and puts in the register for x a superposition of the form $|+\rangle^{\otimes n}$. Then $\mathcal{A}$ passes the result as a challenge query to the challenger. Upon receiving the answer from the challenger, $\mathcal{A}$ performs the algorithm presented in Lemma 7 to the last part of the ciphertext to recover l. Let $\hat{l}$ be the output of the algorithm presented in Lemma 7. Then $\mathcal{A}$ uses $\hat{l}$ to decrypt the classical part of the challenge ciphertext, $\mathrm{qPRP}_l(z)$. Let $\hat{c}$ be the output of the decryption using $\hat{l}$. If $\hat{c} = 0^n$ $\mathcal{A}$ returns 0, otherwise it returns 1. We analyse how $\mathcal{A}$ can distinguish the two cases when the challenge bit is $b = 0$ and $b = 1$. When the challenge bit is $b = 0$, the algorithm in Lemma 7 will recover l with high probability and therefore $\mathcal{A}$ returns 0 with high probability. When the challenge bit is $b = 1$ then $\mathcal{A}$ will get back $\mathrm{Enc}_k(\cdot; r) \circ \pi$ applied to the input register. In this case, by Corollary 2 a measurement on the input register remains indistinguishable for $\mathcal{A}$ (with $R := \mathrm{range}\ \mathrm{Enc}_k(\cdot; r)$ in Corollary 2). So we can assume the input register collapses to the classical message. Therefore $\mathcal{A}$ will recover l with negligible probability.

We show that Enc is secure in the learn($*$, ER)-chall($*$, CL, 1ct)-sense. Let G_b be the learn($*$, ER) -chall($*$, CL, 1ct)-IND-CPA game when the challenge bit is b. We show that G_0 and G_1 are indistinguishable. We define the game G' in which the challenge query will be answered with a random string and learning queries are answered with ER. We show that G_b is indistinguishable from G'. We can replace $\mathrm{sPRP}_k(l||s)$ with a random element in the challenge query. Since s is a fresh randomness in the challenge query by Lemma 6 $F_{s,l}(x_b)$ is indistinguishable from a random element. Finally, we can replace $\mathrm{qPRP}_l(z_b)$ with a random element. Therefore, games G_b and G' are indistinguishable. ☐

7.4 Separations by Shi's SetEquality problem

Definition 9 (SetEquality problem)**.** *The general* SetEquality *problem can be described as follows. Given oracle access to two injective functions*

$$f, g : \{0,1\}^m \to \{0,1\}^n$$

and the promise that

$$\mathrm{im}\, f = \mathrm{im}\, g \vee (\mathrm{im}\, f \cap \mathrm{im}\, g) = \varnothing)$$

decide which of the two holds. (Here the notation $\mathrm{im}\, f$ *means* $\{f(x) : x \in \{0,1\}^m\}$*)*

Here we will be consider the average-case problem, which involves *random* injective functions f and g. For SetEquality, the average-case and worst-case problem are equivalent: if we have an average-case distinguisher $\mathcal{D}$ then we can construct a worst-case-distinguisher by applying random permutations on the inputs and outputs of queries to f and g, which simulates an oracle for $\mathcal{D}$.

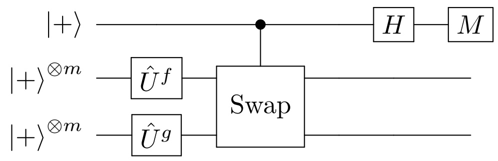
The SetEquality problem was first posed by Shi [Shi02] in the context of quantum query complexity. In [Zha15] it is proven that with ST-type-oracle access this problem is hard in m. However, a trivial implication of the swap-test shows that with ER-type oracle access it has constant complexity.

Lemma 8. *The* SetEquality *problem is indistinguishable under polynomial ST-type queries.*

Proof. This follows from Theorem 4 in [Zha15], which shows that $\Omega\left(2^{m/3}\right)$ ST-type queries are required to distinguish the two cases. ☐

Lemma 9. *The* SetEquality *problem is distinguishable under one ER-type query. That is, an adversary can, by only accessing f once and g once, decide whether they have equal or disjoint ranges with non-negligible probability.*

Proof. The attack works by a so-called swap-test, shown in the following circuit where the unitary control-Swap is defined as $\mathrm{cSwap} : |b, m_0, m_1\rangle \to |b, m_{b\oplus 0}, m_{b\oplus 1}\rangle$.



Let $|\Phi\rangle = 2^{-m/2}\sum_x |x\rangle$ and $|\phi_\mathcal{M}\rangle = \sum_x |\mathcal{M}(x)\rangle, \mathcal{M} \in \{f, g\}$, where the sums are over all $x \in \{0,1\}^m$. Then, up to normalization, the quantum circuit above implements the following:

$$\begin{aligned}|+\rangle|\Phi\rangle|\Phi\rangle &\overset{I\otimes\hat{U}^f\otimes\hat{U}^g}{\longmapsto} |+\rangle|\phi_f\rangle|\phi_g\rangle \\ &\overset{\text{cSwap}}{\mapsto} |0\rangle|\phi_f\rangle|\phi_g\rangle + |1\rangle|\phi_g\rangle|\phi_f\rangle \\ &\overset{H\otimes I}{\mapsto} |0\rangle\left(|\phi_f\rangle|\phi_g\rangle + |\phi_g\rangle|\phi_f\rangle\right) + |1\rangle\left(|\phi_f\rangle|\phi_g\rangle - |\phi_g\rangle|\phi_f\rangle\right)\end{aligned}$$

If the ranges of f and g are equal, then a measurement of the top qubit in the computational basis is guaranteed to yield 0. If the ranges are disjoint, then the measurement yields 0 or 1 with probability $\frac{1}{2}$. □

In order to apply the SetEquality problem to encryption schemes, we define constructions for f and g that use a random seed s.

Definition 10. *Let $\sigma_{s_1}, \sigma'_{s_2} : \{0,1\}^m \to \{0,1\}^m$ be qPRPs with seed s_1, s_2. Let J_{s_3}, J_{s_4} be a pseudorandom sparse injection built from a qPRP, i.e., for some qPRP $\tilde{J}_{s_3}, \tilde{J}_{s_4} : \{0,1\}^n \to \{0,1\}^n$, and any $x \in \{0,1\}^m$ with $n > m$, define $J_{s_3}(x) := \tilde{J}_{s_3}(x||0^{n-m})$ and $J_{s_4}(x) := \tilde{J}_{s_4}(x||0^{n-m})$. We can then define $F_{0,s_1,s_2,s_3}, G_{0,s_1,s_2,s_3} : \{0,1\}^m \to \{0,1\}^n$ to be a pair of pseudorandom sparse injections with equal range:*

$$F_{0,s_1,s_2} := J_{s_3} \circ \sigma_{s_1}, \qquad G_{0,s_1,s_2} := J_{s_4} \circ \sigma'_{s_2}.$$

Let $\tau_{s_5}, \tau_{s_6} : \{0,1\}^n \to \{0,1\}^n$ be a qPRP with seed s_5, s_6. Let $\tilde{K}_{s_7}, \tilde{K}'_{s_8} : \{0,1\}^m \to \{0,1\}^{n-1}$ be a pair of pseudorandom sparse injections, and define $K_{s_7} := 0||\tilde{K}_{s_7}, K'_{s_8} := 1||\tilde{K}'_{s_8}$. We can then define $F_{1,s'}, G_{1,s'} : \{0,1\}^m \to \{0,1\}^n$ (where $s' = (s_1, s_2, s_5, s_6, s_7, s_8)$) to be a pair of pseudorandom sparse injections with disjoint ranges:

$$F_{1,s_1,s_4,s_5} := \tau_{s_5} \circ K_{s_7} \circ \sigma_{s_1}, \qquad G_{1,s_1,s_4,s_5} := \tau_{s_6} \circ K'_{s_8} \circ \sigma'_{s_2}.$$

Let $s = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8)$. Note that $F_{b,s}$ and $G_{b,s}$ are decryptable using b, s.

Theorem 14. *If there exists a quantum secure one-way function then* learn$(*, ST)$-chall$(1, ST, \text{ror}) \not\Rightarrow$ learn$(*, ER)$-chall$(1, CL, \text{1ct})$ *in the quantum random oracle model. This shows that Panel 2 $\not\Rightarrow$ Panel 10.*

Proof. Let $H : \{0,1\}^h \to \{0,1\}^{|s|}$ be a random oracle. Let $sPRP$ be a standard secure pseudo random permutation with seed of length $|s|$. Let $\gamma_k(m_1||m_2; r, j) := F_{k_j,H(r)}(m_1)||G_{k_j,H(r)}(m_2)$ where k_j is j-th bit of k. Consider the encryption function

$$\text{Enc}_k(m_1||m_2; r, \tilde{r}, j) := \gamma_k(qPRP_r(m_1||m_2); r, j)||sPRP_{H(k)}(r)||j, \tag{1}$$

where $qPRP_r$ is a quantum secure pseudo random permutation with seed r. The encryption scheme above is decryptable as follows. First one can decrypt $sPRP_{H(k)}$ using the random oracle H and the secret key k and then decrypt the other part of the ciphertext using j, r, the secret key k and the random oracle H. We show that the above encryption scheme is learn$(*, ST)$-chall$(1, ST, \text{ror})$ secure. Let $\mathcal{A}$ be an adversary that attacks in the sense of learn$(*, ST)$-chall$(1, ST, \text{ror})$ IND-CPA. In the following, we abuse the notation and use $\pi(qPRP_r(m_1))_1, \pi(qPRP_r(m_2))_2$ to indicate the first m bits and the second m bits of $\pi(qPRP_r(m_1||m_2))$, respectively. The challenge query submitted by the adversary is two registers Q_{in} and Q_{out} that may contain superposition of many $|m_1, m_2\rangle_{Q_{in}}|y\rangle_{Q_{out}}$ basis states $(Q_{in}Q_{out} : \sum_{m_1,m_2,y} \alpha_{m_1,m_2,y}|m_1, m_2\rangle|y\rangle)$. For simplicity, we only show one of the computational basis states in the presentation of the games and with linearity of U_{Enc} it will be similar for the rest.

Game 0 : learn$(*, ST)$-chall$(*, ST, \text{ror})$ IND-CPA

$$\left[\begin{array}{l} k \overset{\$}{\leftarrow} \{0,1\}^h, b \overset{\$}{\leftarrow} \{0,1\}, \pi \overset{\$}{\leftarrow} (\{0,1\}^{2m+1} \to \{0,1\}^{2m+1}), r \overset{\$}{\leftarrow} \{0,1\}^t, j \overset{\$}{\leftarrow} \{1, \cdots, n\}, \\ |m_1, m_2\rangle|y\rangle \leftarrow \mathcal{A}^{H,\text{Enc}}(), \\ c := |m_1, m_2\rangle\big|y \oplus (F_{k_j,H(r)}(\pi^b(qPRP_r(m_1))_1), G_{k_j,H(r)}(\pi^b(qPRP_r(m_2))_2, sPRP_{H(k)}(r), j))\big\rangle \\ b' \leftarrow \mathcal{A}^{H,\text{Enc}}(c), \\ \textbf{return } [b = b'] \end{array}\right.$$

Let $\{r, r_2, \cdots, r_q\}$ is the set of all randomness used in the learning queries and the challenge query in γ part of encryption. In the following game we replace $H(k)$, $H(r)$, $H(r_2)$ $\cdots$, $H(r_q)$ with random values in the learning queries and challenge queries. We call this Game 1 and in the presentation below we only show the replacement in the challenge query. The same replacement will occur in all learning queries.

$$\begin{aligned}&Game\ 1:\\&\left[\begin{array}{l}k \stackrel{\$}{\leftarrow} \{0,1\}^h, b \stackrel{\$}{\leftarrow} \{0,1\}, \pi \stackrel{\$}{\leftarrow} (\{0,1\}^{2m+1} \to \{0,1\}^{2m+1}), r \stackrel{\$}{\leftarrow} \{0,1\}^t, j \stackrel{\$}{\leftarrow} \{1,\cdots,n\}, r^*, k^* \stackrel{\$}{\leftarrow} \{0,1\}^{|s|}\\|m_1,m_2\rangle|y\rangle \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(),\\c := |m_1,m_2\rangle \big| y \oplus (F_{k_j,r^*}(\pi^b(qPRP_r(m_1))_1), G_{k_j,r^*}(\pi^b(qPRP_r(m_2))_2, sPRP_{k^*}(r), j))\big\rangle\\b' \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(c),\\\mathbf{return}\ [b = b']\ .\end{array}\right.\end{aligned}$$

In order to show that Game 0 and Game 1 are indistinguishable, we use Theorem 3 in [AHU18]. Let q be the total number of queries to the random oracle H. By Theorem 3 in [AHU18], there exists a polynomial time adversary $\mathcal{B}$ that returns the output x such that

$$|\Pr[1 \leftarrow Game\ 0] - \Pr[1 \leftarrow Game\ 1]| \leq \sqrt{q \Pr[x \in \{r, r_2, \cdots, r_q, k\} : Game\ 2]}$$

where Game 2 is defined as below (with randomness $r^*, r_2^*, \cdots, r_q^*$ and random key k^*):

$$\begin{aligned}&Game\ 2:\\&\left[\begin{array}{l}k \stackrel{\$}{\leftarrow} \{0,1\}^h, b \stackrel{\$}{\leftarrow} \{0,1\}, \pi \stackrel{\$}{\leftarrow} (\{0,1\}^{2m+1} \to \{0,1\}^{2m+1}), r \stackrel{\$}{\leftarrow} \{0,1\}^t, j \stackrel{\$}{\leftarrow} \{1,\cdots,n\}, r^*, k^* \stackrel{\$}{\leftarrow} \{0,1\}^{|s|}\\|m_1,m_2\rangle|y\rangle \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(),\\c := |m_1,m_2\rangle \big| y \oplus (F_{k_j,r^*}(\pi^b(qPRP_r(m_1))_1), G_{k_j,r^*}(\pi^b(qPRP_r(m_2))_2, sPRP_{k^*}(r), j))\big\rangle\\b' \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(c),\\x \leftarrow \mathcal{B}^{H,\mathrm{Enc}}(c).\end{array}\right.\end{aligned}$$

Let F_0^* and G_0^* be random injection functions with equal ranges. Let F_1^* and G_1^* be random injection functions with disjoint ranges. Note that since r^* is a fresh randomness by construction of F_{k_j,r^*} and G_{k_j,r^*} in Definition 10 they are indistinguishable from F_0^* and G_0^* when $k_j = 0$ and they are indistinguishable from F_1^* and G_1^* when $k_j = 1$. Next, we replace $F_{k_j r^*}$ and $G_{k_j r^*}$ with F_1^* and G_1^* respectively in the challenge query in Game 2. Note that the same argument holds for the learning queries and we replace all F_{k_j,r_i^*} and G_{k_j,r_i^*} with independent random injective functions $F_1^{(i)}$ and $G_1^{(i)}$ with disjoint ranges. Let call the modified game Game 2a. Note that two games are indistinguishable because the set-equality problem is hard for ST-type queries by Lemma 8.

$$\begin{aligned}&Game\ 2a:\\&\left[\begin{array}{l}k \stackrel{\$}{\leftarrow} \{0,1\}^h, b \stackrel{\$}{\leftarrow} \{0,1\}, \pi \stackrel{\$}{\leftarrow} (\{0,1\}^{2m+1} \to \{0,1\}^{2m+1}), r \stackrel{\$}{\leftarrow} \{0,1\}^t, j \stackrel{\$}{\leftarrow} \{1,\cdots,n\}, r^*, k^* \stackrel{\$}{\leftarrow} \{0,1\}^{|s|}\\|m_1,m_2\rangle|y\rangle \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(),\\c := |m_1,m_2\rangle \big| y \oplus (F_1^*(\pi^b(qPRP_r(m_1))_1), G_1^*(\pi^b(qPRP_r(m_2))_2, sPRP_{k^*}(r), j))\big\rangle\\b' \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(c),\\x \leftarrow \mathcal{B}^{H,\mathrm{Enc}}(c).\end{array}\right.\end{aligned}$$

Since in each query a fresh randomness will be encrypted by $sPRP_{k^*}$, we can replace $sPRP_{k^*}(randomness)$ with random values in Game 2b. Next, in Game 2c we can replace $qPRP$ with a independent random permutation in each query because the seed of $qPRP$ is chosen independently at random in each query and it is not used elsewhere in Game 2b. It is clear that the success probability of Game 2c is $(q+1)/2^h$ because $k,\ r,\ r_2, \cdots, r_q$ has not been used in Game 2c. Now we show that the success probability in Game 1 is $1/2 + neg$. We can do similar modification presented above to define Game 1a. So in each query, two random injective functions with disjoint ranges will be used. Next we define Game 1b in which we replace $sPRP_{k^*}(r)$ with a random value α^* in the challenge query. This can be done since r is a fresh randomness and $sPRP$ is a standard secure pseudo random permutation. Finally, we replace $qPRP_r$ with a random permutation π' in the challenge query in Game 1c. This can be done because r is a fresh randomness that has been used only as seed of $qRPR$ in Game 1b.

$$\begin{aligned}&Game\ 1c:\\&\left[\begin{array}{l}k \stackrel{\$}{\leftarrow} \{0,1\}^h, b \stackrel{\$}{\leftarrow} \{0,1\}, \pi \stackrel{\$}{\leftarrow} (\{0,1\}^{2m+1} \to \{0,1\}^{2m+1}), r \stackrel{\$}{\leftarrow} \{0,1\}^t, j \stackrel{\$}{\leftarrow} \{1,\cdots,n\}, r^*, k^* \stackrel{\$}{\leftarrow} \{0,1\}^{|s|}\\|m_1,m_2\rangle|y\rangle \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(),\\c := |m_1,m_2\rangle \big| y \oplus (F_1^*(\pi^b(\pi'(m_1))_1), G_1^*(\pi^b(\pi'(m_2))_2, \alpha^*)\big\rangle\\b' \leftarrow \mathcal{A}^{H,\mathrm{Enc}}(c),\\\mathbf{return}\ [b = b']\ .\end{array}\right.\end{aligned}$$

It is clear that the success probability of Game 1c is $1/2 + neg$. Overall, we showed that the success probability of Game 1 is $1/2 + neg$ and this finishes the security proof.

Now we show that Enc can be broken in learn($*$, ER)-chall(1, CL, 1ct). Let $\mathcal{A}'^{\text{Enc}}$ denote the adversary that plays the learn($*$, ER)-chall(1, CL, 1ct)-IND-CPA game. By Lemma 9, it is possible for $\mathcal{A}'^{\text{Enc}}$ to perform a learn($*$, ER)-learning-query for $m \leftarrow |+\rangle^{\otimes m}|+\rangle^{\otimes m}$ and conduct a swap-test to determine k_j with high probability for a random j (Note that j is the last part of ciphertext and is known to the adversary). The procedure is repeated polynomially many times until $\mathcal{A}'^{\text{Enc}}$ has enough information about the key k to guess it with sufficiently high probability. Finally, $\mathcal{A}'^{\text{Enc}}$ can choose any two classical messages m_0, m_1 for challenge query, and use the private key k to decrypt the result and determine the challenge bit b. □

7.5 Separations by other arguments

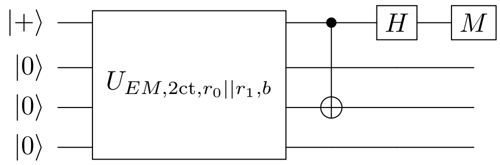
Theorem 15. *On the existence of a quantum secure one-way function, the following separation holds:*

1. learn($*$, ST)-chall($*$, CL, 1ct), learn($*$, ER)-chall($*$, CL, 1ct) $\not\Longrightarrow$ learn($*$, CL)-chall(1, EM, 2ct)
   P6, P10 $\not\Longrightarrow$ *P7*

Proof. Consider

$$\text{Enc}_k(m; r) = r||\text{PRP}_k(r) \oplus m \text{ for } m, r \in \{0,1\}^n$$

where PRP is a standard secure pseudorandom permutation. The security in learn($*$, ST)-chall($*$, CL, 1ct) and learn($*$, ER)-chall($*$, CL, 1ct) senses follows by Lemma 3 in [ATTU16]. We show the insecurity using a challenge query of type chall(1, EM, 2ct). The attack is described by the following quantum circuit. For simplicity, we omit the wires corresponding to the r-parts of two ciphertexts.



When $b = 0$, the measurement returns 0 with probability 1 and it outputs 0 only with negligible probability when $b = 1$. □

Theorem 16. *On the existence of a quantum secure one-way function,* learn($*$, ST)-chall($*$, CL, 1ct) $\not\Longrightarrow$ learn($*$, CL)-chall(1, EM, 1ct). *This shows that P6* $\not\Longrightarrow$ *P13.*

Proof. Consider

$$\text{Enc}_k(m; r) = r||\text{PRP}_k(r) \oplus m \text{ for } m, r \in \{0,1\}^n$$

where PRP is a standard secure pseudorandom permutation. The security in learn($*$, ST)-chall($*$, CL, 1ct) and learn($*$, ER)-chall($*$, CL, 1ct) senses follows by Lemma 3 in [ATTU16]. The insecurity follows from Lemma 10 in [CEV20]. □

Theorem 17. *On the existence of quantum secure one-way function,* learn($*$, ER)-chall(1, ER, 1ct) $\not\Longrightarrow$ learn($*$, CL)-chall(1, ST, ror). *This shows that P1* $\not\Longrightarrow$ *P12.*

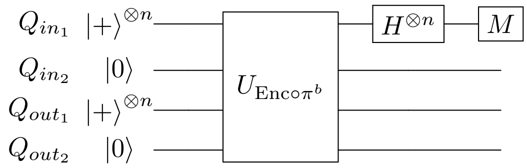
Proof. Let $qPRP$ and $qPRP'$ be two quantum secure pseudo random permutations with input/output $\{0,1\}^{2n}$. Let $sPRP$ be a standard secure pseudo random permutation. For m_1 and m_2 of length n-bits, we define Enc as following:

$$\text{Enc}_k(m_1, m_2; r_1, r_2) = qPRP_{r_1}(0^n||m_1)||qPRP_{r_2}(0^n||m_2)||sPRP_k(r_1, r_2).$$

First, we prove that Enc is learn($*$, ER)-chall(1, ER, 1ct) secure. Note that in each query we can replace $sPRP_k(r_1, r_2)$ with a random value because r_1 and r_2 are fresh randomness and $sPRP$ is a standard secure pseudo random permutation. Then in each query we can replace $qPRP_{r_1}$ and $qPRP_{r_2}$ with random permutations π_1 and π_2, respectively. Now we can measure the input register by Lemma 3 and the security follows from learn($*$, CL)-chall(1, CL, 1ct) security of Enc.
Now we show that Enc is not secure respected to learn($*$, CL)-chall(1, ST, ror) notion. Let Q_{in_1} and Q_{in_2} be input registers corresponding to first n bits and second n bits of message, respectively. Similarly, Q_{in_1} and Q_{in_2} be output registers. The adversary can query

$$Q_{in_1}Q_{in_2}Q_{out_1}Q_{out_2} := |+\rangle^{\otimes n}|0\rangle^{\otimes n}|+\rangle^{\otimes n}|0\rangle^{\otimes n}$$

in the challenge query. After receiving the answer, it applies the Hadamard operator to Q_{in_1} then measures the register in the computational basis:



When $b = 0$ the measurement returns 0 with probability 1. In other hand, when $b = 1$ the permutation will be applied to the message and therefore Q_{in_1} register will be entangled with output registers. In this case, the measurement returns 0 with negligible probability. □

Conjecture 1. *We conjecture that the following non-implications hold.*
(P2 $\not\Rightarrow$ P9), (P3 $\not\Rightarrow$ P11), (P4 $\not\Rightarrow$ P6), (P8 $\not\Rightarrow$ P7), (P12 $\not\Rightarrow$ P11, P7).

8 Encryption secure in all notions

In this section we propose an encryption function that is secure for all security notions described in this paper. From Figure 1, Panel 1 and Panel 2 imply all other panels. Therefore it is sufficient to construct an encryption function that is secure in a setting where there are no learning queries, and where the challenge queries are either $\mathfrak{c}_1 = \mathrm{chall}(*, ER, 1\mathrm{ct})$ or $\mathfrak{c}_2 = \mathrm{chall}(*, ST, \mathrm{ror})$. Consider the encryption scheme Enc as $\mathrm{Enc}_k(m; r, r') = qPRP_r(r'||m)||sPRP_k(r)$ for $r', m \in \{0, 1\}^n$. In order to decrypt the ciphertext, first we decrypt $sPRP_k(r)$ using the secret key k and obtain r then we can obtain the message m using r. Now we show that Enc is $\mathfrak{c}_1 = \mathrm{chall}(*, ER, 1\mathrm{ct})$ and $\mathfrak{c}_2 = \mathrm{chall}(*, ST, \mathrm{ror})$ secure in the following:

Theorem 18. *The encryption scheme* $\mathrm{Enc}_k(m; r, r') = qPRP_r(r'||m)||sPRP_k(r)$ *presented above is* $\mathrm{chall}(*, ER, 1\mathrm{ct})$ *and* $\mathrm{chall}(*, ST, \mathrm{ror})$ *secure.*

Proof. $\mathrm{chall}(*, ER, 1\mathrm{ct})$ **security:** In each query we can replace $sPRP_k(r)$ with a random bit string because r is a fresh randomness and $sPRP$ is a standard secure pseudo random function. Now we can replace $qPRP_r$ with a random permutation π' in each query and use Lemma 3 to measure the input register (with $f := \pi'(r'||\cdot)$). This collapses to the security against $\mathrm{chall}(*, CL, 1\mathrm{ct})$ queries that it is trivial.

$\mathrm{chall}(*, ST, \mathrm{ror})$ **security:** In each query we can replace $sPRP_k(r)$ with a random bit string because r is a fresh randomness and $sPRP$ is a standard secure pseudo random function. Then we can replace $qPRP_r$ with a random permutation π' in each query. The security is trivial because for a random r', $f_1(m) = \pi'(r'||m))$ (when the challenge bit is 0) and $f_2(m) = \pi'(r'||\pi(m))$ (when the challenge bit is 1) have the same distribution . □

References

[AHU18] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. *IACR Cryptology ePrint Archive*, 2018:904, 2018.

[AMRS18] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-secure message authentication via blind-unforgeability. *IACR Cryptology ePrint Archive*, 2018:1150, 2018.

[ATTU16] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the cbc, cfb, ofb, ctr, and XTS modes of operation. In Tsuyoshi 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 44–63. Springer, 2016.

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 394–403. IEEE Computer Society, 1997.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 609–629. Springer, 2015.

[BZ13a] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608. Springer, 2013.

[BZ13b] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a post-quantum world. *IACR Cryptology ePrint Archive*, 2013:88, 2013.

[CEV20] Céline Chevalier, Ehsan Ebrahimi, and Quoc Huy Vu. On the security notions for encryption in a quantum world. *IACR Cryptology ePrint Archive*, 2020:237, 2020.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 60–89. Springer, 2016.

[GKS20] Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. Make quantum indistinguishability great again. *IACR Cryptology ePrint Archive*, 2020:266, 2020.

[KKVB02] Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Phys. Rev. A*, 65:050304, May 2002.

[MS16] Shahram Mossayebi and Rüdiger Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *CoRR*, abs/1609.03780, 2016.

[NC16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.

[Shi02] Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 513–519, 2002.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, October 1997.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 679–687. IEEE Computer Society, 2012.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.

[Zha16] Mark Zhandry. A note on quantum-secure prps. *IACR Cryptology ePrint Archive*, 2016:1076, 2016.