Relationships between quantum IND-CPA notions

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

[1] University of Tartu, Estonia
[2] FSTM & SnT, University of Luxembourg
[3] National Tsing Hua University & National Cheng Kung University, Taiwan

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. There are other variants of this definition but they all turn out to be equivalent in the classical case. In this paper, we give a comprehensive overview of these different variants of IND-CPA for symmetric encryption schemes in the quantum setting. We investigate the relationships between these notions and prove various equivalences, implications, non-equivalences, and non-implications between these variants.

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

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 upon. There has been extensive research toward proposing quantum counterparts of classical security definitions for different cryptographic primitives: encryption schemes [6,12,9], message authentication codes [5,1], hash functions [26,24], 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 *quantum queries a.k.a. superposition queries* to the cyptographic primitives. In this paper, we focus on quantum versions of indistinguishability under chosen plaintext attack for *symmetric* encryption schemes. There are some proposals for a quantum IND-CPA notion in the literature [6,12,19] (see Section 1.1 for more details). However, there are a number of design choices (e.g., how queries are performed, when they are classical, etc.) in those works, each work considers different combinations of those design decisions, and the choice which combinations are investigated and which are not is somewhat ad-hoc. In addition, it was not known (prior to our work) how the different definitions relate to each other, or whether they are even all

equivalent. (The latter would show that the design choices are in fact irrelevant, but unfortunately we find that this is not the case.) The aim of our work is to comprehensively study the resulting variants of the IND-CPA definition and the relationship (implication/equivalence/non-implication) between them.

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 encryptions of m_b, $m_{\bar{b}}$ for a random bit b.
(c) The adversary chooses a message m and sends it to the challenger. The challenger will send back either the encryption of m or a randomly chosen message depending 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. (We use the abbreviation "1ct".) b) The challenger returns two ciphertexts. (We use the abbreviation "2ct".) c) The challenger returns a real or random ciphertext. (We use the abbreviation "ror".) A comprehensive study of these notions has been done in [4] in the classical setting and it turns out these notions are equivalent up to a polynomial loss in the reductions. (The notion 2ct has not been studied in [4], however, it is easy to see that 1ct and 2ct are equivalent in the classical setting.)

In addition, there are different kinds of quantum queries, differing in what registers are returned or discarded or used as input/output. (We make the different possibilities more explicit in the following.) This distinction has no counterpart in the classical setting.

In the following, we present existing quantum IND-CPA notions in the literature [6,12,19]. We make the type of quantum query and the return type (1ct, 2ct or ror) in the definitions explicit.

1.1 Previous works

Boneh-Zhandry definition. In [6], Boneh and Zhandry initiate developing a quantum security version of IND-CPA. They consider that the adversary has "standard oracle access" (ST) to the encryption oracle in the learning phase. The standard oracle access to the encryption oracle Enc is defined as

the unitary operator $U_{\mathrm{Enc}} : |x, y\rangle \rightarrow |x, y \oplus \mathrm{Enc}(x)\rangle$ (see Section 3). For the challenge phase, they attempt to translate the classical notion of one-ciphertext and two-ciphertext return types (presented in (a) and (b) above) to the quantum case using the standard query model. However, they show that the natural translation leads to an impossible notion of IND-CPA. So instead they consider classical challenge queries in their proposed definition combined with standard quantum queries in the learning phase. This inconsistency between the learning phase and the challenge phase resulted in further investigation of the quantum IND-CPA notion in [12].

Quantum IND-CPA notions in [12]. In [12], the authors attempt to resolve the inconsistency of the learning and the challenge phase of the security definition proposed in [6]. 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 adversary sends the challenge queries: the adversary sends quantum messages during the challenge phase or it sends a classical description of quantum messages; 2) whether the challenger sends back the input registers to the adversary or keeps them; and 3) the query model: the adversary has standard oracle access to the challenger or it has "minimal oracle" access [15] (that is defined as $|x\rangle \rightarrow |\mathrm{Enc}(x)\rangle$, called the "erasing query model" in this work).[4] Even though in total there are $2^3 = 8$ possible security definitions, only two are investigated in [12]. These two definitions are (according to their terminology briefed above): 1) quantum messages, not returning the input register and minimal oracle access [5]. 2) classical description of 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. In this paper, we refer to the minimal query model as the "erasing query model" (*ER*) (see Section 3).

Quantum IND-CPA notion in [19]. In [19], Mossayebi and Schack focus on translating the real-or-random case (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

[4] They additionally distinguish between what they call the "oracle model" and the "challenger model" queries. The difference is that in the "oracle model", only unitary query oracles are allowed, while in the "challenger model", query oracles are allowed that, e.g., erase register. The security definitions that can be expressed in the "challenger model" trivially subsume those that can be stated in the "oracle model". So the distinction has no effect on the set of possible security definitions. (In fact [12] never formally defines the distinction.)

[5] This security definition is equivalent to the indistinguishability notion proposed in [7] for secret key encryption of quantum messages when restricted to a classical encryption function operating in the minimal query type.

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 learning queries but the adversary can perform many challenge queries. The adversary has the standard oracle access to the challenger and the challenge phase is implemented by the real-or-random return type.

Therefore, there are three achievable definitions for quantum IND-CPA notion in the literature so far. These three notions only cover a small part of the different combinations of the design choices made in those papers – the query models (classical, *ST*, and *ER* etc.), the challenge return type (1ct, 2ct, and ror), the number of queries (none, one, many) – even if we only consider different combinations of the design decisions already made in those papers. The choice which combinations are considered seems ad-hoc (in the sense that there is no systematic consideration of other combinations), and the combinations actually matter (different from the classical setting where we tend to arrive at the same notion of IND-CPA in many different ways).

In this paper, our aim is to answer the following questions:

What is a comprehensive list of distinct possible quantum IND-CPA notions?
How do these notions relate to each other?
Which one is the strongest (achievable) security notion?

Why should we care? Encryption schemes (and other cryptographic primitives) secure under quantum queries (a.k.a. superposition queries) have been studied in prior work from a number of angles, e.g., [16,17,6,5,10,14,19,12,3,18,11]. There are two main reasons for studying them: The fear that future cryptographic devices will be quantum and will therefore either intentionally or due to manipulation by the adversary perform encryption and similar operations in superposition. And the fact that in security proofs, intermediate games may involve oracles that answer quantum queries even if the original games were purely classical.[6] While these reasons give motivation for studying quantum queries, they do not answer the question which model is the right one, and which security definition is the right one. While we cannot give a definitive answer which definition is right (although we can answer, e.g., which is strongest), we do clarify which options there are, and how they relate (at least in the case of IND-CPA security of symmetric encryption). And by showing equivalences, we also narrow down the field to a more manageable number of choices (namely 14 instead of 72). This enables designers of symmetric encryption schemes or modes of operations to know which security notions can be or need to be considered (e.g., they could simply show security with respect to the strongest ones). It provides guidance to cryptographers using symmetric encryption as subprotocols what options there are to make the proofs go through, and

[6] For example, in a post-quantum security proof involving quantum rewinding [25,23], the adversary (including any oracles it queries) is first transformed into a unitary operation. As a side effect, any classical oracle would also be transformed into a unitary one.

it provides foundational insight into the structure of security definitions, and tells us which design choice does or does not matter. We note that it is very easy to get misled here by one's intuition, and to assume relationships between the notions that are not correct. For example, [12] mistakenly states that that the security notion based on erasing queries ER are stronger than those based on standard or embedding queries ST and then restricts their attention only to ER queries because this supposedly leads to the strongest result.[7] To the best of our knowledge, this claim has not been disputed so far. Our results show that this is not correct and the notions are actually incomparable. Last but not least, understanding IND-CPA with quantum queries is an important first step towards finding good notion for IND-CCA with quantum queries. The latter is a hard problem with partial success [13,9] that has so far eluded a definitive answer.

1.2 Our contribution

We study all possible quantum IND-CPA security notions. We classify the notions according to the following criteria:

(1) Number of queries that the adversary can make during the learning and challenge phase: zero (0), one (1) or many ($*$) queries. Note that in the learning phase either there are no queries or many queries, while in the challenge phase there is one query or many queries.

(2) Query model in which the adversary is interacting with the challenger: classical (CL), standard (ST), erasing (ER), or "embedding query model" (EM). 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: 1ct, 2ct, or ror.

This gives 5 choices for the learning phase and 24 choices for the challenge phase. Therefore, there are 120 variants of the security notion altogether. We use the notation learn(?, ?)-chall(?, ?, ?) for the security notions where the question marks are identified from the choices above. For instance, Boneh-Zhandry definition [6] can be represented with learn($*, ST$)-chall($1, CL,$ 1ct) which means many ST queries in the learning phase and one classical challenge query, both returning one ciphertext.

Excluded security notions. We do not consider security notions with different quantum query models in the learning phase and the challenge phase. E.g., ST challenge queries with ER learning queries. While technically possible, we consider such combinations to be too "exotic" and do not expect them to be used.[8] (Classical queries can be combined with any of quantum query models

[7] Their precise wording is "*we will focus on the (. . . 2) models in order to be on the 'safe side', as they lead to security notions which are harder to achieve.*". In their language, type-(2) models correspond to our ER queries, and type-(1) models to our ST queries.

[8] This is, of course, arguable. But without this restriction, the number of possible combinations would grow beyond what is manageable in the scope of this paper.

though. E.g., the Boneh-Zhandry definition [6] is of this type.) Also, we do not consider a security notion with no learning queries and only one challenge query since this corresponds to the IND-OT-CPA notion (one-time IND-CPA security) that will not be considered in this paper. This leaves us with 72 notions.

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 achieve by any encryption scheme [6]. Any query model with the embedding query type EM and the one-ciphertext return type in the challenge phase is impossible to achieve. (See Section 5).

Impossible security notions

learn$(0, -)$-chall$(*, ST, 1\text{ct})$,	learn$(0, -)$-chall$(*, ST, 2\text{ct})$,
learn$(*, CL)$-chall$(1, ST, 1\text{ct})$,	learn$(*, CL)$-chall$(1, ST, 2\text{ct})$,
learn$(*, CL)$-chall$(*, ST, 1\text{ct})$,	learn$(*, CL)$-chall$(*, ST, 2\text{ct})$,
learn$(*, ST)$-chall$(1, ST, 1\text{ct})$,	learn$(*, ST)$-chall$(1, ST, 2\text{ct})$,
learn$(*, ST)$-chall$(*, ST, 1\text{ct})$,	learn$(*, ST)$-chall$(*, ST, 2\text{ct})$,
learn$(0, -)$-chall$(*, EM, 1\text{ct})$,	learn$(*, CL)$-chall$(1, EM, 1\text{ct})$,
learn$(*, CL)$-chall$(*, EM, 1\text{ct})$,	learn$(*, EM)$-chall$(1, EM, 1\text{ct})$,
learn$(*, EM)$-chall$(*, EM, 1\text{ct})$	

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

Security notions that are equivalent (see Section 6): The definitions inside each box are equivalent.

Panel 1

learn$(0, -)$-chall$(*, ER, 1\text{ct})$,	learn$(0, -)$-chall$(*, ER, 2\text{ct})$,
learn$(*, CL)$-chall$(*, ER, 1\text{ct})$,	learn$(*, CL)$-chall$(*, ER, 2\text{ct})$,
learn$(*, ER)$-chall$(1, ER, 1\text{ct})$,	learn$(*, ER)$-chall$(1, ER, 2\text{ct})$,
learn$(*, ER)$-chall$(*, ER, 1\text{ct})$,	learn$(*, ER)$-chall$(*, ER, 2\text{ct})$

Note that Panel 1 includes the security notion from [12]. These equivalences have been achieved by Theorem 15, Theorem 17 and Theorem 20.

Panel 2

learn$(0, -)$-chall$(*, ST, \text{ror})$,	learn$(*, CL)$-chall$(*, ST, \text{ror})$,
learn$(*, ST)$-chall$(1, ST, \text{ror})$,	learn$(*, ST)$-chall$(*, ST, \text{ror})$

Note that Panel 2 includes the security notion from [19]. These equivalences have been obtained by Theorem 15 and Theorem 18.

Panel 3

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

Panel 4

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

The equivalences in Panel 4 have been concluded by Theorem 15 and Theorem 18.

Panel 5

learn$(0, -)$-chall$(*, EM, \text{ror})$, learn$(0, -)$-chall$(*, EM, \text{2ct})$,
learn$(*, CL)$-chall$(*, EM, \text{ror})$, learn$(*, CL)$-chall$(*, EM, \text{2ct})$,
learn$(*, EM)$-chall$(1, EM, \text{2ct})$, learn$(*, EM)$-chall$(1, EM, \text{2ct})$,
learn$(*, EM)$-chall$(*, EM, \text{2ct})$, learn$(*, EM)$-chall$(*, EM, \text{2ct})$

We can conclude the equivalences in Panel 5 by Theorem 15, Theorem 17, Theorem 19, Theorem 18, and Theorem 22.

Panel 6

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

Note that this panel includes the security notion from [6]. We can conclude these equivalences by Theorem 15 and Theorem 16.

Panel 7

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

Panel 8

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

Panel 9

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

Panel 10

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

We can conclude the equivalences in Panel 10 by Theorem 15 and Theorem 16.

Panel 11

learn($*$, ER)-chall(1, CL, 1ct),	learn($*$, ER)-chall(1, CL, 2ct),
learn($*$, ER)-chall(1, CL, ror),	learn($*$, ER)-chall($*$, CL, 1ct),
learn($*$, ER)-chall($*$, CL, 2ct),	learn($*$, ER)-chall($*$, CL, ror)

Panel 12

learn($*$, CL)-chall(1, ST, ror)

Panel 13

learn($*$, CL)-chall(1, EM, ror)

Panel 14

learn(0, $-$)-chall($*$, CL, ror),	learn(0, $-$)-chall($*$, CL, 1ct),
learn(0, $-$)-chall($*$, CL, 2ct),	learn($*$, CL)-chall($*$, CL, ror),
learn($*$, CL)-chall($*$, CL, 1ct),	learn($*$, CL)-chall($*$, CL, 2ct),
learn($*$, CL)-chall(1, CL, ror),	learn($*$, CL)-chall(1, CL, 1ct),
learn($*$, CL)-chall(1, CL, 2ct)	

We can conclude the equivalences in Panel 14 by Theorem 15 and Theorem 16.

Main Conclusion. We observe that different from the the classical case in which IND-CPA notions with different types of challenge queries (1ct, 2ct or ror) are equivalent (see Panel 14), when the challenge query is quantum (ST, EM or ER), the notions are not equivalent. More specifically: 1) for the standard query model, only the real-or-random return type is achievable (and two others are impossible to achieve). 2) for the embedding query model, the one-ciphertext return type is impossible to achieve, however, other two cases are equivalent (see Panel 5). 3) for the erasing query model, the one-ciphertext and two-ciphertexts 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.)

Implications and non-implications (Section 6 and Section 7). The implications and separation have been drawn in Table 1. The cells with a question mark remain open questions. We conclude that a notion P does not imply Q if there exists an encryption scheme that is secure with respect to the notion P and insecure with respect to the notion Q. All of non-implications hold on the assumption of the existence of a quantum secure one-way function. They all hold in the standard model except the non-implication in the Theorem 38 that holds in the quantum random oracle model.

Main conclusions of Table 1.

- Panels P1 and P2 together imply all other security notions. We present an encryption scheme that is secure in the sense of the notions in Panels 1 and 2 (see Section 8), and therefore it is secure with respect to all notions.
- Panel 1 and 2 are not comparable to each other. This resolves an open question stated in [19,13] for a comparison between these security notions.

Decoherence Lemmas: As a technical tool, we introduce several "decoherence lemmas". Essentially, a decoherence lemma states that a certain randomized query effectively measures the input of that query (even if the query is actually performed in superposition). Specifically, we show that a query to a random sparse injective function in the erasing query model ER will effectively measure its input (even if no register is actually measured or erased). And we show an analogous result for the embedding query model EM and a random function (see Section 4). These decoherence lemmas make it much easier to compare different query models because we can use them to prove that the queries are essentially classical. They are an essential tool in our analysis, both for showing implications and separations. However, we believe that they are a tool of independent interest for the analysis of superposition queries in cryptographic settings.

Simulating learning queries with challenge queries. Classically, it is easy to see that one can simulate the learning queries with the challenge queries. For instance, for the return types of 1ct, 2ct, the reduction makes a copy of the learning query and sends the query along with its copy to the challenger and forwards back the ciphertext (for 1ct) or one of the ciphertexts (for 2ct) to the adversary. But when the queries are quantum, this approach will not work due to no-cloning theorem. We resolve this obstacle and show that the simulation of learning queries using challenge queries is possible in the quantum setting as well (see Theorem 17 and Theorem 18.).

Impossibility results for natural modes of operation. We show (Corollary 42) that any out of a large class of modes of operation is insecure with respect to challenge queries of type (ST, ror). Basically, this includes all modes of operation where at least one output block is not dependent on all input blocks. While we do propose an encryption scheme that is secure with respect to all (achievable) notions presented in this work, an efficient mode of operation with this property is an open problem. Corollary 42 gives an indication why this is the case. (Modes of operation have been studied with respect to the Boneh-Zhandry's definition in [3].)

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

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
P1		$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow^{21}$	$\Rightarrow$	$\Rightarrow^{23}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow^{41}$	$\Rightarrow$	$\Rightarrow$
P2	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow^{28}$	$\not\Rightarrow^{?}$	$\not\Rightarrow^{38}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P3	?	$\not\Rightarrow$		?	?	?	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	?	$\not\Rightarrow^{?}$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P4	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\Rightarrow$	$\not\Rightarrow^{?}$	$\Rightarrow$	$\not\Rightarrow^{28}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P5	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		?	$\Rightarrow$	$\not\Rightarrow$	?	$\not\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P6	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow^{39}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{40}$	$\Rightarrow$
P7	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	?	?		$\not\Rightarrow$	?	$\not\Rightarrow$	?	$\not\Rightarrow$	$\Rightarrow^{22}$	$\Rightarrow$
P8	?	$\not\Rightarrow$	?	?	?	?	$\not\Rightarrow^{?}$		$\Rightarrow^{21}$	?	?	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P9	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	?	?	?	?	$\not\Rightarrow$		?	?	$\not\Rightarrow$	$\Rightarrow$	$\Rightarrow$
P10	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	?	$\not\Rightarrow^{39}$	$\not\Rightarrow$	$\not\Rightarrow$		$\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{33}$	$\Rightarrow$
P11	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	?	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$		$\not\Rightarrow$	$\not\Rightarrow$	$\Rightarrow$
P12	$\not\Rightarrow$	?	$\not\Rightarrow$	$\not\Rightarrow$	?	?	$\not\Rightarrow^{?}$	$\not\Rightarrow$	?	$\not\Rightarrow$	$\not\Rightarrow^{?}$		$\Rightarrow$	$\Rightarrow$
P13	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	?	?	?	$\not\Rightarrow$	?	$\not\Rightarrow$	?	$\not\Rightarrow$		$\Rightarrow$
P14	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{32}$	$\not\Rightarrow$	$\not\Rightarrow$	

Table 1. Implications and separations between panels. The cells with question marks remain open problems. An arrow in row Pn, column Pm indicates whether Pn implies or does not imply Pm. The superscript number next to an arrow indicates the number of the corresponding theorem. Arrows without a superscript follow by transitivity. See Section 7 for more details. If the red non-implications with a question mark superscript hold, all the remaining open cases will be non-implications by transitivity.

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 some implications between these 14 panels. The Section 7 is dedicated to show non-implications between panels. The relation between few panels are left as open questions. 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 basics of quantum information and computation needed for our paper below. Interested reader can refer to [20] 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. A 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 n vectors $|b_i\rangle$

with 1 in the position i and 0 elsewhere (these vectors will be represented by n vectors $\{|x\rangle : x \in \{0,1\}^{\log n}\}$). With this basis, the unitary CNOT is defined as CNOT: $|m_1, m_2\rangle \to |m_1, m_1 \oplus m_2\rangle$ 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 form $\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|$ where p_i are non-negative and add up to 1. This represents 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 "random" we mean "uniformly random". For a function f, the notation $\mathrm{im}\, f$ means $\{f(x) : x \in \{0,1\}^m\}$. 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

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

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 $f : \{0,1\}^h \to \{0,1\}^n$ (i.e., how to represent a classical function f 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.

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.

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.

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

$$Q: \quad |x\rangle \text{ —}\boxed{\hat{U}^f}\text{—} |f(x)\rangle$$

Definition 1. *A symmetric encryption scheme consists of there efficient algorithms* $(\mathrm{KGen}, \mathrm{Enc}, \mathrm{Dec})$ *as follows.*

- *The key generating algorithm* KGen *on the input of the security parameter returns a random secret key* k*.*
- *The encryption algorithm* Enc *on the inputs of* k *and a message* m *chooses a randomness* r *and returns* $\mathrm{Enc}_k(m;r)$*.*
- Dec *on the inputs of* k *and* $c = \mathrm{Enc}_k(m;r)$ *returns* m*. For an invalid ciphertext, the decryption algorithm* Dec *returns* $\perp$*.*

Definition 2. *We call two functions* f_1, f_2 *s-indistinguishable (short for standard indistinguishable) iff there exists a negligible* ε *such that for all quantum polynomial time adversaries* $\mathcal{A}$ *and all auxiliary quantum states* $|\psi\rangle$ *chosen by* $\mathcal{A}$ *(since* $\mathcal{A}$ *can use an internal quantum register to distinguish) 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 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 adversaries* $\mathcal{A}$ *making polynomial number of queries to its oracle in the query model* qm *and all auxiliary quantum states* $|\psi\rangle$ *chosen by* $\mathcal{A}$ *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:

- 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 distinguished, but as shown below they are equivalent. More formally cPRP, sPRP, qPRP are defined by:

Definition 3. *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 4. Note that Zhandry showed in [27] 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 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 5. *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 an invertible permutation. □

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 6. *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.*

The $\mathfrak{l}$-$\mathfrak{c}$-CPA game:

Key Gen: *The challenger runs* KGen *to obtain a key k, i.e., $k \xleftarrow{\$} \mathrm{KGen}()$, and it picks a random bit b.*
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 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.*
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. A fresh randomness will be chosen for each query (quantum or classical), but, for a superposition query, all the messages in the query will be encrypted with the same randomness [6].
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.

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.
Learning Query type EM. Upon receiving the provided register of the adversary, say Q_{in}, 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.
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.

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

We present different challenge query types in this section.
Challenge Query type chall($\cdot$, *CL*, 1ct). (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 $Enc_k(m_b; r)$
Challenge Query type chall($\cdot$, *ST*, 1ct). 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 three registers and returns the registers to the adversary.

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

Challenge Query type chall($\cdot$, *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 three registers and returns the registers to the adversary.

$$U_{EM,1\mathrm{ct},r,b} : |m_0, m_1, 0\rangle \mapsto |m_0, m_1, \oplus\mathrm{Enc}_k(m_b; r)\rangle.$$

Challenge Query type chall($\cdot$, *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.

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

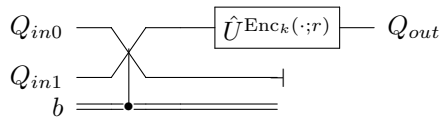
Challenge Query type chall($\cdot$, EM, 2ct). In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} and sends them to the challenger. The challenger prepares two registers Q_{out0}, Q_{out1} containing $|0\rangle^{\otimes n'}$, picks $r_0, r_1 \xleftarrow{\$} \{0,1\}^t$ and a random bit b, applies the following operation on these four registers and returns the registers to the adversary.

$$U_{EM,2\mathrm{ct},r_0||r_1,b} : |m_0, m_1, 0, 0\rangle \mapsto |m_0, m_1, \mathrm{Enc}_k(m_b; r_0), \mathrm{Enc}_k(m_{\bar{b}}; r_1)\rangle.$$

Challenge Query type chall($\cdot$, ER, 2ct).
In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} 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 two registers and returns the registers to the adversary.

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

Challenge Query type chall($\cdot$, ER, 1ct). In this query model, the adversary prepares two registers Q_{in0}, Q_{in1} 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 register Q_{inb}, and passes it back to the adversary.



where registers Q_{in0}, Q_{in1} will be swapped if and only if $b = 1$.
Challenge Query type chall($\cdot$, 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.
Challenge Query type chall($\cdot$, 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.
Challenge Query type chall($\cdot$, ER, ror). In this query model, the adversary prepares a register Q_{in} 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.

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

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, $\mathfrak{l} = \text{learn}(\mathfrak{l}_{nb}, \mathfrak{l}_{qm})$ 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 $\mathfrak{c} := \text{chall}(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt})$ ("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, $\mathfrak{c} = \text{chall}(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt})$ where $(\mathfrak{c}_{nb}, \mathfrak{c}_{qm}, \mathfrak{c}_{rt}) \in \{1, *\} \times \{CL, ST, EM, ER\} \times \{1\text{ct}, 2\text{ct}, \text{ror}\}$.

The valid combinations of the learning and challenge queries. We explicitly specify which combination of the learning queries, $\mathfrak{l}$, and the challenge queries, $\mathfrak{c}$, are considered in this paper. We consider only combinations where,

- $(\mathfrak{l}_{nb}, \mathfrak{c}_{nb}) \in \{(*, 1), (*, *), (0, *)\}$ i.e., $(\mathfrak{l}_{nb}, \mathfrak{c}_{nb}) \neq (0, 1)$.
- $(\mathfrak{l}_{qm}, \mathfrak{c}_{qm}) \in \{(CL, CL)\} \cup \{(CL, x), (x, CL), (x, x) | x \in \{ST, EM, ER\}\}$.

That is, we have excluded IND-OT-CPA definitions and notions that combine two different notions of superposition queries.

4 Decoherence lemmas

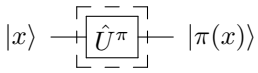
In this section, we present some lemmas needed in our paper without proof. For the proof, refer to the full version of the paper [8].

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.

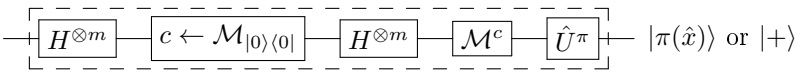
Lemma 7. *For a bijective function $\pi : \{0,1\}^m \to \{0,1\}^m$ let $\hat{U}^\pi$ be the unitary that performs the ER-type mapping $|x\rangle \mapsto |\pi(x)\rangle$. Let X be a quantum register with m qubits. Then the following two oracles can be distinguished in a single query with probability at most 2^{-m+2}:*

- *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:

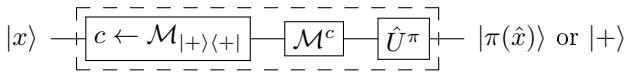


and for F_1 it is:



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 $\text{span}(|0\rangle^{\otimes m})$ (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 span$(|+\rangle^{\otimes m})$, we can write F_1 simply as:

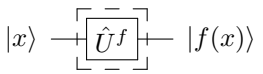


On a very high level, the proof proceeds as follows: We explicitly represent the density operators ρ_0, ρ_1 after execution of F_0, F_1, respectively (for a generic initial state). Then we show by explicit calculation that $\rho_0 = \rho'$ where ρ' is the state after F_1 if we omit the measurement $\mathcal{M}^c$. Finally we proceed to bound the trace distance between ρ_1 and ρ'. (This then gives a bound on the adversary's distinguishing probability.) This is done by explicitly computing $\rho_1 - \rho'$ and noting that this difference is a tensor product of two matrices σ_1, σ_2, both of reasonably simple form, and one of them having very small trace norm.

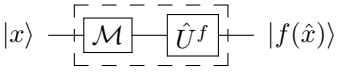
Lemma 8. *For numbers m and n and an injective function $f : \{0,1\}^m \to \{0,1\}^{m+n}$ let $\hat{U}^f$ be the isometry that performs the ER-type mapping $|x\rangle \mapsto |f(x)\rangle$. Let X be a quantum register containing m qubits. Then the following two oracles can be distinguished with probability 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:

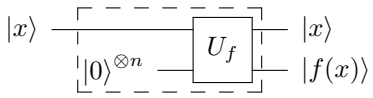


where $\mathcal{M}$ is a computational basis measurement (in the picture we denote the outcome of this measurement with $\hat{x}$).

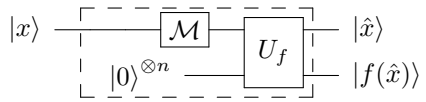
Proof. Intuitively this follows from Lemma 7 because: Picking a random injection has the same distribution as composing concatenation of sufficiently many 0s with a random permutation. □

Lemma 9. *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:*



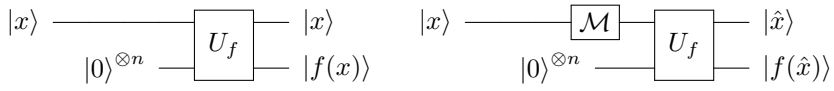
2. *F_1: measure X in the computational basis and apply U_f to the result and another register containing zeros. The circuit for F_1 is:*



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

Corollary 10. *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 9 are distinguishable with probability at most $1/2^n + C/2^n$ where C is a universal constant.*

Corollary 11. *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.*



5 Impossible Security Notions

Proposition 12. *[Theorem 4.2 in [6]] 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.*

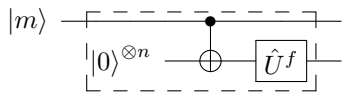
Proposition 13. *[Theorem 4.4 in [6]] There is no $\mathfrak{l}$-chall($\mathfrak{c}_{\mathfrak{nb}}, ST, 2\text{ct}$)-IND-CPA-secure encryption scheme where the $\mathfrak{l}$ and $\mathfrak{c}_{\mathfrak{nb}}$ can be replaced by any of the possible parameters.*

Proposition 14. *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.*

6 Implications

From the theoretically $(4+1) \times 2 \times 4 \times 3 = 120$ possible IND-CPA-notions, 12 correspond to IND-OT-CPA, 36 are considered exotic, 15 are impossible to achieve and this leaves 57 notions that are grouped in 14 Panels as described in the introduction. Inside each panel all the notions are equivalent and apart from that, there are 20 implications between the panels. The full set of implications between all notions can be derived by transitivity. Some of 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.

1. **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.
2. **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.
3. **other ciphertext:** If two security notions just differ by the fact, that one of them allows chall($\mathfrak{c}_{nb}, ER, 1\text{ct}$) challenge queries and the other chall($\mathfrak{c}_{nb}, ER, 2\text{ct}$) challenge queries, then trivially the notions allowing chall($\mathfrak{c}_{nb}, ER, 2\text{ct}$) challenge queries implies the notion allowing chall($\mathfrak{c}_{nb}, ER, 1\text{ct}$) challenge queries.
4. **simulate classical:** Classical queries can be simulated with any quantum query type by measuring the result in the computational basis.
5. **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 learning query, we can query (m, m) as a challenge query and simulate the learning query.
6. **EM simulation by ST**. The query type EM can be simulated by ST-type by putting $|0\rangle$ in the output register Q_{out}.
7. **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:



We show how the equivalences of the notions inside of the panels are derived.
Panel P1 (8 security notions):
learn$(*, CL)$-chall$(*, ER, 2\text{ct}) \implies$ learn$(*, CL)$-chall$(*, ER, 1\text{ct})$ by Item 3.
learn$(*, CL)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(0, -)$-chall$(*, ER, 1\text{ct})$ by Item 2.
learn$(0, -)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(*, ER, 1\text{ct})$ by Theorem 17.
learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(1, ER, 1\text{ct})$ by Item 1.
learn$(*, ER)$-chall$(1, ER, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(1, ER, 2\text{ct})$ by Theorem 20.
learn$(*, ER)$-chall$(1, ER, 2\text{ct}) \implies$ learn$(*, ER)$-chall$(*, ER, 2\text{ct})$ by Theorem 15.
learn$(*, ER)$-chall$(*, ER, 2\text{ct}) \implies$ learn$(0, -)$-chall$(*, ER, 2\text{ct})$ by Item 2.
learn$(0, -)$-chall$(*, ER, 2\text{ct}) \implies$ learn$(*, CL)$-chall$(*, ER, 2\text{ct})$ by Item 5.
Panel P2 (4 security notions):
learn$(*, ST)$-chall$(*, ST, \text{ror}) \implies$ learn$(*, CL)$-chall$(*, ST, \text{ror})$ by Item 4.
learn$(*, CL)$-chall$(*, ST, \text{ror}) \implies$ learn$(0, -)$-chall$(*, ST, \text{ror})$ Item 2.
learn$(0, -)$-chall$(*, ST, \text{ror}) \implies$ learn$(*, ST)$-chall$(*, ST, \text{ror})$ by Theorem 18.
learn$(*, ST)$-chall$(*, ST, \text{ror}) \implies$ learn$(*, ST)$-chall$(1, ST, \text{ror})$ Item 1.

learn$(*, ST)$-chall$(1, ST, \text{ror}) \implies$ learn$(*, ST)$-chall$(*, ST, \text{ror})$ by Theorem 15.

Panel P4 (4 security notions):

learn$(*, ER)$-chall$(*, ER, \text{ror}) \implies$ learn$(*, CL)$-chall$(*, ER, \text{ror})$ by Item 4.
learn$(*, CL)$-chall$(*, ER, \text{ror}) \implies$ learn$(0, -)$-chall$(*, ER, \text{ror})$ by Item 2.
learn$(0, -)$-chall$(*, ER, \text{ror}) \implies$ learn$(*, ER)$-chall$(*, ER, \text{ror})$ by Theorem 18.
learn$(*, ER)$-chall$(*, ER, \text{ror}) \implies$ learn$(*, ER)$-chall$(1, ER, \text{ror})$ by Item 1.
learn$(*, ER)$-chall$(1, ER, \text{ror}) \implies$ learn$(*, ER)$-chall$(*, ER, \text{ror})$ by Theorem 15.

Panel P5 (8 security notions):

learn$(*, EM)$-chall$(*, EM, \text{ror}) \implies$ learn$(*, CL)$-chall$(*, EM, \text{ror})$ by Item 4.
learn$(*, CL)$-chall$(*, EM, \text{ror}) \implies$ learn$(0, -)$-chall$(*, EM, \text{ror})$ by Item 2.
learn$(0, -)$-chall$(*, EM, \text{ror}) \implies$ learn$(*, EM)$-chall$(*, EM, \text{ror})$ by Theorem 18.
learn$(*, EM)$-chall$(*, EM, \text{ror}) \implies$ learn$(*, EM)$-chall$(1, EM, \text{ror})$ by Item 1.
learn$(*, EM)$-chall$(1, EM, \text{ror}) \implies$ learn$(*, EM)$-chall$(1, EM, 2\text{ct})$ by Theorem 19.
learn$(*, EM)$-chall$(1, EM, 2\text{ct}) \implies$ learn$(*, EM)$-chall$(*, EM, 2\text{ct})$ by Theorem 15.
learn$(*, EM)$-chall$(*, EM, 2\text{ct}) \implies$ learn$(*, CL)$-chall$(*, EM, 2\text{ct})$ by Item 4.
learn$(*, CL)$-chall$(*, EM, 2\text{ct}) \implies$ learn$(0, -)$-chall$(*, EM, 2\text{ct})$ by Item 2.
learn$(0, -)$-chall$(*, EM, 2\text{ct}) \implies$ learn$(*, EM)$-chall$(*, EM, 2\text{ct})$ by Theorem 17.
learn$(*, EM)$-chall$(*, EM, 2\text{ct}) \implies$ learn$(*, EM)$-chall$(*, EM, \text{ror})$ by Theorem 22.

Panel P6 (6 security notions):

learn$(*, ST)$-chall$(1, CL, 1\text{ct}) \implies$ learn$(*, ST)$-chall$(*, CL, 1\text{ct})$ by Theorem 15.
learn$(*, ST)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(*, ST)$-chall$(1, CL, 1\text{ct})$ by Item 1.
The rest of equivalences hold by Theorem 16.

Panel P10 (6 security notions):

learn$(*, ER)$-chall$(1, CL, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(*, CL, 1\text{ct})$ by Theorem 15.
learn$(*, ER)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(1, CL, 1\text{ct})$ by Item 1.
The rest of equivalences hold by Theorem 16.

Panel P11 (6 security notions):

learn$(*, EM)$-chall$(1, CL, 1\text{ct}) \implies$ learn$(*, EM)$-chall$(*, CL, 1\text{ct})$ by Theorem 15.
learn$(*, EM)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(*, EM)$-chall$(1, CL, 1\text{ct})$ by Item 1.
The rest of equivalences hold by Theorem 16.

Panel P14 (9 security notions):

learn$(*, CL)$-chall$(1, CL, 1\text{ct}) \implies$ learn$(*, CL)$-chall$(*, CL, 1\text{ct})$ by Theorem 15.
learn$(*, CL)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(*, CL)$-chall$(1, CL, 1\text{ct})$ by Item 1.
learn$(*, CL)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(0, -)$-chall$(*, CL, 1\text{ct})$ by Item 2.
learn$(0, -)$-chall$(*, CL, 1\text{ct}) \implies$ learn$(*, CL)$-chall$(*, CL, 1\text{ct})$ by Item 5.
The rest of equivalences hold by Theorem 16.

The implications between the panels that does not have superscript in Table 1 hold using one of the described arguments above.

Now we present the theorem mentioned in Table 1 and we refer to the full version of the paper [8] for a detailed proof.

In Theorem 15, we prove that if we fix all the parameters in two notions expect the number of the challenge queries (that can be one or many), the notion with many challenge queries implies the notion with one challenge query if one can simulate the challenge queries with the learning queries (when knowing the challenge bit).

Theorem 15. *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* $\text{learn}(*, \mathfrak{l}_{\text{qm}})$ -$\text{chall}(1, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}}) \implies \text{learn}(*, \mathfrak{l}_{\text{qm}})$-$\text{chall}(*, \mathfrak{c}_{\text{qm}}, \mathfrak{c}_{\text{rt}})$.

In the following theorem, we show that when the challenge queries are classical and we fix other parameters except the return types, these notions (with different return types $1\text{ct}, 2\text{ct}, \text{ror}$) are equivalent.

Theorem 16. *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.)*

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

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.

Theorem 17. $\text{learn}(0,-)$-$\mathfrak{c} \implies \text{learn}(*, \mathfrak{l}_{\text{qm}})$-$\mathfrak{c}$ *where*

$\mathfrak{c} \in \{\text{chall}(*, EM, 2\text{ct}), \text{chall}(*, ER, 2\text{ct}), \text{chall}(*, ER, 1\text{ct})\}$ *and* $\mathfrak{l}_{\text{qm}} \in \{EM, ER\}$.

In the theorem below, we show that the security definition with no learning queries imply the security definition that performs ST, EM and ER type learning queries when the return type of the challenge queries is ror.

Theorem 18. $\text{learn}(0,-)$-$\text{chall}(*, \mathfrak{c}_{\text{qm}}, \text{ror}) \implies \text{learn}(*, \mathfrak{c}_{\text{qm}})$-$\text{chall}(*, \mathfrak{c}_{\text{qm}}, \text{ror})$, *where* $\mathfrak{c}_{\text{qm}} \in \{ST, EM, ER\}$.

In the theorem below, we show that for the embedding query type, ror-challenge queries imply 2ct-challenge queries. A less general version of this theorem (when there is only one challenge query) is used to show the equivalences of the notions in Panel 5.

Theorem 19. $\text{learn}(*, EM)$-$\text{chall}(*, EM, \text{ror}) \implies \text{learn}(*, EM)$-$\text{chall}(*, EM, 2\text{ct})$

In Theorem 20, we show that when the query model is ER in both the learning queries and the challenge queries, the return type 1ct implies 2ct.

Theorem 20. learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(*, ER)$-chall$(*, ER, 2\text{ct})$

In Theorem 21, we show that 1ct return type implies ror return type for ER query model.

Theorem 21. *The following implications hold:*

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

In Theorem 22, we show that the 2ct return type implies the ror return type for the EM query model.

Theorem 22. *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$(*, EM, \text{ror})$

Theorem 23. learn$(*, ER)$-chall$(*, ER, 1\text{ct}) \implies$ learn$(*, ST)$-chall$(*, CL, 1\text{ct})$. *This shows that P1* $\implies$ *P6.*

7 Separations

In this section, we show all the non-implications presented in Table 1. For more detailed proofs of theorems, refer to [8]. Note that some of non-implications in Table 1 hold trivially by transitivity.

7.1 Separations by Quasi-Length-Preserving Encryptions

The notion of a core function and quasi-length-preserving encryption schemes was first formally introduced in [12]. 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 24 (Core function). *A function g is called the core function of an encryption scheme* (KGen, Enc, Dec) *if*

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

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

   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 25 (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.

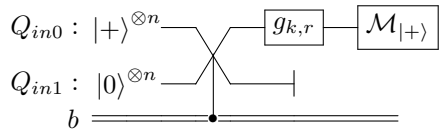
In the theorem below we show that any quasi-length-preserving encryption scheme is insecure for the query model in Panel 8.

Theorem 26. *Any quasi-length-preserving encryption scheme is insecure for the query model* learn$(*, CL)$-chall$(1, ER, 1\text{ct})$. *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

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

for some functions f and g such that $|g(k,m,r)| = |m|$. We draw the circuit of the attack below where Q_{in0}, Q_{in1} are two input registers. 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.

Theorem 27. *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 and 4.

Proof. Let $\text{Enc}_k(m;r) = \text{sPRP}_k(r)||\text{qPRP}_r(m)$ where qPRP is a strong quantum-secure pseudorandom permutation [27] and sPRP is a standard-secure pseudorandom permutation. Since in each query r is a fresh randomness, then qPRP_r is a fresh permutation. Now the security is straightforward. □

Corollary 28. *The security notions mentioned in Theorem 27 do not imply the security notions mentioned in Theorem 26. Specifically, P2, P4 $\not\Rightarrow$ P8.*

7.2 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 [22]) 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 [22] that $\hat{n} = O(|\sigma|)$ queries 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 29. *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*[9] *(qPRP) with the seed s_i and input/output length of $n/2$. Let*

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

Note that $f_{s,\sigma,i}$ is σ-periodic ignoring its second part. 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 decryptable.

Lemma 30. *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 indistinguishable from a truly random function for any quantum adversary restricted to make only one classical query.*

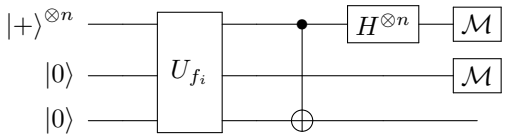
Proof. We show that for every i and y, $f_{s,\sigma,i}(y)$ is indistinguishable from a random bitstring. Since $y \oplus r_i$ is indistinguishable from a random bitstring (for random r_i), it is left to show $g_{s,\sigma,i}(y) = P_{s_i}(y) \oplus P_{s_i}(y \oplus \sigma)$ is indistinguishable from a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

□

Lemma 31. *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.)*

[9] Quantum secure pseudorandom permutation can be constructed from a quantum secure one-way function [27].

Proof. The attack is a variation of Simon's attack [22]. 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 32. *If there exists a quantum secure one-way function then* learn$(*, CL)$-chall$(*, CL, 1\text{ct}) \not\Longrightarrow$ learn$(*, EM)$-chall$(1, CL, 1\text{ct})$. *This shows that Panel 14* $\not\Longrightarrow$ *Panel 11.*

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

Here PRF_k and $\text{PRF}_{k'}$ are standard secure pseudorandom functions with the key k, k' respectively. It is easy to see $\text{Enc}_{k,k'}$ is decryptable. Since r, r' are fresh randomness in each query, the security of Enc in the sense of learn$(*, CL)$-chall$(*, CL, 1\text{ct})$ follows by the Lemma 30. Now we show how EM leaning queries can break the security of Enc. In the attack, the adversary uses one learning query to retrieve k, according to Lemma 31 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 $\text{PRF}_k(r') \oplus m'$.) □

Theorem 33. *If there exists a quantum secure one-way function then the following non-implication holds:*

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

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.

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

where $F_{s,l}$ is defined in Definition 29. It is easy to see that Enc_k is decryptable. Now we show that Enc is insecure in the learn$(*, CL)$-chall$(*, EM, \text{ror})$-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 31 to the last part of the ciphertext to recover l. Let $\hat{l}$ be the output of the algorithm presented in Lemma 31. 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. When the challenge bit is $b = 0$, the algorithm in Lemma 31 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 11 a measurement on the input register remains indistinguishable for $\mathcal{A}$ (with $R := \text{range } \mathrm{Enc}_k(\cdot; r)$ in Corollary 11). So we can assume the input register collapses to a classical message. Therefore $\mathcal{A}$ will recover l with negligible probability.

We show that Enc is secure in the learn$(*, ER)$-chall$(*, CL, 1\text{ct})$-sense. Let G_b be the learn$(*, ER)$-chall$(*, CL, 1\text{ct})$-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 30 $F_{s,l}(x_b)$ 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. □

7.3 Separations by Shi's SetEquality problem

Definition 34 (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 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 $\mathcal{D}$.

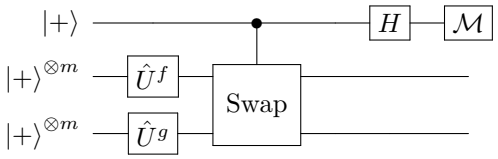
The SetEquality problem was first posed by Shi [21] in the context of quantum query complexity. In [26] it is proven that with ST-type-oracle access this problem is hard in m. However, a trivial implication of the swap-test shows that with ER-type oracle access it has constant complexity.

Lemma 35. *The* SetEquality *problem is indistinguishable under polynomial ST-type queries.*

Proof. This follows from Theorem 4 in [26], which shows that $\Omega\left(2^{m/3}\right)$ ST-type queries are required to distinguish the two cases. □

Lemma 36. *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 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 37. *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:*

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

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:

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

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

Proof. Let $H : \{0,1\}^h \to \{0,1\}^{|s|}$ be a random oracle. Let $sPRP$ be a standard secure pseudorandom 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

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

where $qPRP_r$ is a quantum secure pseudorandom permutation with seed r. It is easy to see that the encryption scheme above is decryptable. We sketch the proof of the security in the sense of learn$(*, ST)$-chall$(1, ST, \mathrm{ror})$. We start with an adversary that attacks the encryption scheme in the sense of learn$(*, ST)$-chall$(1, ST, \mathrm{ror})$ IND-CPA. Then, in each query we replace $H(r)$ and $H(k)$ with random values. To bound the advantage of $\mathcal{A}$ in distinguishing these replacements, we use the Theorem 3 in [2]. Since the set-equality problem is hard for ST-type queries by Lemma 35, we can ignore the key k in γ_k function and simply choose two random injection functions F_1^* and G_1^* with disjoint ranges in the challenge query. Let $\gamma'(m_1^*||m_2^*; r^*, j^*) := F_1^*(m_1^*)||G_1^*(m_2^*)$. It is clear that the advantage of $\mathcal{A}$ in the last game is $1/2$ since $\gamma' \circ qPRP \circ (m_1^*||m_2^*)$ (when $b = 0$) and $\gamma' \circ qPRP \circ \pi(m_1^*||m_2^*)$ (when $b = 1$) are indistinguishable.

Insecurity. Now we show that Enc can be broken with ER learning queries. By Lemma 36, it is possible that the adversary performs 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 the ciphertext and is known to the adversary.) The procedure is repeated polynomially many times until the adversary has enough information about the key k to guess it with a sufficiently high probability. Finally, the adversary 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. □

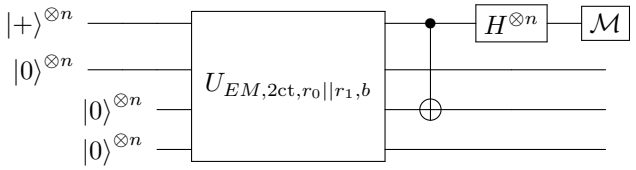
7.4 Separations by other arguments

Theorem 39. *On the existence of a quantum secure one-way function, the following separation holds:* learn$(*, ST)$-chall$(*, CL, 1\mathrm{ct})$, learn$(*, ER)$-chall$(*, CL, 1\mathrm{ct}) \not\Longrightarrow$ learn$(*, CL)$-chall$(1, EM, 2\mathrm{ct})$. *That is P6, P10 $\not\Longrightarrow$ P7.*

Proof. Consider

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

where PRF is a standard secure pseudorandom function. The security in learn$(*, ST)$-chall$(*, CL, 1\mathrm{ct})$ and learn$(*, ER)$-chall$(*, CL, 1\mathrm{ct})$ senses follows by Lemma 3 in [3]. We show the insecurity using a challenge query of type chall$(1, EM, 2\mathrm{ct})$. 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 40. *On the existence of a quantum secure one-way function,* learn$(*, ST)$-chall$(*, CL, 1\mathrm{ct})$ $\not\Longrightarrow$ learn$(*, CL)$-chall$(1, EM, 1\mathrm{ct})$. *This shows that P6* $\not\Longrightarrow$ *P13.*

Proof. Consider

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

where PRP is a standard secure pseudorandom permutation. The security in learn$(*, ST)$-chall$(*, CL, 1\mathrm{ct})$ and learn$(*, ER)$-chall$(*, CL, 1\mathrm{ct})$ senses follows by Lemma 3 in [3]. The insecurity follows from Lemma 10 in [9]. □

Theorem 41. *On the existence of a quantum secure one-way function,* learn$(*, ER)$-chall$(1, ER, 1\mathrm{ct})$ $\not\Longrightarrow$ learn$(*, CL)$-chall$(1, ST, \mathrm{ror})$. *This shows that P1* $\not\Longrightarrow$ *P12.*

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

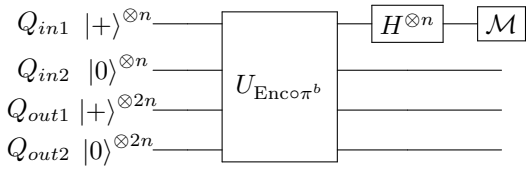
$$\mathrm{Enc}_k(m_1, m_2; r_1, r_2) = qPRP_{r_1}(0^n||m_1)||qPRP_{r_2}(0^n||m_2)||sPRP_k(r_1, r_2).$$

The security in the sense of learn$(*, ER)$-chall$(1, ER, 1\mathrm{ct})$ follows because in each query a fresh $qPRP$ is used and we can measure the input register of queries by Lemma 8. Thus, the security follows by the security in the sense of learn$(*, CL)$ -chall$(1, CL, 1\mathrm{ct})$.

Now we show that Enc is not secure with respect to learn$(*, CL)$-chall$(1, ST, \mathrm{ror})$ notion. Let Q_{in1} and Q_{in2} be input registers corresponding to first n bits and second n bits of message, respectively. Similarly, Q_{out1} and Q_{out2} be the output registers. The adversary can query

$$Q_{in1}Q_{in2}Q_{out1}Q_{out2} := |+\rangle^{\otimes n}|0\rangle^{\otimes n}|+\rangle^{\otimes 2n}|0\rangle^{\otimes 2n}$$

in the challenge query. After receiving the answer, it applies the Hadamard operator to Q_{in1} then measures the register in the computational basis. We draw the circuit to attack Enc in the following. For simplicity, we omit the wires corresponding to the last parts of two ciphertexts.



When $b = 0$, since no permutation is applied and Enc works component-wise, the output of the circuit right before applying the Hadamard operators is

$$|+\rangle^{\otimes n}|0\rangle^{\otimes n}|+\rangle^{\otimes 2n}|qPRP_{\tau_2}(0^n||0^n)\rangle^{\otimes 2n}.$$

Therefore, the measurement returns 0 with probability 1. On the other hand, when $b = 1$ a permutation will be applied to both input registers Q_{in1}, Q_{in2} and it shuffles the input. Therefore Q_{in1} register will be entangled with output registers. In this case, the measurement returns 0 with negligible probability. □

Note that a block cipher mode of operation uses a block cipher several times to encrypt a message of longer size. In the following we show that the attack presented above can be applied to a large class of modes of operation and show their insecurity with respect to learn$(*, CL)$-chall$(1, ST, \mathrm{ror})$ notion. This can be extended to authentication encryption schemes and tweakable block ciphers.

Corollary 42. *We call a mode of operation* natural *if it has the following property: For some message length ℓ, there exists an input block i and an output block j such that output block j does not depend on i, but, ranging over all possible input messages, output block j can take any value. (Note that this includes many modes of operation. E.g., CBC mode satisfies this with i being the second and j being the first block.) Then, no natural mode of operation is secure in the sense of* learn$(*, CL)$-chall$(1, ST, \mathrm{ror})$ *notion.*

Proof. The attack is similar to the above by inserting $|+\rangle$ in the M_i register and $|0\rangle$ for the rest of the input registers and inserting $|0\rangle$ in the j-th output register and $|+\rangle$ elsewhere. Then applying the Hadamard operator to the register M_i followed with a computational basis measurement. □

8 Encryption secure in all notions

In this section we propose an encryption scheme that is secure for all security notions described in this paper. From Table 1, Panel 1 and Panel 2 imply all other panels. Therefore it is sufficient to construct an encryption scheme 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 $\mathfrak{c}_2 = \mathrm{chall}(*, ST, \mathrm{ror})$.

Theorem 43. *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 pseudorandom function. Now we can replace $qPRP_r$ with a random permutation π' in each query and use Lemma 8 to measure the input register (with $f := \pi'(r'||\cdot)$). This collapses to the security against $\mathrm{chall}(*, CL, 1\mathrm{ct})$ queries that 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 pseudorandom 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. □

Acknowledgments. This work was supported by the United States Air Force Office of Scientific Research (AFOSR) via AOARD Grant "Verification of Quantum Cryptography" (FA2386-17-1-4022), by the ERC consolidator grant CerQuS (819317), by the Estonian Centre of Exellence in IT (EXCITE) funded by ERDF, and by the IUT2-1 grant and the PUT team grant PRG946 from the Estonian Research Council.

References

1. G. Alagic, C. Majenz, A. Russell, and F. Song. Quantum-access-secure message authentication via blind-unforgeability. In *EUROCRYPT 2020*, volume 12107 of *LNCS*, pages 788–817. Springer, 2020.
2. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019*, volume 11693 of *LNCS*, pages 269–295. Springer, 2019.
3. M. V. Anand, E. E. Targhi, G. N. Tabia, and D. Unruh. Post-quantum security of the cbc, cfb, ofb, ctr, and XTS modes of operation. In *PQCrypto 2016*, volume 9606 of *LNCS*, pages 44–63. Springer, 2016.
4. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *FOCS 1997*, pages 394–403. IEEE Computer Society, 1997.
5. D. Boneh and M. Zhandry. Quantum-secure message authentication codes. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, 2013.
6. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *CRYPTO 2013*, volume 8043 of *LNCS*, pages 361–379. Springer, 2013.
7. A. Broadbent and S. Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In *CRYPTO 2015*, volume 9216 of *LNCS*, pages 609–629. Springer, 2015.

8. T. V. Carstens, E. Ebrahimi, G. N. Tabia, and D. Unruh. Relationships between quantum IND-CPA notions. IACR ePrint 2020/596, 2021. Full version of this paper.
9. C. Chevalier, E. Ebrahimi, and Q. H. Vu. On the security notions for encryption in a quantum world. *IACR Cryptology ePrint Archive*, 2020:237, 2020.
10. I. Damgård, J. Funder, J. B. Nielsen, and L. Salvail. Superposition attacks on cryptographic protocols. In *ICITS 2013*, volume 8317 of *LNCS*, pages 142–161. Springer, 2013.
11. E. Ebrahimi, C. Chevalier, M. Kaplan, and M. Minelli. Superposition attack on OT protocols. *IACR Cryptol. ePrint Arch.*, 2020:798, 2020.
12. T. Gagliardoni, A. Hülsing, and C. Schaffner. Semantic security and indistinguishability in the quantum world. In *CRYPTO 2016*, volume 9816 of *LNCS*, pages 60–89. Springer, 2016.
13. T. Gagliardoni, J. Krämer, and P. Struck. Quantum indistinguishability for public key encryption. *IACR Cryptol. ePrint Arch.*, 2020:266, 2020.
14. M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO 2016*, volume 9815 of *LNCS*, pages 207–237. Springer, 2016.
15. E. Kashefi, A. Kent, V. Vedral, and K. Banaszek. Comparison of quantum oracles. *Phys. Rev. A*, 65:050304, May 2002.
16. H. Kuwakado and M. Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *ISIT 2010*, pages 2682–2685. IEEE, 2010.
17. H. Kuwakado and M. Morii. Security on the quantum-type even-mansour cipher. In *ISITA 2012*, pages 312–316. IEEE, 2012.
18. Q. Liu, A. Sahai, and M. Zhandry. Quantum immune one-time memories. *IACR Cryptol. ePrint Arch.*, 2020:871, 2020.
19. S. Mossayebi and R. Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *CoRR*, abs/1609.03780, 2016.
20. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.
21. Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. In *FOCS 2002*, pages 513–519, 2002.
22. D. R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, Oct. 1997.
23. D. Unruh. Quantum proofs of knowledge. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, 2012.
24. D. Unruh. Computationally binding quantum commitments. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 497–527. Springer, 2016.
25. J. Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.
26. M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.
27. M. Zhandry. A note on quantum-secure prps. *IACR Cryptology ePrint Archive*, 2016:1076, 2016.