On Security Notions for Encryption in a Quantum World

Céline Chevalier[1], Ehsan Ebrahimi[2,3], and Quoc-Huy Vu[1]

[1] CRED, Université Panthéon-Assas, Paris II, France
{celine.chevalier, quoc.huy.vu}@ens.fr
[2] SnT, University of Luxembourg
[3] Work done while at École Normale Supérieure
ehsan.ebrahimi@uni.lu

Abstract. Indistinguishability against adaptive chosen-ciphertext attacks (IND-CCA2) is usually considered the most desirable security notion for classical encryption. In this work, we investigate its adaptation in the quantum world, when an adversary can perform superposition queries. The security of quantum-secure classical encryption has first been studied by Boneh and Zhandry (CRYPTO'13), but they restricted the adversary to classical challenge queries, which makes the indistinguishability only hold for classical messages (IND-qCCA2). We extend their work by giving the first security notions for fully quantum indistinguishability under quantum adaptive chosen-ciphertext attacks, where the indistinguishability holds for superposition of plaintexts (qIND-qCCA2). This resolves an open problem asked by Gagliardoni *et al.* (CRYPTO'16).

The qCCA2 security is defined in Boneh-Zhandry's paper using string copying and comparison, which is inherent in the classical setting. Quantumly, it is unclear what it means for a ciphertext to be different from the challenge ciphertext, and how the challenger can check the equality. The classical approach would either violate the no-cloning theorem or lead to perturbing the adversary's state, which may be detectable. To remedy these problems, from the recent groundbreaking compressed oracle technique introduced by Zhandry (CRYPTO'19), we develop a generic framework that allows recording quantum queries for probabilistic functions. We then give definitions for fully quantum real-or-random indistinguishability under adaptive chosen-ciphertext attacks (qIND-qCCA2).

In the symmetric setting, we show that various classical modes of encryption are trivially broken in our security notions. We then provide the first formal proof for quantum security of the Encrypt-then-MAC paradigm, which also answers an open problem posed by Boneh and Zhandry.

In the public-key setting, we show how to achieve these stronger security notions (qIND-qCCA2) from any encryption scheme secure in the sense of Boneh-Zhandry (IND-qCCA2). Along the way, we also give the first definitions of non-malleability for classical encryption in the quantum world and show that the picture of the relations between these notions is essentially the same as in the classical setting.

1 Introduction

Recent advances in quantum computing [AAB$^+$19] show the possible emergence of new kinds of attacks due to quantum adversaries. The first type of attacks would be due to adversaries owning a quantum computer and using it to break computational assumptions (thus attacking classical cryptographic cryptosystems). This has been made possible by the invention of quantum algorithms that solve factoring and discrete logarithm problems in polynomial time [Sho99] and consequently, break the security of many classical public-key encryption schemes based on these assumptions. This threat has led to the emergence of so-called *post-quantum cryptography*, based on arguably quantum-resistant assumptions. But this change of assumptions may not be sufficient, and symmetric cryptosystems may also be impacted, in case we allow a quantum adversary, not only to perform computation on a quantum computer it may own, but also to carry out a second type of attacks, by interacting with the target in superposition. Quantum algorithms for unstructured search [Gro96] or period finding [Sim94] could then be applied to attack classical constructions using superposition queries [DFNS14,KLLN16]. Cryptosystems secure against this type of attacks would be called *quantum secure*.

As we approach the quantum era, it thus becomes necessary to construct new public-key cryptosystems based on quantum-resistant assumptions, and to investigate the security of both symmetric and public-key cryptosystems against an attacker allowed to interact with honest parties using quantum communication. Recently, there has been towards this goal extensive research works that consider this scenario of quantum superposition attacks for different classical cryptographic constructions such as random oracles, pseudorandom functions, encryption and signature schemes [BDF$^+$11,Zha12,BZ13b,BZ13a,GHS16] and give corresponding new security definitions. Furthermore, this new field of research is also motivated by the existence of concrete attacks against classical constructions using superposition queries (e.g., see [DFNS14,KLLN16] and their follow up works). In this paper, we continue this line of work and focus on the security for classical encryption schemes against quantum adversaries allowed to make quantum encryption and decryption queries.

1.1 Defining Security for Encryption Against Quantum Adversaries

Classical Security Notions. Indistinguishability-based security definitions are modeled as a game between a challenger and an adversary $\mathcal{A}$. The game starts with a first learning phase (with access to some oracles), followed by a challenge phase where $\mathcal{A}$ sends a challenge query (two messages x_0 and x_1 to encrypt) and receives a challenge ciphertext (encryption of x_b). Afterward, a second learning phase follows, and finally, $\mathcal{A}$ outputs a solution (its guess for the bit b). The security reduction consists of constructing a new adversary which simulates $\mathcal{A}$ and solves some hard underlying problem. The learning phases define the type of attacks: chosen-plaintext attacks (CPA) if the adversary has access to an encryption oracle in both learning phases, and chosen-ciphertext attacks (CCA) in case it also has access to a decryption oracle in the learning phases (non-adaptive or CCA1 if it is restricted to the first learning phase, and adaptive or CCA2 otherwise).

Indistinguishability against adaptive chosen-ciphertext attack (IND-CCA2) is usually considered the most desirable security notion for encryption. In the CCA2 games, the adversary is restricted not to ask for decryption of the challenge ciphertext, otherwise, this would lead to a trivial guess of the bit b. It is the role of the challenger to ensure that the adversary obeys this rule, which intrinsically requires the ability to copy, store and compare classical strings.

Quantum Attacks on Encryption. With recent advances in quantum computing, a quantum adversary may become a tangible threat in not so long. Switching to post-quantum computational assumptions is a beginning but may not be enough in case the adversary gains quantum access to honest parties and protocols. Consider for instance the well-known construction of CCA2 secure encryption schemes from lossy trapdoor functions [PW08]: if the construction is instantiated with lattice-based problems, it is arguably post-quantum secure. But we show later that, the insecurity may arise from the use of a one-time pad inside the construction (if one implements this scheme naively). Furthermore, [DFNS14,KLLN16] and their follow up works show that the security of several classical constructions can be compromised if the adversary can perform superposition attacks.

Boneh-Zhandry's Security Notions [BZ13b]. Boneh and Zhandry propose the first definition of IND-CCA for both symmetric and public-key encryption schemes against quantum adversaries allowed to make quantum encryption and decryption queries. But they show that the natural translation of the classical Find-then-Guess paradigm to the quantum setting is unachievable, even for IND-CPA security (we discuss in detail in Section 1.4). To overcome this impossibility, they resort to considering quantum queries during the learning phases only, and classical queries during the challenge phase. In addition to looking artificial, this inconsistency between the learning phases and the challenge phase may lead to a cryptographic construction that fulfills this security notion (IND-qCPA or IND-qCCA) while being subject to an attack.

For instance, in [ATTU16], the authors verify IND-qCPA security of XTS mode of operation (with quantum learning queries and classical challenge queries). They design a block cipher such that an encryption scheme in XTS mode, instantiated with that block cipher, can be attacked during the learning phase using quantum learning queries. However, this attack cannot be used to violate the IND-qCPA security definition. The explanation for this inconsistency is that this attack cannot be implemented in the challenge phase due to the classical restriction imposed on the adversary. This example supports our claim that the inconsistency between the learning phases and the challenge phase can be problematic and should be overcome.

IND-CCA2 Security Notions. To date, defining the CCA2 security with quantum challenge queries remains unsolved. In [GHS16], the authors address the inconsistency described above for the case of symmetric encryption, but only for IND-CPA, and leave as an open problem for the IND-CCA definitions.

The main obstacle is to define how the challenger should reply to the quantum decryption queries after the adversary has made the quantum challenge queries. When the challenge queries are classical, they can be stored and later the challenger can return $\perp$ if the adversary submits one of them as a decryption query. Although it is trivial and inherent to store the challenge ciphertext in the classical setting, it is highly non-trivial to store ciphertexts in the quantum world, due to a number of technical obstacles, all of which can be traced to quantum no-cloning [WZ82] and the destructiveness of quantum measurements [FP96].

Since we now consider the adversary's challenge queries as quantum states, it may be tempting to think that the approaches from the literature on quantum encryption (that is, the problem of encrypting *quantum* data) would work here. The notorious "recording barrier" that we face in this work has arisen previously in the literature on quantum encryption. In particular, devising the notions of quantum ciphertext indistinguishability under adaptive chosen-ciphertext attack and quantum authenticated encryption [BJ15,AGM18] requires circumventing similar obstacles.

In this paper, we manage to overcome this recording barrier by using Zhandry's compressed oracle technique [Zha19] (an overview is given in Section 1.2) and we propose the first quantum version for IND-CCA security notion. We justify our definitions in Section 1.3. Finally, in Section 1.4, we discuss our work compared to that of [GHS16] and we also briefly restate the approach of [AGM18] and explain why it does not obliviously work in our setting.

1.2 Our Technique

Towards resolution, we depart from a very recent groundbreaking technique that allows for on-the-fly simulation of random oracles in the quantum setting: Zhandry's compressed oracles [Zha19]. The goal of his work is to overcome the recording barrier, by allowing the reduction to record information about the adversary's queries, which is a key feature of many classical ROM proofs.

Zhandry's key observations are threefold. First, instead of considering a random function h being chosen beforehand, one can purify the adversary's mixed state by putting h in uniform superposition $\sum_h |h\rangle$. This observation is a technicality that allows us to fulfill the two next points. Then, the next observation is that, by doing the queries in the Fourier basis, the data will be written to the oracle's registers instead of writing to the opposite direction. This enables the simulator to get some information about the adversary's queries. Finally, the last and most important one is that the simulator needs to be ready to forget some point it simulated previously, by performing a particular test on the database after answering the query. In particular, Zhandry defines a test computation that maps $|+\rangle \mapsto |+\rangle |1\rangle$ and $|\phi\rangle \mapsto |\phi\rangle |0\rangle$ for any $|\phi\rangle$ is orthogonal to $|+\rangle$, where $|+\rangle = \sum_x |x\rangle$ is the uniform superposition state. The "test-and-forget" procedure can be implemented by first performing the query in the Fourier basis and then doing the test operation on the output registers (of the simulator).

This test determines whether the adversary has any information of the oracle at some input. If not, that pair will be removed from the database so that the adversary cannot detect that it is interacting with a simulated oracle.

This technique has been extended from random oracles to lazy-sampling of non-uniform random functions in [CMSZ19]. The intuition is almost the same, except that now one starts from the all-zero state, performs an *efficient* operation that computes the function $f(x)$ according to some non-uniform distribution–it is the quantum Fourier transform (QFT) operation in the uniform setting. One then performs the query in the Fourier basis, transforms back to the computational basis and applies the "test-and-forget" operation (which is defined similarly as in the uniform setting).

This idea seems to give us some possibilities to record the challenge queries. In order to make the challenger able to record the quantum encryption queries in the challenge phase and to answer the quantum decryption queries in the second learning phase, we implement the encryption oracle as a compressed oracle which keeps in its internal state a database of the adversary's queries. Informally, it would work as follows, where we denote the randomized encryption algorithm as a probabilistic function $f : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$.

- For each encryption query, sample a uniformly and independently random coin. From the adversary's perspective, it is equivalent to purifying a coin toss. The joint system state can be written as $\sum_{x,y} \alpha_{x,y} \left|x, y\right\rangle_{XY} \otimes \sum_r \left|r\right\rangle_R$.
- Initiate new registers and compute $\left|x, y\right\rangle_{XY} \left|r\right\rangle_R \mapsto \left|x, y\right\rangle_{XY} \left|r\right\rangle_R \left|f(x; r)\right\rangle_F$.
- Change the basis of the adversary's Y registers and the oracle's F registers to the Fourier basis (by applying QFT operations on these registers), and update the F registers to contain $(f(x; r) \oplus y)$.
- Change the F registers back to the computational basis, un-compute $f(x; r)$ and check if it is all-zero. If yes, discard it. Otherwise, compute f again and record a pair $(x, f(x; r))$ into the database D (initially empty).

Instantiating the encryption oracle with this approach would offer a simple way to keep track of the information needed to formulate the CCA2 notions, namely the challenge queries the adversary has made, and the challenge ciphertexts it has received. Given the ability to record the challenge ciphertext, the decryption procedure in the second phase of a CCA2 game can look up for a pair $(x, y) \in D$ on every query on y (in superposition), and return $\perp$ if it is found; otherwise, the normal decryption algorithm is applied.

The above idea gives us a reasonable way to define adaptive chosen ciphertext security against quantum challenge queries, which partially fulfill our goals. Unfortunately, there are several shortcomings to this approach. First, implementing the compressed encryption oracle this way needs one to make at least three calls to the encryption algorithm: one computing query and one un-computing query to perform the "test-and-forget" procedure, and one last computing query to record the information. Second, we note that these computations of the encryption (within a query of the adversary) are needed to be always the same. This requirement is guaranteed in the security definition because the challenger has controls over all the randomness.

Now, imagine that we want to use this approach to prove the security of the Encrypt-then-MAC paradigm. At some point in the proof, the reduction algorithm would play a game against the underlying CPA encryption. In this game, the reduction no longer has access to the decryption oracle, thus, it needs to be able to record all the encryption queries to answer subsequent decryption queries itself. However, in this case, the CPA encryption, which is done by some external challenger, would return different ciphertexts (which are computed with different randomness). This renders the recording technique unusable.

Another example is that this approach would not help us to prove the security of some classical paradigm including one-time adaptive chosen ciphertext security from one-time CPA encryption and one-time MAC, for a clear reason that the reduction algorithm needs to make at least three calls to the challenger to implement the database, which would break the one-time security anyway.

To remedy the problem, we go one step further and change the order of the computations: first, check whether the adversary's response register is zero (in the Fourier basis), and only perform the computation of f if it is not. To see why this works, notice that since the encryption is randomized, each query is treated with an independent, separated instance of the oracle. With our compressed oracle, this means that before each query, the database for that query is always empty. Then, in order to avoid entanglement attacks by the adversary, the compressed oracle operation on a pair (x, y) can be informally defined as: if $y = 0$, do nothing; otherwise add $(x, f(x; r))$ to the database. By implementing the compressed encryption oracle this way, we only compute f once for each query, which thus resolves all the problems mentioned above. In Section 3, we formally show how to obtain the compressed encryption oracle.

1.3 Our Contributions

Quantum Oracle Queries Recording Framework. Towards our main goal, we first build on Zhandry's compressed oracle technique to propose a quantum queries recording framework for probabilistic functions that supports answering inverse queries in Section 3. That is, the adversary is allowed to make quantum queries to the oracles of both a function f and its inverse f^{-1}. The family of functions we consider covers a broad class of cryptographic primitives including permutations, symmetric encryption, and public-key encryption *with decryption failures*. We use this recording technique in various ways in what follows but this contribution is of independent interest and we hope it can find applications in other settings as well.

New Notions of Quantum Indistinguishability. Based on this framework, we define novel security notions for encryption in both the symmetric (Definition 2 in Section 4) and public-key settings (Definition 3 in Section 5). Our main contribution is to propose the first definitions for adaptive chosen ciphertext security that support *fully quantum indistinguishability*, resolving an outstanding open problem posed by Gagliardoni *et al.* [GHS16]. Furthermore, to justify our formalization, we show that our notions

- are all closed under composition (see Theorem 1 and Theorem 4).
- are strictly stronger than previous notions with classical challenge queries (see Theorem 2 and Theorem 6). In particular, this shows the (in)security of various symmetric encryption schemes including stream cipher and some block cipher modes of operation such as CFB, OFB, CTR. This even extends to authenticated encryption, in which some most widely used encryption modes like GCM are also resulting in an insecure scheme.
- (when restricted to classical challenge queries) are equivalent to Boneh-Zhandry's notions [BZ13b].

In this work, we adopt the real-or-random security definition. At first glance, the naive attempt is to use the compressed oracle in the challenge phase, with $f = \mathsf{Enc}$ in the real-world, and $f = \mathsf{Enc} \circ \pi$ in the random-world for a random permutation π. However, keeping a database (in superposition) requires quantum memory. Let us consider a motivating example of quantum security: the *frozen smart-card* attacks [GHS16]. In this setting, the target is a purely classical device with no quantum memory, but a *quantum hacker* can trick the encryption chip into quantum behavior, which allows the adversary to query the target device (i.e., encryption) in superposition. Mapping this to the game-based definitions, the target device would play the role of the challenger. This shows that the requirement of quantum memory for the challenger could be artificial in the real world.

Here, we take an alternative approach. Informally, in the real game, the adversary has no restrictions on the use of the decryption oracle. Only in the random game, the challenge encryption oracle is implemented as a compressed oracle: it applies a random permutation π to the plaintext register before doing the encryption. For each decryption query, the challenger looks for the query's basis state in the database (in superposition) and if found, it reasonably guesses that the adversary is trying to decrypt the challenge ciphertext, and so it returns the adversary's original message (which is what is stored in the database). Otherwise, it decrypts normally. Intuitively, the security is established by the distinguishing probability of the adversary between whether its messages is encrypted with Enc or $\mathsf{Enc} \circ \pi$.

New Notions of Quantum Non-Malleability. We initiate the study of definitions of non-malleability for classical public-key encryption in the quantum world. This notion, first introduced by Dolev, Dwork and Naor [DDN00], is the strongest integrity-like notion that is achievable using public-key encryption only. The goal of the adversary, given a ciphertext y, is not to learn something about its plaintext x, but rather to output a different ciphertext y' such that its plaintext x' is "meaningfully related" to x. In the classical setting, the notion of non-malleability has been formalized using different definitional approaches: the indistinguishability-based approach [BDPR98,BS06,PsV07] and the simulation-based approach [DDN00,BS06,PsV07]. In the scope of this paper, we give indistinguishability-based definitions (Definition 4) and leave the semantic-based approach, as well as their full characterization as a future work.

We show that our notions are closed under composition (Theorem 5) and we give the relations between indistinguishability and non-malleability notions (Figure 2).

Formal Proof for Quantum Security of Encrypt-then-MAC. Encrypt-then-MAC is a well-known paradigm first proven in [BN08] to show that the combination of an IND-CPA symmetric encryption scheme and a EUF-CMA secure MAC scheme yields an IND-CCA symmetric encryption scheme. As an application of our technique, we solve the open problem of the quantum security of Encrypt-then-MAC (EtM), which was asked by Boneh and Zhandry in [BZ13b]. The classical proof does not work directly, because in one step of the proof, the reduction needs to store all the encryption queries in order to answer decryption queries. This is crucial to reduce to the security of the underlying CPA encryption, without requiring the decryption oracle. Even in the weaker model of classical challenge queries with quantum learning queries, the proof is not obvious, because all encryption queries (including the ones in the learning phases) need to be recorded, and it is not straightforward to do so in the quantum setting. In Section 4.3, we show how to adapt our technique to prove the quantum security of EtM (Theorem 3)[1].

A Lifting Theorem for Public-Key Cryptosystems. Concerning the public-key setting, we propose a compiler that lifts any secure encryption scheme in the sense of [BZ13b] to an encryption scheme secure in the sense of our notions in Section 5.3 (Theorem 7). The compiler follows the classical hybrid encryption paradigm, where we encrypt the message with a one-time symmetric encryption which can be constructed from pseudorandom functions, and then encrypt the symmetric key with a secure public-key scheme (in the sense of [BZ13b]).

1.4 Related Work and Discussion

Boneh-Zhandry's Impossibility. A notion of fully quantum indistinguishability (BZ-IND-fqCPA) was previously proposed by Boneh and Zhandry [BZ13b], in which the adversary is allowed to send the two input-message superpositions in the challenge phase:

$$\sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0, x_1, y\right\rangle \mapsto \sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0, x_1, y \oplus \mathsf{Enc}(x_b)\right\rangle .$$

As already observed in [BZ13b], this notion is unachievable, even for CPA security. This shows that some seemingly natural notions of quantum indistinguishability are too strong and might go beyond what a meaningful notion of indistinguishability should achieve. Recall the basic security requirement of an encryption scheme: *the ciphertext does not leak any information about the message, except its length.* A scheme is broken if it leaks at least one bit of information to the adversary. However, any scheme would be broken in the sense

[1] We note that in [SJS16], the authors claim to have solved this question (in the weaker model proposed by Boneh-Zhandry [BZ13b]), but a closer look at their proof shows that they make certain implicit assumptions on the ability to record quantum encryption queries, which was not known how to do at that time.

of BZ-IND-fqCPA notion even the adversary has no information whatsoever. To put it another way, there are encryption schemes that are insecure in the sense of BZ-IND-fqCPA but may be arguably secure for practical use.

Notice that Boneh-Zhandry's impossibility uses Fourier sampling (i.e., apply Hadamard transform and then measure) to distinguish between a pair of plaintext and ciphertext registers that are entangled after the encryption, and another separate plaintext register. In our security model, there is only a single plaintext register, which is always entangled with the ciphertext register after the query. This thwarts Boneh-Zhandry's attack and allows us to bypass their impossibility[2].

Comparison with [GHS16]'s Quantum Indistinguishability. In [GHS16], Gagliardoni, Hülsing and Schaffner discuss a "security tree" of some possible choices for quantum indistinguishability against quantum chosen-plaintext attacks (in regard to symmetric encryption cryptosystems), and analyze some meaningful notions.

On a high-level, their security tree is built from different perspectives of how we model the interaction between the adversary and the challenger. In particular, the candidates are: 1) how the challenge phase is implemented: a black-box unitary (the oracle model) or an external challenger (the challenger model) who has additional input and/or output out of $\mathcal{A}$'s control; 2) how the adversary sends its challenge queries: by a quantum state or by a classical description (which prevents the entanglement of the messages with other registers); 3) the availability of the plaintext registers after the query; and 4) the query model: the standard query model or the minimal oracle model (see [KKVB02]).

Since we aim to achieve the strongest possible security notions, we focus our discussion on the first and the fourth point, and stay on the safe-side for the others (that is, we choose the models that give the adversary more power). This can be seen in our definition, as we use the oracle model in the real game and the challenger model in the random game. However, as we will show, the two type of models are completely equivalent.

[GHS16]'s strongest notion is the one considered in the minimal oracle model, which is a map $|x\rangle \mapsto |\mathsf{Enc}(x)\rangle$. They show that with the decryption oracle, minimal oracle can be efficiently simulated by standard oracles. However, we stress that in general, unlike the symmetric setting, in the public-key setting, the requirement of having the decryption key simultaneously with the public key is unrealistic. The encryption machine should not hold the secret key for practical use. Another important reason is that this approach is only applicable to injective functions, which definitely does not include decryption. We note that one can use our technique for the minimal encryption oracle and the standard decryption oracle. For consistency, we decide to stick with only the standard query model.

[2] In [CETU20], the authors study all possible IND-CPA security notion for symmetric encryption schemes against a quantum adversary and their relations to each other. The quantum real-or-random indistinguishability is one of the strongest and at the same time realizable security notions studied in [CETU20].

Quantum Encryption Approaches [AGM18]. In this paper, on a high level, an adversary $\mathcal{A}$ has negligible probability in distinguishing between two experiments: in the real one, it has access to encryption and decryption oracles with no restrictions, whereas in the random one, the challenge encryption oracle replaces $\mathcal{A}$'s queried plaintexts by random ones (half of a maximally-entangled state), and the decryption oracle answers with the originally queried plaintexts if the adversary asked for decryption of a challenge ciphertext (which can be done by first decrypting the ciphertext and applying a measurement on the entangled state), otherwise it answers normally. However, in the context of classical encryption, the standard oracle for encryption is a map $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$ where f is the encryption algorithm. The adversary can then use the same strategy to detect the random experiment's simulation: it prepares a maximally-entangled state $|\phi^+\rangle_{XX'}$ and uses half of it (the registers X) as the challenge plaintext, and keeps X'. After receiving the challenge ciphertext, it measures the plaintext registers and X', and trivially distinguishes whether it is in the random experiment. We note that this attack cannot be performed without relaying, that is the plaintext registers X need to be available to $\mathcal{A}$ after the challenge encryption. However, non-relaying is indistinguishable from being traced out the plaintext registers (from $\mathcal{A}$'s perspective). This inherently reduces to a definition with classical challenge queries, which defeats our goals[3].

Related Work. The real-or-random approach that we use here was first proposed by Mossayebi and Schack in [MS16] (in which they call "real-or-permuted") for defining quantum security for symmetric encryption. Up to some small modifications in the formalization, their notion for chosen-plaintext security and ours are equivalent. However, they have not overcome the main obstacle in defining notions for adaptive chosen-ciphertext security, that is, how the challenger can check if a quantum decryption query submitted by the adversary is not "related" to the challenge queries. Instead, in their definition, the adversary is imposed by a restriction that it cannot submit such decryption queries, which cannot be verified by the challenger. Without being able to verify that the adversary follows the restriction, the definition is meaningless because the adversary can trivially break security without the challenger being aware of it. In our paper, we explicitly show how to impose this restriction on the adversary, and present a meaningful quantum counterpart of chosen-ciphertext security.

Concurrent Work. In concurrent and independent work, Gagliardoni, Krämer, and Struck [GKS20] propose alternative quantum security notions for public-key encryption against *chosen-plaintext attacks.* Their security notions extend the results from [GHS16] to the public-key scenario, using the minimal oracle model. Most importantly, they show that for many real-world public-key encryption schemes, the minimal oracle for encryption can be implemented efficiently *without knowledge of the secret key* (i.e., without the decryption oracle). Recall that in general, one would need both the standard encryption oracle and decryption oracle to efficiently implement a minimal encryption oracle.

[3] A similar discussion also appeared in [GHS16].

Their notions and ours are inherently incomparable due to the difference in how we model quantum oracles access. We leave the problem of unifying these security notions for future study.

2 Preliminaries

2.1 Notations

Let $\lambda \in \mathbb{N}$ be the security parameter. We say that a function is *negligible* in λ if it is a $f(\lambda) = \lambda^{-\omega(1)}$. When sampling uniformly at random a value a from a set $\mathcal{U}$, we employ the notation $a \overset{\$}{\leftarrow} \mathcal{U}$. When sampling a value a from a probabilistic algorithm $\mathcal{A}$, we employ the notation $a \leftarrow \mathcal{A}$. For $a \in \mathbb{N}$, $[a] = \{x \in \mathbb{N} \mid x \leq a\}$ will denote the closed integer interval with endpoints 0 and a. Let $|\cdot|$ denote either the length of a string, or the cardinal of a finite set, or the absolute value. By PPT we mean a polynomial-time non-uniform family of probabilistic circuits, and by QPT we mean a polynomial-time non-uniform family of quantum circuits. Let $\delta_{x,x'}$ denote the Kronecker delta function.

2.2 Quantum Computing

For notation and conventions regarding quantum information, we refer the reader to [NC11]. We recall a few basics here. We let $|\phi\rangle$ denote an arbitrary pure quantum state, let $|x\rangle$ denote an element of the standard (computational) basis. A mixed state will be denoted by lowercase Greek letters, e.g., ρ. We let $|+\rangle$ denote the uniform superposition, that is $|+\rangle \coloneqq \sum_x |x\rangle$.

A pure state $|\phi\rangle$ can be manipulated by performing a unitary transformation U to the state $|\phi\rangle$, which we denote $U|\phi\rangle$. The identity on a n-bit quantum system is denoted $\mathbb{1}_n$. Given two quantum systems A, B, with corresponding Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, let $|\phi\rangle = |\phi_0, \phi_1\rangle$ be a state of the joint system. We write $U^A|\phi\rangle$ to denote that we act with U on register A, and with identity $\mathbb{1}$ on register B, and we write U^{AB} to denote that we act with U on both registers A, B simultaneously, that is $U^{AB} = U^A \otimes U^B$.

Partial Measurement. Given two quantum systems A, B, with corresponding Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, let ρ_{AB} be the density matrix of the joint system. We write $\mathrm{Tr}_B(\rho_{AB})$ for the state obtained by tracing out system A.

Quantum Computations. Let Q be a n-bit quantum system over $\mathbb{Z}_q$ for some integer q. The Quantum Fourier Transform (QFT) performs the following operation efficiently:

$$\mathsf{QFT}|x\rangle \coloneqq \frac{1}{\sqrt{q^n}} \sum_{y \in \{0,1\}^n} \omega_q^{x \cdot y} |y\rangle ,$$

where $\omega_q \coloneqq \exp(\frac{2\pi i}{q})$, and $x \cdot y$ denotes the dot product. In this paper, we usually consider $q = 2$, so that $\omega_q = (-1)$.

Given a function $f : \mathcal{X} \to \mathcal{Y}$, we model a quantum-accessible oracle $\mathcal{O}$ for f as a unitary transformation $\mathcal{O}_f$ acting on three registers X, Y, Z with the property that $\mathcal{O}_f : |x, y, 0\rangle \mapsto |x, y \oplus f(x), 0\rangle$, where $\oplus$ is some involutive group operation (so-called quantum query model). Given an algorithm $\mathcal{A}$, as access to oracles $\mathcal{O}_i$, we sometime write $y \leftarrow \mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\cdots}(x)$ for the event that a quantum adversary $\mathcal{A}$ takes x as input, makes quantum queries to $\mathcal{O}_1, \mathcal{O}_2, \ldots$, and finally outputs y.

2.3 Cryptosystems and Notions of Security

Here we briefly recall standard notations of classical cryptosystems [Gol04], see Appendix A for complete definitions.

Symmetric-key Encryption. A symmetric-key cryptosystem $\mathcal{SE}$ consists of three PPT algorithms $\mathcal{SE} = (\mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec})$.

The standard correctness requirement is that for any $\mathtt{k} \leftarrow \mathcal{K}()$, any random coin r of SymEnc and any $x \in \mathcal{X}$, we have $\mathsf{SymDec}_{\mathtt{k}}(\mathsf{SymEnc}_{\mathtt{k}}(x; r)) = x$. We sometimes omit the randomness r in SymEnc.

Public-key Encryption. A public-key cryptosystem $\mathcal{E}$ consists of three PPT algorithms $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$.

The following correctness definition is taken from [HHK17]. We call a public-key encryption scheme $\mathcal{E}$ is *δ-correct* if

$$\mathbb{E}\left[\max_{x\in\mathcal{X}, r\in\mathcal{R}} \Pr\left[\mathsf{Dec}_{\mathtt{sk}}(\mathsf{Enc}_{\mathtt{pk}}(x; r)) \neq x\right]\right] \leq \delta,$$

where the expectation is taken over $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$.

Game-based Definitions. Previously, quantum indistinguishability for adaptive chosen-ciphertext security has been defined in the work of Boneh and Zhandry [BZ13b]. At a high level, their notions allow quantum encryption and decryption queries, but require challenge queries to be *classical*. Regarding the attack models, the following security notions are then defined: IND-qCPA, IND-qCCA1, IND-qCCA2. We briefly recall their definitions in Appendix A.

Message Authentication Code. A message authentication code $\mathcal{MA}$ consists of three PPT algorithms $\mathcal{MA} = (\mathcal{K}, \mathsf{MAC}, \mathsf{Ver})$. We require that $\mathsf{Ver}_{\mathtt{k}}(x, \tau) = 1$ if and only if $\tau = \mathsf{MAC}_{\mathtt{k}}(x)$, for all $x \in \mathcal{X}$ and for all $\mathtt{k} \leftarrow \mathcal{K}()$. We also require that for all $\mathtt{k} \leftarrow \mathcal{K}()$ and for all τ, we have $\mathsf{Ver}_{\mathtt{k}}(x, \tau) = 0$ if $x = \perp$.

In this work, we use Boneh-Zhandry security for a MAC scheme [BZ13a]: a MAC is EUF-qCMA secure, if no adversary can use q quantum queries to the MAC to produce $q + 1$ valid message/tag pairs except with negligible probability.

3 Quantum Oracle Queries Recording Framework

The starting point towards our goal of defining indistinguishability-based security notions for encryption is to explain how the challenger should reply to the quantum decryption queries in the second learning phase after the adversary has made the quantum encryption queries in the challenge phase. This implies explaining how it could record these quantum challenge queries.

We give these explanations in this section in the more general case of a function f (later instantiated by the encryption algorithm) accessible via an oracle $\mathcal{O}_f$ and its inverse f^{-1} (which models the decryption algorithm) accessible via an oracle $\mathcal{O}_{f^{-1}}$. We first describe in Section 3.1 how to record the quantum queries to $\mathcal{O}_f$, and in Section 3.2 how to answer the inverse queries to $\mathcal{O}_{f^{-1}}$ using the constructed database.

3.1 How to Record Quantum Oracle Queries?

We formalize the technique by considering some probabilistic quantum-polynomial-time computable function $f : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$, in the quantum query model, where an adversary $\mathcal{A}$ is given black-box access to f via an oracle $\mathcal{O}_f$. By probabilistic, we mean that a new random coin is flipped independently and uniformly for each query. On a high-level, in order to record quantum queries to f, we will give the adversary oracle access to a *compressed oracle*, denoted CStO_f, that we construct in this section and that is perfectly indistinguishable from $\mathcal{O}_f$.

For simplicity, we assume that the domain of f is $\mathcal{X} = \{0,1\}^m$, its range is $\mathcal{Y} = \{0,1\}^n$, and the randomness space $\mathcal{R} = \{0,1\}^\ell$. We also make a convention that $f(\perp) = 0$ where $\perp$ is a bit string that lies outside the message space ($\perp \notin \mathcal{X}$).

Single-query Setting. We first describe the oracle operations handling a single query. We begin with the usual quantum oracles, the so-called *standard oracle* and *Fourier oracle*. These oracles and their equivalence are proven in much of literature on quantum-accessible oracles (e.g., see [KKVB02,Zha19,CMSZ19]). The standard oracle for a function f is the unitary defined as:

$$\mathcal{O}_f \sum_{x,y} \alpha_{x,y} \left|x, y\right\rangle_{XY} \mapsto \sum_{x,y} \alpha_{x,y} \left|x, y \oplus f(x)\right\rangle_{XY} .$$

The function f is computed by first sampling a uniform and independent randomness r and then applying f on the input $(x; r)$. From the adversary's point of view, this is equivalent to r being in uniform superposition $\sum_r |r\rangle$ and perform the following map: $|x, y\rangle_{XY} \otimes \sum_r |r\rangle_R \mapsto \sum_r |x, y \oplus f(x; r)\rangle_{XY} |r\rangle_R$. Augmenting the joint system with an additional register R is a *purification* of the adversary's mixed state, and tracing out R (i.e., projecting onto the one-dimensional subspace spanned by $|r\rangle$) recovers the original mixed state. Moreover, this projection, which is outside of the adversary's view, is undetectable by $\mathcal{A}$.

Next, we consider the oracle in the truth table form: for each query, its internal state consists of two sets of registers: ℓ-qubit R registers representing randomness to the function, $n2^m$-qubit F registers containing the truth table of the function with the given randomness. For short, we write $|r\rangle \left|f(0; r)\| \dots \| f(2^m - 1; r)\right\rangle$ as $|r\rangle |D_r\rangle$. D_r can be seen as the truth table of f, where $D_r(x) := f(x; r)$. We call this oracle StO_f, which performs the map (on the adversary's basis states):

$$\mathsf{StO}_f \left|x,y\right\rangle_{XY} \otimes \frac{1}{\sqrt{2^\ell}} \sum_r \left|r\right\rangle_R \left|D_r\right\rangle_F \mapsto \frac{1}{\sqrt{2^\ell}} \sum_r \left|x, y \oplus D_r(x)\right\rangle_{XY} \left|r\right\rangle_R \left|D_r\right\rangle_F$$
$$= \frac{1}{\sqrt{2^\ell}} \sum_r \left|x, y \oplus f(x;r)\right\rangle_{XY} \left|r\right\rangle_R \left|D_r\right\rangle_F$$

Lemma 1. *$\mathcal{O}_f$ and StO_f are perfectly indistinguishable.*

Proof. The lemma follows directly from the fact that for each query, if we trace out the oracle's internal registers, the mixed state of the adversary in both cases will be identical. □

Next, we consider the Fourier oracle model $\mathsf{FourierO}_f$, which technically provides a different interface to the adversary, but can be mapped to the standard oracle by QFT operations. The initial state of $\mathsf{FourierO}_f$ is

$$\frac{1}{\sqrt{2^\ell}} \sum_r \left|r\right\rangle_R \mathsf{QFT}^F \left|D_r\right\rangle_F = \frac{1}{\sqrt{2^\ell}} \frac{1}{\sqrt{2^{n2^m}}} \sum_r \sum_E (-1)^{E \cdot D_r} \left|r\right\rangle_R \left|E\right\rangle_F .$$

On the basis states, the Fourier oracle $\mathsf{FourierO}_f$ is defined as follows.

$$\mathsf{FourierO}_f \left|x,z\right\rangle_{XY} \otimes \frac{1}{\sqrt{2^\ell}} \frac{1}{\sqrt{2^{n2^m}}} \sum_r \sum_E (-1)^{E \cdot D_r} \left|r\right\rangle_R \left|E\right\rangle_F$$
$$\mapsto \frac{1}{\sqrt{2^\ell}} \frac{1}{\sqrt{2^{n2^m}}} \sum_r \sum_E (-1)^{E \cdot D_r} \left|x,z\right\rangle_{XY} \left|r\right\rangle_R \left|E \oplus P_{x,z}\right\rangle_F .$$

where $P_{x,z}$ is the point function that outputs z on x and 0 everywhere else. Intuitively, with the Fourier oracle, instead of adding data from the oracle's registers to the adversary's registers, it adds in the opposite direction.

Lemma 2. *$\mathsf{FourierO}_f$ and StO_f are equivalent.*

Proof. Each can be constructed by an f-independent quantum circuit containing just one copy of the other, that is

$$\mathsf{QFT}^{YF} \circ \mathsf{StO}_f \circ \mathsf{QFT}^{\dagger YF} = \mathsf{FourierO}_f,$$
$$\mathsf{QFT}^{\dagger YF} \circ \mathsf{FourierO}_f \circ \mathsf{QFT}^{YF} = \mathsf{StO}_f.$$

For completeness, we provide concrete computations for these equalities in Appendix B.1. □

We now describe our compressed oracles. Consider the Fourier oracle $\mathsf{FourierO}_f$, after the query, the oracle's F registers will be XORed with a point function. If we transform this register back to the computational basis (by applying QFT operations), and un-compute f, F will be zero in all but at most one location. Therefore, we can discard all except at most one pair (x,z) such that $z \neq 0$. We then obtain the compressed Fourier oracle for f, whose initial state is empty. The action of the compressed oracle and its indistinguishability from the Fourier oracle are formally stated as follows.

Lemma 3. *In the single-query setting, the compressed Fourier oracle $\mathsf{CFourierO}_f$ acts on a quantum state $\left|x,z\right\rangle$ where $x \in \mathcal{X}$ and $z \in \mathcal{Y}$, as follows.*

- *If $z = 0^n$, then* $\mathsf{CFourierO}_f \left|x, z\right\rangle \mapsto \left|x, z\right\rangle \frac{1}{\sqrt{2^\ell}} \sum_r \left|r\right\rangle$.
- *If $z \neq 0^n$, then* $\mathsf{CFourierO}_f \left|x, z\right\rangle \mapsto \left|x, z\right\rangle \otimes \left|\phi_{x,z}\right\rangle$*, where*

$$\left|\phi_{x,z}\right\rangle \coloneqq \frac{1}{\sqrt{2^\ell}} \sum_r \sum_u \frac{1}{\sqrt{2^n}} (-1)^{f(x;r)\cdot u} \left|x, r, z \oplus u\right\rangle .$$

Furthermore, $\mathsf{CFourierO}_f$ *and* $\mathsf{FourierO}_f$ *are perfectly indistinguishable.*

Proof Sketch. Let U_f be the unitary implementing f, that is: $U_f \left|r, x, u\right\rangle = \left|r, x, u \oplus f(x;r)\right\rangle$. We define a two-input conditional unitary Decomp_x acting on R and n-qubit registers, based on x, as $\mathsf{Decomp}_x \coloneqq \mathsf{QFT} \circ U_f$. Let Decomp to be the unitary which applies Decomp_x for all x in the domain, one at a time from 0 up to $2^m - 1$.

1. We start with the Fourier oracle, whose initial state is Fourier-transformed truth table of f. Then, the initial state of the oracle can be prepared by first initializing the uniform superposition R registers, and applying Decomp to x, R and F. This action can be seen as a sort of local decompression for F.
2. Next, we apply the Fourier oracle operation, which replaces the x-th row of the truth table F with some $u \oplus z$.
3. Finally, we reverse the oracle's initialization step (i.e., $\mathsf{Decomp}^\dagger$). Notice that $\mathsf{Decomp}_{x'}$ and the Fourier oracle operation commute for any x' different from the input x, so that the two applications of $\mathsf{Decomp}_{x'}$ and $\mathsf{Decomp}_{x'}^\dagger$ cancel out. A computational basis test on the x'-th row of the F registers to test if they contain 0 will always return true for any $x' \neq x$, thus these registers can be discarded. This means that after the query, the oracle's F registers only have at most 1 defined point. This action can be seen as a sort of local re-compression for F.
4. Now, if we change the representation of the F registers from the truth table to a database D, and decompress D again, we would obtain the compressed Fourier oracle.

The formal proof is given in Appendix B.2.

Notice that from its description above, we implement this oracle with three computations of U_f (by three computations of Decomp_x). However, as we will see in later sections, it is crucial for our security reductions to simulate $\mathsf{CFourierO}_f$ with only one computation of U_f, which allows us to "outsource" parts of U_f computations to other oracles. We now give an intuition why we can reduce three computations of U_f to one computation. Let's consider the following cases.

- The z register is all-zero. Note that since the initial state of the database D is also all-zero, applying Decomp_x in the first step (i.e., locally decompressing D) and then XORing z to the database's u register (in Step 2) does not change the database's state. Finally, locally re-compressing D (Step 3) brings it back to all-zero state, which can be discarded. At the end of this step, D is empty, thus decompressing it also yields an empty database. In this case, we see that we can skip Step 2, and two applications of Decomp_x in Step 1 and $\mathsf{Decomp}_x^\dagger$ in Step 3 cancel out, leaving us only one application of U_f in Step 4.

- The z register is not zero. By a similar argument, we have that at the end of Step 3, D has one defined point, thus the two applications of $\mathsf{Decomp}_x^\dagger$ in Step 3 and Decomp_x in Step 4 cancel out, leaving us only one application of U_f in Step 1.

We describe a quantum circuit in Figure 1, which applies a single computation of U_f, implementing our compressed oracle. Let Test be the unitary defined as $\mathsf{Test}\,|0\rangle\,|b\rangle \mapsto |0\rangle\,|b\rangle$ and $\mathsf{Test}\,|\phi\rangle\,|b\rangle \mapsto |\phi\rangle\,|b \oplus 1\rangle$ for any $|\phi\rangle$ orthogonal to $|0\rangle$ and $b \in \{0, 1\}$. A concrete computation (given in Appendix B.2) reveals that this circuit outputs the same quantum state as stated in the Lemma, when the oracle's registers are initialized with $|0\rangle$. □

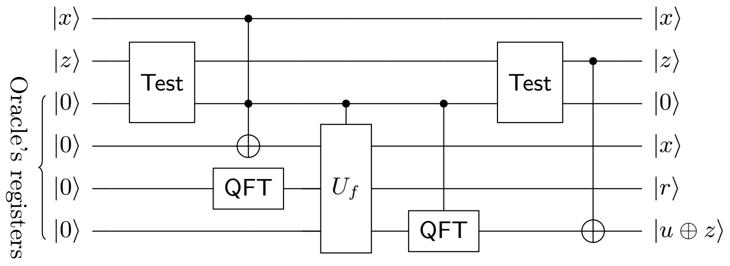

Fig. 1. A quantum circuit implementing our $\mathsf{CFourierO}_f$ oracle. Depending on the control bit b which is the output of Test, if $b = 1$, we apply U_f, otherwise, we apply the identity. The bit b will be discarded after the computation.

By applying QFT operations to the adversary's response registers, and the oracle's output registers, we restore the standard oracle, and the oracle's state will be (in superposition of) $|x, r, f(x; r)\rangle$. We will call this oracle CStO_f. The following Lemma follows immediately from Lemma 2 and Lemma 3. Note that since the transformation between CStO_f and $\mathsf{CFourierO}_f$ is independent of function f, CStO_f also applies U_f only once.

Lemma 4. CStO_f *is a* QPT *unitary operation. Furthermore,* CStO_f *and* StO_f *are perfectly indistinguishable.*

Many-query Setting. Since the functions we consider are randomized, the oracle would need to flip a new random coin for each query, that is to create a new register $\sum_r |r\rangle$ to purify the adversary's mixed state for each query. For the compressed oracles, it is equivalent to initiate a new, independent database. By the standard hybrid argument, it is easy to verify that:

Lemma 5. CStO_f *and* StO_f *are perfectly indistinguishable, in the many-query setting.*

For each query, its oracle's database is $|D_i\rangle := |r\rangle\,|x, f(x; r)\rangle$. Overall, the oracle's database D will be a collection of many tuples (r, x, y) where $(r, x, y) \in D$ means $y = f(x; r)$.

Notation. We formally define the following notation to denote the compressed oracle of a function f: $\mathsf{CStO}_f^{|r\rangle}[x \mapsto y]$, where r is the purification of the randomness of f, x is the input and y is the output. We sometimes omit the superscript if it is clear from the context.

Other Variations. We also consider some more oracle variations which give a different perspective on our compressed oracles. They will be used in the technical contributions presented in later sections.

Compress then Measure. We define the compress-then-measure oracle for a function f which is identical to $\mathsf{CStO}_f^{|r\rangle}$, except that after the query, we measure the purification registers (without measuring the whole database), and get some classical value r. We will denote this oracle as $\mathsf{CtMStO}_f^{|r\rangle}[x \mapsto y]$.

Lemma 6. *CtMStO_f and CStO_f are perfectly indistinguishable.*

Proof. We delay the measurement, and un-compute all the computations described above. We then end up with the standard oracle, which we can measure the purification registers as a part of the tracing out operation, and get a classical value r. From the adversary's point of view, it is undetectable. □

Randomness. We consider the following probabilistic function $f : \mathcal{X} \times \mathcal{R} \times \mathcal{K} \to \mathcal{Y}$, where both $\mathcal{R}$ and $\mathcal{K}$ act as randomness, which are controlled by the oracle. We consider the following implementations of the compressed oracle CStO_f:

- $\mathsf{CStO}_f^{|r\rangle}[(x;k) \mapsto y]$, which purifies the randomness $r \in \mathcal{R}$.
- $\mathsf{CStO}_f^{|k\rangle}[(x;r) \mapsto y]$, which purifies the randomness $k \in \mathcal{K}$.
- $\mathsf{CStO}_f^{|r,k\rangle}[x \mapsto y]$, which purifies both $(r,k) \in \mathcal{R} \times \mathcal{K}$.

Lemma 7. *$\mathsf{CStO}_f^{|r\rangle}[(x;k) \mapsto y]$, $\mathsf{CStO}_f^{|k\rangle}[(x;r) \mapsto y]$, and $\mathsf{CStO}_f^{|r,k\rangle}[x \mapsto y]$ are perfectly indistinguishable.*

Proof. The statement follows from the fact that the first and the third variations can be seen as compress-then-measure oracles of the second one. □

3.2 How to Answer Inverse Queries?

We now describe how to answer inverse queries to f^{-1} using the database constructed above. We first define a class of functions for which our technique will apply, and we call them *δ-almost invertible functions*. This notion captures a generic, broad class of cryptographic primitives including pseudorandom permutations and encryption. We then give the adversary access to a new oracle denoted $\mathsf{CInvO}_{f^{-1}}$ which acts on the database, instead of $\mathcal{O}_{f^{-1}}$. Given access to $\mathsf{CInvO}_{f^{-1}}$, the bound on the distinguishing probability of the adversary when interacting with the compressed oracle CStO_f is stated in Lemma 8.

Definition 1 **(δ-almost invertible Functions).** *Let $\mathcal{F} = \{f : \mathcal{X} \times \mathcal{K} \to \mathcal{Y}\}$ be a family of functions such that for each $f \in \mathcal{F}$, there is a function $f^{-1} : \mathcal{Y} \times \mathcal{K} \to \mathcal{X}$. $\mathcal{F}$ is δ-almost invertible if*

$$\mathop{\mathbb{E}}_{k \in \mathcal{K}} \left[\max_{x \in \mathcal{X}} \Pr\left[f^{-1}(k, f(k,x)) \neq x \right] \right] \leq \delta.$$

For example, a pseudorandom permutation, or a symmetric encryption scheme is an invertible function (with $\delta = 0$), whereas a δ-correct public-key encryption scheme is a δ-almost invertible function.

Let f be a QPT probabilistic and δ-almost invertible function. We define a classical procedure $\mathsf{FindImage}$ which takes as input an image $y \in \mathcal{Y}$, and a database D. Then, it looks for a triple $(x, r, y) \in D$. If found, it outputs $(b = 1, w = x)$, otherwise, it outputs $(b = 0, w = 0^m)$.

We define the unitary operation $\mathsf{CInvO}_{f^{-1}}$ for the inverse queries which maps the basis state $|y, z\rangle \otimes |D\rangle$ to:

$$\begin{cases} U_{f^{-1}} |y, z\rangle \otimes |D\rangle = |y, z \oplus f^{-1}(y)\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (0, 0^m), \\ |y, z \oplus w\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w). \end{cases}$$

This unitary is implemented by a single call to f^{-1}, controlled by the output bit b of $\mathsf{FindImage}$.

Lemma 8. *For any (unbounded) oracle algorithm $\mathcal{A}$:*

$$\left| \Pr\left[\mathcal{A}^{\mathcal{O}_f, \mathcal{O}_{f^{-1}}}() = 1 \right] - \Pr\left[\mathcal{A}^{\mathsf{CStO}_f, \mathsf{CInvO}_{f^{-1}}}() = 1 \right] \right| \leq O(q_i \cdot \delta),$$

where q_i is the number of inverse queries.

Proof. We prove through a sequence of games.

Game G_0: This is the game where $\mathcal{A}$ interacts with the standard oracles $\mathcal{O}_f$ and $\mathcal{O}_{f^{-1}}$.

Game G_1: This is identical to G_0, except that now the oracle $\mathcal{O}_f$ is simulated using the compressed oracle CStO_f. Notice that $\mathcal{O}_{f^{-1}}$ operation does not touch the database registers, thus it commutes with any CStO_f operation. Since CStO_f is equivalent to the standard oracle $\mathcal{O}_f$, $\mathcal{A}$ cannot distinguish G_1 and G_0.

Game G_2: This is identical to G_1, except that now the oracle $\mathcal{O}_{f^{-1}}$ is replaced by the oracle $\mathsf{CInvO}_{f^{-1}}$.

Let $|\Psi\rangle$ be the joint system state of the adversary and the oracle before making any inverse query. Denote $\Delta = \mathcal{O}_{f^{-1}} - \mathsf{CInvO}_{f^{-1}}$. For each query $|y, z\rangle$ to the inverse oracle, we consider the registers y, z, D. We now examine three cases.

(a) Let D be such that $y \notin D$, that is, $\mathsf{FindImage}(y, D) = (0, 0)$. Let P_1 be the projection onto the registers y, D such that $y \notin D$. In this case, the inverse oracle in both games applies the unitary mapping $|y, z\rangle \otimes |D\rangle \mapsto |y, z \oplus f^{-1}(y)\rangle \otimes |D\rangle$. Thus, $\Delta P_1 |\Psi\rangle = 0$.

(b) Let D be such that $y \in D$, that is, $\mathsf{FindImage}(y, D) = (1, w)$. Let P_2 be the projection onto the registers y, D such that $y \in D$ and $f^{-1}(y) = w$. In this case, we also have $\Delta P_2 |\Psi\rangle = 0$.

(c) Let D be such that $y \in D$. Let P_3 be the projection onto the registers y, D such that $y \in D$ but $f^{-1}(y) \neq w$. Thus $\left\| P_3 \left|\Psi\right\rangle \right\|^2$ is the probability of measuring y, D and get $y \notin D$ such that $f^{-1}(y = f(x)) \neq x$ for some preimage x of y. In this case, we have $\left\| \Delta P_3 \left|\Psi\right\rangle \right\|^2 \leq \delta$, by the definition of f.

Notice that $P_1 + P_2 + P_3 = \mathbb{1}$. Therefore, we have $\left\| \Delta \left|\Psi\right\rangle \right\|^2 \overset{(*)}{\leq} \left\| \sum_{i=1}^{3} \Delta P_i \left|\Psi\right\rangle \right\|^2 \overset{(**)}{\leq} 3 \cdot \sum_{i=1}^{3} \left\| \Delta P_i \left|\Psi\right\rangle \right\|^2 \leq 3 \cdot \delta$, where $(*)$ uses the triangle inequality and $(**)$ uses AM-QM inequality. Then the same holds true for any mixed state since any mixed state is in the convex hull of pure states. If $\mathcal{A}$ makes at most q_i inverse queries, the trace distance of the mixed state of the adversary in games G_2 and G_1 is at most $O(q_i \cdot \delta)$. This completes the proof. □

4 Quantum-Secure Symmetric Encryption

4.1 Definitions of Security

In this section, we use the compressed oracle technique defined above to define quantum real-or-random indistinguishability security notions. During the learning phases, $\mathcal{A}$ has access to the encryption standard oracle $\mathcal{O}_{\mathsf{SymEnc}_\mathsf{k}}$. In the CCA case, it also has access to $\mathcal{O}_{\mathsf{SymDec}_\mathsf{k}}$ in the first learning phase.

We now describe how we handle the challenge phase and the decryption queries in the second learning phase. Informally, in the real-world ($b = 1$), the adversary has no restrictions on the use of the decryption oracle (in particular, $\mathcal{A}$ can freely decrypt the challenge ciphertext), so that the encryption oracle is simply implemented as the standard encryption oracle $\mathcal{O}_{\mathsf{SymEnc}_\mathsf{k}}$ and the decryption oracle as the standard decryption oracle $\mathcal{O}_{\mathsf{SymDec}_\mathsf{k}}$. Only in the random-world ($b = 0$), the challenge encryption oracle is implemented as a compressed oracle which purifies the randomness r of the encryption: it applies a random permutation π to the plaintext register before doing the encryption, leading to $\mathsf{CStO}_{\mathsf{SymEnc}_\mathsf{k} \circ \pi}$[4]. Applying naively the recording technique of Section 3.2 with $f_\pi \coloneqq \mathsf{SymEnc}_\mathsf{k} \circ \pi$ would make the records in the database be of the form $(x, r, \pi, f_\pi(x; r))$ (where we consider π as a part of the randomness of f_π). Accordingly, the decryption oracle in the random-world would be implemented as $\mathsf{CInvO}_{f_\pi^{-1}} = \mathsf{CInvO}_{\pi^{-1} \circ \mathsf{SymDec}_\mathsf{k}}$. But since the permutation π is being chosen randomly for each invocation of the challenge encryption oracle, it can be chosen classically beforehand and there is no need to keep it in the database. Thus, it is easier (and equivalent) to consider that the decryption oracle in the random-world is implemented using $\mathsf{CInvO}_{\mathsf{SymDec}}$, with records in the database of the form $(x, r, f_\pi(x; r))$. This decryption oracle always returns the original plaintext (x) if the query is a challenge one. In both cases, the adversary outputs an internal state $\left|\Phi\right\rangle$ in the first phase, which will be given to the second phase.

Formally, let Π be a family of random permutations over $\mathcal{X}$, we define a "real-or-random" oracle allowing quantum queries as follows.

[4] For the sake of simplicity, we abuse notation here and denote $f_\pi \coloneqq \mathsf{SymEnc}_\mathsf{k} \circ \pi$, where one should understand $f_\pi(x; r) \coloneqq \mathsf{SymEnc}_\mathsf{k}(\pi(x); r)$ if $x \in \mathcal{X}$ and $r \in \mathcal{R}$.

$$\mathcal{RR}(b) = \begin{cases} \mathsf{SymEnc}_{\mathtt{k}} & \text{if } b = 1, \\ \mathsf{CStO}_{\mathsf{SymEnc}_{\mathtt{k}} \circ \pi} & \text{if } b = 0, \text{where } \pi \xleftarrow{\$} \Pi. \end{cases}$$

For the sake of simplicity, we overload the notation of $\mathsf{CInvO}_{\mathsf{SymDec}_{\mathtt{k}}}$ to denote both the standard oracle $\mathcal{O}_{\mathsf{SymDec}_{\mathtt{k}}}$ in the real-world (when there is no database D) and the actual $\mathsf{CInvO}_{\mathsf{SymDec}_{\mathtt{k}}}$ oracle in the random-world.

Definition 2 (Indistinguishability notions for symmetric encryption ($\mathsf{qIND\text{-}qCPA}, \mathsf{qIND\text{-}qCCA1}, \mathsf{qIND\text{-}qCCA2}$)). *Let $\mathcal{SE} = (\mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec})$ be a symmetric encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:*

```
Experiment $\mathsf{Expt}_{\mathcal{SE}}^{qind\text{-}qatk-b}(\lambda, \mathcal{A})$:
$\mathtt{k} \xleftarrow{\$} \mathcal{K}$
$|\Phi\rangle \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{SymEnc}_{\mathtt{k}}}, \mathcal{O}_1}(\lambda)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{RR}(b), \mathcal{O}_{\mathsf{SymEnc}_{\mathtt{k}}}, \mathcal{O}_2}(|\Phi\rangle)$
return $b'$
```

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{SymDec}_{\mathtt{k}}}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{SymDec}_{\mathtt{k}}}$	$\mathsf{CInvO}_{\mathsf{SymDec}_{\mathtt{k}}}$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{qind\text{-}qatk}(\lambda) := \left| \Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{qind\text{-}qatk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{qind\text{-}qatk-0}(\lambda, \mathcal{A}) = 1\right] \right|.$$

We say $\mathcal{SE}$ is secure in the sense of qIND-$\textsc{qATK}$ *if $\mathcal{A}$ being* QPT *implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{qind\text{-}qatk}(\lambda)$ is negligible.*

Equivalence with Boneh-Zhandry's Notions. To justify our notions, we show that when restricting our definitions to classical challenge queries, they are equivalent to Boneh-Zhandry's notions (IND-$\textsc{qATK}$). If we denote our restricted notions by IND-$\textsc{qATK}'$, a scheme $\mathcal{SE}$ is IND-$\textsc{qATK}'$ secure iff it is IND-$\textsc{qATK}$ secure.

Indeed, because the challenge queries are classical, the simulator can store the challenge plaintexts and the challenge ciphertexts. Any simulator that returns $\perp$ if the adversary submits a challenge ciphertext in the sense of IND-$\textsc{qATK}$ can be turned to a simulator that returns the original plaintext x in the sense of IND-$\textsc{qATK}'$, and vice versa. More precisely, we have that:

$$\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk}(\lambda) \leq 2 \cdot \mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk'}(\lambda), \text{ and } \mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk'}(\lambda) \leq \mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk}(\lambda).$$

This is the standard argument (see [BDJR97]), we omit the details.

Single-message Versus Many-message Security. We have presented definitions which allow the adversary to make $q(\lambda)$-many challenge queries to the real-or-random oracle. A scheme satisfying the definitions in the case when $q(\lambda) = 1$ is said to be *single-message* secure. The question of whether single-message security implies many-message security is the question of composability of the definitions, which is answered affirmatively below.

Theorem 1. *A symmetric encryption scheme $\mathcal{SE}$ is many-message* qIND-$\textsc{qATK}$ *secure iff it is single-message* qIND-$\textsc{qATK}$ *secure.*

The proof follows the classical hybrid argument; we give it in Appendix D.1.

4.2 A Separation Example

We show that upgrading from classical challenge queries to quantum challenge queries gives the adversary more power. In particular, we show that the IND-qCCA2 secure symmetric encryption scheme given by Boneh and Zhandry [BZ13b, Construction 4.9] is insecure once the adversary can make even a single quantum challenge query in the sense of chosen plaintext security (qIND-qCPA). Our attack can be considered as an impossibility to achieve quantum indistinguishability for encryption schemes which follow the stream cipher-like paradigm (such as stream ciphers, block cipher modes of operation including CFB, OFB, CTR, or even some most widely used modes like GCM for authenticated encryptions).

Theorem 2. *Under the assumption that quantum-secure pseudorandom functions exist, there is an encryption scheme $\mathcal{SE}$ which is* IND-qCCA2 *secure, but* qIND-qCPA *insecure.*

Proof. We recall Boneh-Zhandry construction as follows.

Construction 1 ([BZ13b]). *Let F and G be quantum-secure pseudorandom functions. We construct the following encryption $\mathcal{SE} = (\mathsf{SymEnc}, \mathsf{SymDec})$ where:*

```
$\mathsf{SymEnc}_{\mathtt{k}_1\|\mathtt{k}_2}(x)$ :
    $r \xleftarrow{\$} \{0,1\}^\lambda$
    $c_1 \leftarrow F(\mathtt{k}_1, r) \oplus x$
    $c_2 \leftarrow G(\mathtt{k}_2, (r, x))$
    return $r\|c_1\|c_2$

$\mathsf{SymDec}_{\mathtt{k}_1\|\mathtt{k}_2}(r\|c_1\|c_2)$ :
    $x \leftarrow c_1 \oplus F(\mathtt{k}_1, r)$
    $c_2' \leftarrow G(\mathtt{k}_2, (r, x))$
    if $c_2 \neq c_2'$ then
        return $\perp$
    return $x$
```

Lemma 9 ([BZ13b, Theorem 4.10]). *The encryption scheme $\mathcal{SE}$ in Construction 1 is* IND-qCCA2 *secure.*

To show the qIND-qCPA insecurity of this scheme, we establish the following quantum computation. Let U_{OTP} be the unitary implementing the one-time pad encryption, but using the same classical randomness r (which is uniformly chosen beforehand) in superposition. For fixed $x_0, x_1 \in \{0,1\}^m$, we prepare the following state:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \left(|x_0\rangle + |x_1\rangle\right) |0^m\rangle .$$

Applying U_{OTP} yields:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}} \sum_{b\in\{0,1\}} |x_b, x_b \oplus r\rangle .$$

Then we apply a Hadamard transform to $2m$ qubits in all the registers. This yields the state:

$$\begin{aligned}|\psi_3\rangle &= 2^{-\frac{2m+1}{2}} \sum_b \sum_{u\in\{0,1\}^m} (-1)^{x_b\cdot u} |u\rangle \sum_{v\in\{0,1\}^m} (-1)^{(x_b\oplus r)\cdot v} |v\rangle \\ &= 2^{-\frac{2m-1}{2}} \sum_{u,v} \delta_{u\cdot(x_0\oplus x_1), v\cdot(x_0\oplus x_1)} (-1)^{x_0\cdot u\oplus(x_0\oplus r)\cdot v} |u, v\rangle \,.\end{aligned}$$

If we measure $|\psi_3\rangle$, with probability 1, we get a random pair (u, v) such that

$$u \cdot (x_0 \oplus x_1) = v \cdot (x_0 \oplus x_1). \tag{1}$$

If we apply a random permutation π to the first registers x_b of $|\psi_1\rangle$ before applying U_{OTP} and then un-compute it, we get the following state:

$$|\psi_3\rangle = 2^{-\frac{2m-1}{2}} \sum_{u,v} \delta_{u\cdot(x_0\oplus x_1), v\cdot(\pi(x_0)\oplus\pi(x_1))} (-1)^{x_0\cdot u\oplus(\pi(x_0)\oplus r)\cdot v} |u, v\rangle \,.$$

Measuring $|\psi_3'\rangle$ yields a random pair (u, v) such that $u \cdot (x_0 \oplus x_1) = v \cdot (\pi(x_0) \oplus \pi(x_1))$ where $\pi(x_b)$ are random m-bit strings. Thus, Equation (1) satisfies with probability at most $\frac{1}{2}$. It is now easy to see that:

Lemma 10. *$\mathcal{SE}$ is* qIND-qCPA *insecure.*

Proof. In the challenge phase, the adversary $\mathcal{A}$ chooses two fixed messages x_0, x_1, and prepares the following state as its challenge:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |0\rangle_R |0\rangle_F |+\rangle_G \,.$$

The challenge ciphertext state will be:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |r\rangle_F |x_b \oplus F(\mathtt{k}_1, r)\rangle_F |+\rangle_G \text{ if } b = 0,$$

or

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |r\rangle_R |\pi(x_b) \oplus F(\mathtt{k}_1, r)\rangle_F |+\rangle_G \text{ if } b = 1.$$

Since r is a classical value, $\mathcal{A}$ can discard two registers R and G, which are separate from the others. $\mathcal{A}$ then applies the Fourier sampling (i.e., Hadamard transform followed by a measurement as described above), and outputs 1 if Equation (1) is satisfied, otherwise it outputs 0. We have $\Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{qind\text{-}qcpa-1}(\lambda, \mathcal{A}) = 1\right] = 1$ and $\Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{qind\text{-}qcpa-0}(\lambda, \mathcal{A}) = 1\right] \le \frac{1}{2}$, thus $\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{qind\text{-}qcpa}(\lambda) \ge \frac{1}{2}$, which is certainly not negligible. □

4.3 Quantum CCA2 Security of Encrypt-then-MAC

The classical Encrypt-then-MAC paradigm [BN08] shows that an $\mathsf{IND\text{-}CPA}$ secure symmetric encryption scheme can be made $\mathsf{IND\text{-}CCA2}$ secure if combined with an $\mathsf{EUF\text{-}CMA}$ MAC scheme. We prove here that this result also holds for the security notions defined above. We first recall the Encrypt-then-MAC construction and proceed with its security proof.

Construction 2. *Let $\mathcal{SE} = (\mathcal{K}_{\mathcal{SE}}, \mathsf{SymEnc}, \mathsf{SymDec})$ be a symmetric encryption scheme and $\mathcal{MA} = (\mathcal{K}_{\mathcal{MA}}, \mathsf{MAC}, \mathsf{Ver})$ be a MAC scheme. The Encrypt-then-MAC composition of base schemes $\mathcal{SE}$ and $\mathcal{MA}$ is the symmetric encryption scheme $\mathcal{SE}' = (\mathcal{K}', \mathsf{SymEnc}', \mathsf{SymDec}')$ whose constituent algorithms are defined as follows.*

```
$\mathcal{K}'(\lambda)$ :
  $\mathtt{k}_1 \xleftarrow{\$} \mathcal{K}_{\mathcal{SE}}()$
  $\mathtt{k}_2 \xleftarrow{\$} \mathcal{K}_{\mathcal{MA}}()$
  return $\mathtt{k}_1 \parallel \mathtt{k}_2$

$\mathsf{SymEnc}'_{\mathtt{k}_1\|\mathtt{k}_2}(x)$ :
  $c \leftarrow \mathsf{SymEnc}_{\mathtt{k}_1}(x)$
  $\tau \leftarrow \mathsf{MAC}_{\mathtt{k}_2}(c)$
  return $c \parallel \tau$

$\mathsf{SymDec}'_{\mathtt{k}_1\|\mathtt{k}_2}(c \parallel \tau)$ :
  if $\mathsf{Ver}_{\mathtt{k}_2}(c, \tau) = 0$ then
    return $\perp$
  $x \leftarrow \mathsf{SymDec}_{\mathtt{k}_1}(c)$
  return $x$
```

Theorem 3. *Let $\mathcal{SE}$ be an $\mathsf{qIND\text{-}qCPA}$ secure symmetric encryption scheme. Let $\mathcal{MA}$ be an $\mathsf{EUF\text{-}qCMA}$ secure message authentication code. Then the encryption scheme $\mathcal{SE}'$ defined in Construction 2 is $\mathsf{qIND\text{-}qCCA2}$ secure. In particular, for any QPT adversary $\mathcal{A}$ making at most q_d decryption queries, there exist QPT adversaries $\mathcal{B}, \mathcal{C}$ such that*

$$\mathsf{Adv}^{qind\text{-}qcca2}_{\mathcal{A},\mathcal{SE}'}(\lambda) \leq \mathsf{Adv}^{qind\text{-}qcpa}_{\mathcal{B},\mathcal{SE}}(\lambda) + O(q_d) \cdot \mathsf{Adv}^{euf\text{-}qcma}_{\mathcal{C},\mathcal{MA}}(\lambda).$$

Remark 1. As demonstrated by Boneh and Zhandry [BZ13a], $\mathsf{EUF\text{-}qCMA}$ MACs can be constructed from quantum-secure pseudorandom functions. For $\mathsf{qIND\text{-}qCPA}$ security, it is not difficult to prove that the encryption scheme from [GHS16, Construction 6.6], which is based on quantum-secure pseudorandom permutations, is also secure in our notions.

Proof. We proceed using hybrid games. For the sake of clarity, these games are also described in Figure 4.

Let $\mathcal{A}$ be a QPT adversary. For any game G_{index}, we denote by $\Pr[G_{\mathsf{index}}] \coloneqq |\Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 1] - \Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 0]|$. Also, by event $G_{\mathsf{index}}(\mathcal{A})$, we mean the output of the experiments (defined as in Definition 2) in game G_{index} when interacting with $\mathcal{A}$.

Game G_0: This is the standard attack game. In what follows, let $\mathtt{k} \coloneqq \mathtt{k}_1 \| \mathtt{k}_2 \leftarrow \mathcal{K}'()$.

Game G_1: This is identical to G_0, except that

- The encryption oracle is implemented as $\mathsf{CStO}^{|r\rangle}_{\mathsf{SymEnc}'_{\mathtt{k}}}$, which purifies the randomness r of SymEnc.

- The challenge oracle (in the real-world experiment $b = 1$) is also implemented as $\mathsf{CStO}^{|r\rangle}_{\mathsf{SymEnc'}_{\mathtt{k}}}$.
- The decryption oracle is implemented as $\mathsf{CInvO}_{\mathsf{SymDec'}_{\mathtt{k}}}$.

In particular, let D be the database of both the encryption oracle and the challenge oracle. The decryption oracle $\mathsf{CInvO}_{\mathsf{SymDec'}_{\mathtt{k}}}$ can be written as:

$$\mathsf{CInvO}_1 \left|y, z\right\rangle \left|D\right\rangle = \begin{cases} \left|y, z \oplus \mathsf{SymDec'}_{\mathtt{k}}(y)\right\rangle \left|D\right\rangle & \text{if } \mathsf{FindImage}(y, D) = (0, 0), \\ \left|y, z \oplus w\right\rangle \left|D\right\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w), \end{cases}$$

where $\mathsf{FindImage}$ parses its input component y as $y = c \parallel \tau$.

Claim 3.1. $\Pr[G_1] = \Pr[G_0]$.

Proof. Since the encryption is perfectly correct, by Lemma 8, we have $\Pr[G_1] = \Pr[G_0]$.

Game G_2: This is identical to G_1, except that we change the decryption oracle $\mathsf{CInvO}_{\mathsf{SymDec'}_{\mathtt{k}}}$ to:

$$\mathsf{CInvO}_2 \left|y, z\right\rangle \left|D\right\rangle = \begin{cases} \left|y, z \oplus \perp\right\rangle \left|D\right\rangle & \text{if } \mathsf{FindImage}(y, D) = (0, 0)^{5}, \\ \left|y, z \oplus w\right\rangle \left|D\right\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w). \end{cases}$$

Claim 3.2. $|\Pr[G_2] - \Pr[G_1]| \leq q_d \cdot \mathsf{Adv}^{euf\text{-}qcma}_{\mathcal{A},\mathcal{MA}}(\lambda)$.

Proof. Intuitively, if the adversary could distinguish the two games, it must be able to procedure a ciphertext $(c\|\tau) \notin D$ such that $\mathsf{SymDec'}_{\mathtt{k}}(c\|\tau) \neq \perp$. Since the decryption of the underlying encryption $\mathcal{SE}$ never outputs $\perp$, it means that $\mathsf{Ver}_{\mathtt{k}_2}(c, \tau) = 1$. But since $(c\|\tau) \notin D$, this (c, τ) pair must be a forge of $\mathcal{MA}$.

Formally, we construct a QPT adversary $\mathcal{C}$ from $\mathcal{A}$ that makes only q quantum queries to the oracle MAC, and successfully outputs $q + 1$ valid, distinct message/tag pairs. $\mathcal{C}$ runs $\mathcal{A}$ as its subroutine: it samples a random key $\mathtt{k}_1 \xleftarrow{\$} \mathcal{K}_{\mathcal{SE}}$ for $\mathcal{SE}$ and simulates the encryption oracle for $\mathcal{A}$ by implementing $\mathsf{CStO}^{|r\rangle}_{f}[x \mapsto c\|\tau]$ where $f = \mathsf{SymEnc'}$ and r is the randomness of $\mathcal{SE}$. For the i-th encryption query (including the challenge queries), after computing a ciphertext $c_i = \mathsf{SymEnc}_{\mathtt{k}_1}(x_i)$, it sends c_i to the MAC oracle and receives back a tag τ_i for c_i, and forwards $(c_i\|\tau_i)$ to $\mathcal{A}$. For the sake of completeness, a quantum circuit of this unitary is given in Figure 5. Let D be the database kept by $\mathcal{C}$, which is a collection of q pairs $\{(x_i, (c_i\|\tau_i))\}_{i\in[q]}$, where q is the total number of encryption and challenge queries. $\mathcal{C}$ randomly measures one of $\mathcal{A}$'s decryption queries, and then it also measures its database.

[5] Here we use the same definition for $z \oplus \perp$ as in [BZ13b]. More precisely, we take $\perp$ to be some bit string that lies outside of the message space $\mathcal{X}$, and $z \oplus \perp$ to be bitwise XOR.

Let Forge_2 be the event that $\mathcal{A}$ puts a non-negligible weight to a pair $(\widetilde{c}\|\widetilde{\tau})$ in its decryption queries such that $(\widetilde{c}\|\widetilde{\tau}) \notin D$ and $\mathsf{SymDec}'(\widetilde{c}\|\widetilde{\tau}) \neq \perp$. $\mathcal{A}$ could distinguish the two games if and only if $\Pr[\mathsf{Forge}_2]$ is non-negligible.

Now we argue that measuring the database D gives $\mathcal{C}$ q valid, distinct pairs (c_i, τ_i) with overwhelming probability.

Without loss of generality, we assume that the adversary $\mathcal{A}$ never sends a query $\sum_x \alpha_x |x, +\rangle$ to the compressed oracles (since it is equivalent to not making any query at all), and $\mathcal{C}$ only needs to make a query to the MAC oracle when $\mathcal{A}$ does so[6]. Thus, we have $|D| = q$ which is also the number of legitimate queries to the MAC oracle. Since each ciphertext $(c_i\|\tau_i) \in D$ is generated with a random, fresh coin, the probability that there exists a collision among any of the c_i is at most $q^2/|\mathcal{C}|$ where $\mathcal{C}$ is $\mathcal{SE}$'s ciphertext space. This quantity is negligible, so $\mathcal{C}$ gets q distinct pairs except with negligible probability after measuring its database. All these pairs are legitimately computed by the MAC oracle, hence they are valid.

$\mathcal{C}$ then outputs its q pairs from the database measurement, and another pair from its decryption queries measurement. With probability $\Pr[\mathsf{Forge}_2]/q_d$, it gets a *valid* pair $(\widetilde{c}\|\widetilde{\tau}) \notin D$, which is distinct from the other q pairs. Overall, $\mathcal{C}$ output $q+1$ distinct, valid pairs with non-negligible probability when Forge_2 happens with non-negligible probability. The claim follows from the security of $\mathcal{MA}$.

Game G_3: This is identical to G_2, except that for each encryption query, the encryption algorithm now samples a random, independent key $\mathtt{k}_2^*$ to compute the MAC, instead of using the legitimate key $\mathtt{k}_2$.

Claim 3.3. $|\Pr[G_3] - \Pr[G_2]| \leq q_d \cdot \mathsf{Adv}^{euf\text{-}qcma}_{\mathcal{A},\mathcal{MA}}(\lambda)$.

Proof. Let q be the total number of encryption and challenge queries. We construct a QPT adversary $\mathcal{C}$ as in the previous Claim 3.2.

Let Forge_3 be the event that $\mathcal{A}$ puts a non-negligible weight to a pair $(\widetilde{c}\|\widetilde{\tau})$ in its decryption queries such that $(\widetilde{c}\|\widetilde{\tau}) \notin D$ and $\mathsf{SymDec}'_{\mathtt{k}_1\|\mathtt{k}_2}(\widetilde{c}\|\widetilde{\tau}) \neq \perp$ (using the legitimate key). $\mathcal{A}$ could distinguish the two games if and only if $\Pr[\mathsf{Forge}_3]$ is non-negligible.

We first prove for $q = 1$. Notice that in game G_3, $\mathcal{C}$ makes no queries to the MAC oracle. By randomly measuring a decryption query from $\mathcal{A}$, with probability $\Pr[\mathsf{Forge}_3]/q_d$, it gets a *valid* pair $(\widetilde{c}\|\widetilde{\tau}) \notin D$. $\mathcal{C}$ then outputs this pair which is a forge with non-negligible probability if Forge_3 happens with non-negligible probability. By the security of $\mathcal{MA}$, the proof for $q = 1$ follows.

We then use the standard hybrid argument to prove for a general case $q = \mathsf{poly}(\lambda)$: all the queries to the compressed oracles except the last one are answered by querying to the MAC oracle (as in G_2), and the last one is answered with the modified encryption oracle as in G_3. $\mathcal{A}$ could trivially distinguish the two

[6] Recall that the compressed oracle only computes the function f if the adversary's response registers is not $|0\rangle$ in the Fourier basis, which is equivalent to $|+\rangle$ in the computational basis (see Figure 1).

games, if there is a ciphertext $(c_i \| \tau_i) \in D$ such that $i < q$ and $c_i = c_q$. However, analogously to the argument in the previous Claim 3.2, this happens with negligible probability. Thus, $\mathcal{C}$ can measure its database to get $q-1$ distinct, valid pairs, alongside a pair $(\widetilde{c}, \widetilde{\tau})$ it gets from measuring $\mathcal{A}$'s decryption queries. With probability $\Pr[\mathsf{Forge}_3]/q_d$, $\mathcal{C}$ output q distinct, valid pairs while making only $q-1$ queries to the MAC oracle. The claim then follows from the security of $\mathcal{MA}$.

Game G_4: This is identical to G_3, except that we change the implementation of the compressed oracles from $\mathsf{CStO}_f^{|r\rangle}[x \mapsto c\|\tau]$ to $\mathsf{CStO}_f^{|\mathtt{k}_2^*\rangle}[x \mapsto c\|\tau]$, where $\mathtt{k}_2^*$ is a random key sampled for each query. Since the two oracle variations are equivalent, this change does not affect the adversary's success probability. We thus have $\Pr[G_4] = \Pr[G_3]$.

Furthermore, the advantage of $\mathcal{A}$ in this final game can be reduced to its advantage against $\mathcal{SE}$.

Claim 3.4. $\Pr[G_4] \leq \mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{qind\text{-}qcpa}(\lambda)$.

Proof. To see that, we construct a QPT adversary $\mathcal{B}$ from $\mathcal{A}$ as follows: $\mathcal{B}$ runs $\mathcal{A}$ as its subroutine. For each encryption or challenge query, $\mathcal{B}$ implements the compressed oracle which is a purification over the random key $\mathtt{k}_2^*$ of the MAC. It first sends the plaintext registers to its challenger and receives back a ciphertext, it then tags the received ciphertext with the MAC and forwards them to $\mathcal{A}$. A quantum circuit of this unitary is given in Figure 5.

Notice in this game, $\mathcal{B}$ can always answer decryption queries, without needing to query to $\mathcal{SE}$ decryption oracle, by using its own database. The advantage of $\mathcal{B}$ against $\mathcal{SE}$ is exactly the advantage of $\mathcal{A}$ in this game, thus proving the claim.

Putting everything together, we finish the proof of the theorem. □

5 Quantum-Secure Public-key Encryption

5.1 Definition of Security

In this section, we consider the quantum security of public-key cryptosystems, namely indistinguishability and non-malleability. We start by giving the definition for these notions in the quantum world, and then discuss their relative strengths, following classical works [BDPR98].

Indistinguishability Security. The indistinguishability notions can be defined analogously to the ones given in Section 4. In what follows, let Π be a family of random permutations over $\mathcal{X}$, we define a real-or-random oracle allowing quantum queries as follows.

$$\mathcal{RR}(b) = \begin{cases} \mathsf{Enc}_{\mathsf{pk}} & \text{if } b = 1, \\ \mathsf{CStO}_{\mathsf{Enc}_{\mathsf{pk}} \circ \pi}^{|r\rangle} & \text{if } b = 0, \text{where } \pi \xleftarrow{\$} \Pi. \end{cases}$$

Similarly, we overload the notation of $\mathsf{CInvO}_{\mathsf{Dec}_{\mathsf{k}}}$ to denote both the standard oracle $\mathcal{O}_{\mathsf{Dec}_{\mathsf{k}}}$ in the real-world (when there is no database D) and the actual $\mathsf{CInvO}_{\mathsf{Dec}_{\mathsf{k}}}$ oracle in the random-world.

Definition 3 $(\mathsf{qIND\text{-}qCPA}, \mathsf{qIND\text{-}qCCA1}, \mathsf{qIND\text{-}qCCA2})$**.**
Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a quantum adversary. For* $qatk \in [qcpa, qcca1, qcca2]$*, we define the following game, where the oracles* $\mathcal{O}_1, \mathcal{O}_2$ *are defined according to* $qatk$*:*

```
Experiment $\mathsf{Expt}_{\mathcal{E}}^{qind\text{-}qatk-b}(\lambda, \mathcal{A})$:
1 : $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
2 : $|\Phi\rangle \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(\mathtt{pk})$
3 : $b' \leftarrow \mathcal{A}_2^{\mathcal{RR}(b), \mathcal{O}_2}(|\Phi\rangle)$
4 : return $b'$
```

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{Dec}_{\mathsf{sk}}}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{Dec}_{\mathsf{sk}}}$	$\mathsf{CInvO}_{\mathsf{Dec}_{\mathsf{sk}}}$

We define $\mathcal{A}$*'s advantage by*

$$\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qind\text{-}qatk}(\lambda) := \left| \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qind\text{-}qatk-1}(\lambda, \mathcal{A}) = 1 \right] - \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qind\text{-}qatk-0}(\lambda, \mathcal{A}) = 1 \right] \right|.$$

We say $\mathcal{E}$ *is secure in the sense of* qIND-$\textsc{qATK}$ *if* $\mathcal{A}$ *being* QPT *implies that* $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qind\text{-}qatk}(\lambda)$ *is negligible.*

Similarly as in Section 4, our definitions, restricted to classical challenge queries, are equivalent to Boneh-Zhandry's notions (IND-$\textsc{qATK}$). Furthermore, the following theorem shows that our notions are closed under composition.

Theorem 4. *An encryption scheme* $\mathcal{E}$ *is many-message* qIND-$\textsc{qATK}$ *secure iff it is single-message* qIND-$\textsc{qATK}$ *secure.*

The proof is similar to that of Theorem 1; we give it in Appendix D.1.

Non-Malleability Security. Intuitively, the classical definitions [BDPR98,BS06] involve having an adversary play a challenge-response game. In the challenge phase, the adversary is given an encryption y of a message x it produced itself. It must then output a vector of ciphertexts $\mathbf{y}$ (whose components can be y - in this case, the decryption returns $\perp$) called *adversarial ciphertexts*, together with an arbitrary string. The security definitions require that the distribution of the adversary's output and *the decryptions of the adversarial ciphertexts* is indistinguishable from the distribution when the adversary receives an encryption of some random message $\widetilde{x}$ instead of x. The non-malleability property can be established by saying that when an encryption of x given to the adversary is replaced with an encryption of a random $\widetilde{x}$, even the *contents* of encryption messages that the adversary sends cannot change in any computationally noticeable way.

A closer look at the adversarial ciphertexts distribution gives us different classical definitions, which leads to different composability properties. As pointed out by Pass, shelat and Vaikuntanathan [PsV07], indistinguishability-based definitions of encryption may or may not compose in the context of non-malleability, depending on how we treat an "invalid adversary" that outputs *invalid* ciphertexts as part of its adversarial output. In the quantum setting, the adversary can output a superposition of adversarial ciphertexts, which might include invalid ciphertexts, even if it is "hard" to generate invalid ciphertexts. This leaves us no choice but to incorporate invalid adversaries into the definitions. The definitions given here are syntactically close to the classical definitions of [BS06, Definition 4.1].

Definition 4 ($\mathsf{qNME\text{-}qCPA}, \mathsf{qNME\text{-}qCCA1}, \mathsf{qNME\text{-}qCCA2}$)**.**
Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an public-key encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$ and $r \in \mathbb{N}$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

```
Experiment $\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-b}(\lambda, \mathcal{A})$:
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
    $|\Psi_1\rangle \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(\mathsf{pk})$
    $|\Psi_2\rangle \coloneqq \sum_{\mathbf{y},\mathbf{z}} \alpha_{\mathbf{y},\mathbf{z}} |\mathbf{y}, \mathbf{z}\rangle |\phi_{\mathbf{y},\mathbf{z}}\rangle \leftarrow \mathcal{A}_2^{\mathcal{RR}(b),\mathcal{O}_2}(|\Psi_1\rangle)$
      where $|\mathbf{y}| = |\mathbf{z}| = r$
    $|\Psi_3\rangle \leftarrow \mathsf{CInvO}_{\mathsf{Dec}_{\mathsf{sk}}} |\Psi_2\rangle$
    $b' \leftarrow \mathcal{A}_3^{\mathcal{O}_2}(|\Psi_3\rangle)$
    return $b'$
```

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{Dec}_{\mathsf{sk}}}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{Dec}_{\mathsf{sk}}}$	$\mathsf{CInvO}_{\mathsf{Dec}_{\mathsf{sk}}}$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qnme\text{-}qatk}(\lambda) \coloneqq \left| \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-0}(\lambda, \mathcal{A}) = 1\right]\right| .$$

We say $\mathcal{E}$ is secure in the sense of qNME-qATK *if $\mathcal{A}$ being* QPT *implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qnme\text{-}qatk}(\lambda)$ is negligible.*

The following theorem shows that our notions are closed under composition.

Theorem 5. *An encryption scheme $\mathcal{E}$ is many-message* qNME-qATK *secure iff it is single-message* qNME-qATK *secure.*

The proof is similar to that of Theorem 4; we omit the details.

Relating Indistinguishability and Non-Malleability. A full characterization of fully-quantum indistinguishability and non-malleability notions is summarized in Figure 2. These results are identical as in the classical setting [BDPR98]. We use slightly modified constructions of [BDPR98] in the proofs: the attacks carry in the classical manner, only the security proofs need to be adapted.

5.2 A Separation Example

Here we show a separation example of our setting from the classical challenge queries setting of [BZ13b]. The idea is to install a backdoor that only a quantum adversary can use, by doing some quantum computation. We need to ensure that the backdoor is useless even if the adversary has quantum access to the

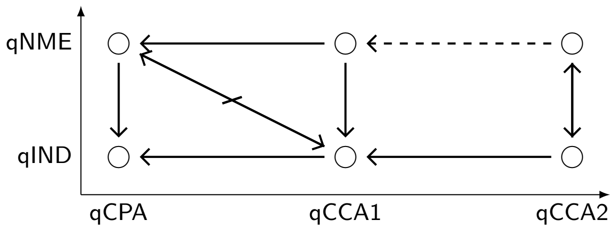

Fig. 2. An arrow is an implication. There is a path from **A** to **B** if and only if **A** implies **B**. The hatched arrows and the dashed arrow represent non-trivial separations we actually prove in Appendix D.2.

decryption oracle in the learning phases. Our construction follows the hybrid encryption paradigm combining a CCA2-secure public-key encryption and a one-time CCA2-secure symmetric encryption [CS03]. The attack is similar in spirit to that for symmetric encryption.

Theorem 6. *If there exists an encryption scheme $\mathcal{E}$ which is* IND-qCCA2 *secure against* QPT *adversaries, then there exists an encryption scheme $\mathcal{E}'$ which is also* IND-qCCA2 *secure, but* qIND-qCPA *insecure.*

Proof Sketch. We construct the new encryption scheme $\mathcal{E}'$ by encrypting each plaintext with a one-time symmetric encryption which is secure against *classical* adaptive chosen-ciphertext attacks and encrypting the key using an IND-qCCA2 secure public-key encryption scheme $\mathcal{E}$.

Each challenge query is encrypted under a random, independent symmetric key. If all challenge queries are classical, the security of the scheme can be easily reduced to the security of the one-time symmetric encryption. Here we only need the symmetric encryption to be secure against classical chosen-ciphertext attacks, because the adversary cannot make quantum queries to the symmetric encryption oracle.

A one-time symmetric encryption that achieves security against adaptive chosen-ciphertext attack can be built from one-time encryption whose ciphertext is attached with its one-time MAC. However, the adversary can use the attack described in Section 4.2, if the symmetric encryption is instantiated in the stream cipher-like style. We give a concrete construction and its proofs in Appendix D.3.

5.3 A Lifting Theorem: From IND-qCCA2 to qIND-qCCA2

We present a compiler transforming IND-qATK security to qIND-qATK security. Our compiler follows the classical hybrid encryption paradigm. The message is encrypted under a random symmetric key each time, and the key is encrypted by the public-key encryption scheme. Since the same randomness is used for each query in superposition, we can use the same random symmetric key in superposition each time. This means that the adversary never has quantum access to the encryption algorithm of the public-key scheme, only the symmetric encryption needs to be secure against quantum queries, which we know how to construct from one-way functions.

Construction 3. *Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme which is* IND-qATK *secure and δ-correct. Let $\mathcal{SE} = (\mathsf{SymEnc}, \mathsf{SymDec})$ be a one-time secure symmetric-key encryption scheme (as defined in Appendix C.1). If* qATK $=$ qCCA2*, $\mathcal{SE}$ needs to be* OT-qCCA2 *secure. Otherwise, $\mathcal{SE}$ is* OT *secure. We construct a new public-key encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows.*

```
$\mathsf{KeyGen}'(\lambda)$ :
  $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
  return $(\mathtt{pk}, \mathtt{sk})$
```

```
$\mathsf{Enc}'_{\mathtt{pk}}(x)$ :
  $\mathtt{k} \xleftarrow{\$} \mathcal{K}()$
  $c_1 \leftarrow \mathsf{Enc}_{\mathtt{pk}}(\mathtt{k})$
  $c_2 \leftarrow \mathsf{SymEnc}_{\mathtt{k}}(x)$
  return $c_1 \| c_2$
```

```
$\mathsf{Dec}'_{\mathtt{sk}}(c_1 \| c_2)$ :
  $\mathtt{k} \leftarrow \mathsf{Dec}_{\mathtt{sk}}(c_1)$
  $x \leftarrow \mathsf{SymDec}_{\mathtt{k}}(c_2)$
  return $x$
```

Remark 2. In this construction, we make no extra assumptions. We know that the existence of IND-qATK secure encryption implies the existence of quantum-secure one-way functions, which in turn implies the existence of quantum-secure pseudorandom permutations [Zha16]. In Appendix C.2, we give concrete constructions achieving one-time security notions defined in Appendix C.1 from quantum-secure pseudorandom permutations. IND-qATK secure public-key encryption can be constructed based on quantum-resistant assumptions (e.g., the Learning With Errors problem) [BZ13b].

We give the security proof for adaptive chosen-ciphertext security in Appendix D.5, the other cases can be treated similarly.

Theorem 7. *The encryption scheme $\mathcal{E}'$ defined in Construction 3 is* qIND-qCCA2 *secure. In particular, for any* QPT *adversary $\mathcal{A}$, there exist* QPT *adversaries $\mathcal{B}, \mathcal{C}$ such that*

$$\mathsf{Adv}^{qind\text{-}qcca2}_{\mathcal{A},\mathcal{E}'}(\lambda) \leq q \cdot \left(O(q_d \cdot \delta) + \mathsf{Adv}^{ind\text{-}qcca2}_{\mathcal{B},\mathcal{E}}(\lambda) + \mathsf{Adv}^{ot\text{-}qcca2}_{\mathcal{C},\mathcal{SE}}(\lambda) \right),$$

where q is the number of oracle calls to the real-or-random oracle and q_d is the number of decryption queries in the second phase.

Remark 3. The q factor comes from the fact that in our IND-qCCA2 definition, the adversary can only make a single challenge query. If we consider the same many-challenge setting, there would be no factor q.

References

AAB+19. Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

AGM18. Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 489–519. Springer, Heidelberg, April / May 2018.

ATTU16. Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 44–63. Springer, Heidelberg, 2016.

BDF$^+$11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

BDJR97. Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, October 1997.

BDPR98. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Heidelberg, August 1998.

BJ15. Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 609–629. Springer, Heidelberg, August 2015.

BN08. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008.

BS06. Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. Cryptology ePrint Archive, Report 2006/228, 2006. http://eprint.iacr.org/2006/228.

BZ13a. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013.

BZ13b. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013.

CETU20. Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Tabia, and Dominique Unruh. On quantum indistinguishability under chosen plaintext attacks. *Unpublished manuscript*, 2020.

CMSZ19. Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428, 2019. https://eprint.iacr.org/2019/428.

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

DDN00. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

DFNS14. Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. In Carles Padró, editor, *ICITS 13*, volume 8317 of *LNCS*, pages 142–161. Springer, Heidelberg, 2014.

FP96. Christopher A Fuchs and Asher Peres. Quantum-state disturbance versus information gain: Uncertainty relations for quantum information. *Physical Review A*, 53(4):2038, 1996.

GHS16. Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 60–89. Springer, Heidelberg, August 2016.

GKS20. Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. Make quantum indistinguishability great again. Cryptology ePrint Archive, Report 2020/266, 2020. https://eprint.iacr.org/2020/266.

Gol04. Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996.

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017.

KKVB02. Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5):050304, 2002.

KLLN16. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, Heidelberg, August 2016.

MS16. Shahram Mossayebi and Rüdiger Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *arXiv preprint arXiv:1609.03780*, 2016.

NC11. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

PsV07. Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Relations among notions of non-malleability for encryption. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 519–535. Springer, Heidelberg, December 2007.

PW08. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

Sho99. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

Sim94. Daniel R. Simon. On the power of quantum computation. In *35th FOCS*, pages 116–123. IEEE Computer Society Press, November 1994.

SJS16. Vladimir Soukharev, David Jao, and Srinath Seshadri. Post-quantum security models for authenticated encryption. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 64–78. Springer, Heidelberg, 2016.

Unr15. Dominique Unruh. Revocable quantum timed-release encryption. *Journal of the ACM (JACM)*, 62(6):49, 2015.

WZ82. William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

Zha12. Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012.

Zha16. Mark Zhandry. A note on quantum-secure PRPs. Cryptology ePrint Archive, Report 2016/1076, 2016. http://eprint.iacr.org/2016/1076.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019.

Supplementary Material

A Security Definitions

A.1 Pseudorandom Functions

Definition 5. *A quantum-secure pseudorandom function (*qPRF*) is a family of efficient classical functions* $\mathsf{PRF} : \{0,1\}^\lambda \times \{0,1\}^m \to \{0,1\}^n$ *such that the following holds. For any polynomially bounded* $m = m(\lambda)$ *and* $n = n(\lambda)$*, and any* QPT *adversary* $\mathcal{A}$*,* $\mathcal{A}$ *cannot distinguish* $\mathsf{PRF}(k, \cdot)$ *for a random* $k \xleftarrow{\$} \{0,1\}^\lambda$ *from a truly random function* $H : \{0,1\}^m \to \{0,1\}^n$*. That is, there exists a negligible* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{PRF}(k,\cdot)}(\lambda) = 1 \mid k \xleftarrow{\$} \{0,1\}^\lambda\right] - \Pr\left[\mathcal{A}^{H(\cdot)}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)\,.$$

A.2 Pseudorandom Permutations

Definition 6. *A (strongly) quantum-secure pseudorandom permutation (*qPRP*) is a family of efficient classical function pairs* $\mathsf{PRP} : \{0,1\}^\lambda \times \{0,1\}^m \to \{0,1\}^m$ *and* $\mathsf{PRP}^{-1} : \{0,1\}^\lambda \times \{0,1\}^m \to \{0,1\}^m$ *such that the following holds. First, for every key* k *and* $m \in \mathbb{N}$*, the functions* PRP *and* PRP^{-1} *are inverses of each other. That is,* $\mathsf{PRP}^{-1}(\mathsf{PRP}(k, x)) = x$ *for any* k, x, m*. This implies that* PRP *is a permutation.*

Second, for any polynomially bounded $m = m(\lambda)$*, and any* QPT *adversary* $\mathcal{A}$*,* $\mathcal{A}$ *cannot distinguish* $\mathsf{PRP}(k, \cdot)$ *for a random* $k \xleftarrow{\$} \{0,1\}^\lambda$ *from a truly random permutation* $P : \{0,1\}^m \to \{0,1\}^m$*. That is, there exists a negligible* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{PRP}(k,\cdot),\mathsf{PRP}^{-1}(k,\cdot)}(\lambda) = 1 \mid k \xleftarrow{\$} \{0,1\}^\lambda\right] - \Pr\left[\mathcal{A}^{P(\cdot),P^{-1}(\cdot)}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)\,.$$

A.3 Symmetric-key Encryption

A symmetric-key cryptosystem $\mathcal{SE} = (\mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec})$ consists of three PPT algorithms.

- $\mathcal{K}()$ is a probabilistic key generation algorithm which takes no input and outputs a secret key $\mathtt{k}$.
- $\mathsf{SymEnc}_{\mathtt{k}}(x; r)$ is a probabilistic encryption algorithm which takes as input a secret key $\mathtt{k}$, a plaintext $x \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), samples a random coin on each invocation $r \in \mathcal{R}$ (where $\mathcal{R}$ is the randomness space), and outputs a ciphertext y. We sometimes omit the random coin and write $\mathsf{SymEnc}_{\mathtt{k}}(x)$.
- $\mathsf{SymDec}_{\mathtt{k}}(y)$ is a deterministic decryption algorithm which takes as input a secret key $\mathtt{k}$ and a ciphertext y, and outputs a message $x \in \mathcal{X} \cup \{\perp\}$, where $\perp$ is a distinguished symbol indicating decryption failure.

Security Definitions. In the following, we let the string *qatk* be instantiated by any of the formal symbols $qcpa, qcca1, qcca2$, while qATK is the corresponding formal symbol from $\mathsf{qCPA}, \mathsf{qCCA1}, \mathsf{qCCA2}$. When we say $\mathcal{O}_i = \varnothing$ where $i \in \{1, 2\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns $\bot$.

Definition 7 ($\mathsf{IND\text{-}qCPA}, \mathsf{IND\text{-}qCCA1}, \mathsf{IND\text{-}qCCA2}$ [BZ13b])**.**
Let $\mathcal{SE} = (\mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec})$ be a symmetric encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to qatk:

```
Experiment $\mathsf{Expt}_{\mathcal{SE}}^{ind\text{-}qatk-b}(\lambda, \mathcal{A})$:
    $\mathtt{k} \xleftarrow{\$} \mathcal{K}()$
    $|x_0, x_1\rangle\,|\phi\rangle \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{SymEnc}_\mathtt{k}}, \mathcal{O}_1}(\lambda)$
    **if** $|x_0| \neq |x_1|$ **then return** $0$
    $y^* \leftarrow \mathcal{O}_{\mathsf{SymEnc}_\mathtt{k}}(x_b)$
    $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{SymEnc}_\mathtt{k}}, \mathcal{O}_2}(|y^*\rangle\,|\phi\rangle)$
    **return** $b'$
```

qatk	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
qcpa	$\varnothing$	$\varnothing$
qcca1	$\mathsf{SymDec}_\mathtt{k}(\cdot)$	$\varnothing$
qcca2	$\mathsf{SymDec}_\mathtt{k}(\cdot)$	$\mathsf{SymDec}_\mathtt{k}(\cdot)$ with $\cdot \neq y^*$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk}(\lambda) := \left|\Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{ind\text{-}qatk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{ind\text{-}qatk-0}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{SE}$ is secure in the sense of IND-qATK *if $\mathcal{A}$ being* QPT *implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{ind\text{-}qatk}(\lambda)$ is negligible.*

A.4 Public-key Encryption

A public-key cryptosystem $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms.

- $\mathsf{KeyGen}(\lambda)$ is a probabilistic key generation algorithm which takes as input the security parameter λ and outputs a pair $(\mathtt{pk}, \mathtt{sk})$ of matching public and secret keys.
- $\mathsf{Enc}_\mathtt{pk}(x; r)$ is a probabilistic encryption algorithm which takes as input a public key $\mathtt{pk}$, a plaintext $x \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), samples a random coin on each invocation $r \in \mathcal{R}$ (where $\mathcal{R}$ is the randomness space), and outputs a ciphertext y. We sometimes omit the random coin and write $\mathsf{Enc}_\mathtt{pk}(x)$.
- $\mathsf{Dec}_\mathtt{sk}(y)$ is a deterministic decryption algorithm which takes as input a secret key $\mathtt{sk}$ and a ciphertext y, and outputs a message $x \in \mathcal{X} \cup \{\bot\}$, where $\bot$ is a distinguished symbol indicating decryption failure.

Security Definitions. For any subset D of the ciphertext space $\mathcal{C}$, we define the "punctured" decryption oracle $\widetilde{\mathsf{Dec}}^{D}_{\mathtt{sk}}(y)$ which returns $\mathsf{Dec}_{\mathtt{sk}}(y)$ if $y \notin D$, else it returns $\perp$.

Definition 8 (IND-qCPA, IND-qCCA1, IND-qCCA2 [BZ13b])**.** *Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an public-key encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For qatk $\in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to qatk:*

```
Experiment $\mathsf{Expt}^{ind\text{-}qatk-b}_{\mathcal{E}}(\lambda, \mathcal{A})$:
$(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
$|x_0, x_1\rangle\,|\phi\rangle \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(\mathtt{pk})$
**if** $|x_0| \neq |x_1|$ **then return** $0$
$y^* \leftarrow \mathsf{Enc}_{\mathtt{pk}}(x_b)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(|y^*\rangle\,|\phi\rangle)$
**return** $b'$
```

qatk	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
qcpa	$\varnothing$	$\varnothing$
qcca1	$\mathsf{Dec}_{\mathtt{sk}}(\cdot)$	$\varnothing$
qcca2	$\mathsf{Dec}_{\mathtt{sk}}(\cdot)$	$\widetilde{\mathsf{Dec}}^{D}_{\mathtt{sk}}(\cdot)$ with $D = \{y^*\}$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{E}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{ind\text{-}qatk-1}_{\mathcal{E}}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}^{ind\text{-}qatk-0}_{\mathcal{E}}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{E}$ is secure in the sense of IND-qATK *if $\mathcal{A}$ being* QPT *implies that $\mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{E}}(\lambda)$ is negligible.*

A.5 Quantum-secure MAC

A message authentication code $\mathcal{MA} = (\mathcal{K}, \mathsf{MAC}, \mathsf{Ver})$ consists of three PPT algorithms.

- $\mathcal{K}()$ is a probabilistic key generation algorithm which takes no input and outputs a secret key $\mathtt{k}$.
- $\mathsf{MAC}_{\mathtt{k}}(x)$ is a deterministic tagging algorithm which takes as input a key $\mathtt{k} \in \mathcal{K}$, a message $x \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), and outputs a tag τ.
- $\mathsf{Ver}_{\mathtt{k}}(x, \tau)$ is a deterministic decryption algorithm which takes as input a key $\mathtt{k}$, a message x, and a candidate tag τ for x, and outputs a bit b.

We require that $\mathsf{Ver}_{\mathtt{k}}(x, \tau) = 1$ if and only if $\tau = \mathsf{MAC}_{\mathtt{k}}(x)$, for all $x \in \mathcal{X}$ and for all $\mathtt{k} \leftarrow \mathcal{K}()$. We also require that for all $\mathtt{k} \leftarrow \mathcal{K}()$ and for all τ, we have $\mathsf{Ver}_{\mathtt{k}}(x, \tau) = 0$ if $x = \perp$.

Security Definitions. Boneh and Zhandry define unforgeability (against quantum queries) for classical MACs as follows [BZ13a]. They also show that random functions satisfy this notion.

Definition 9 (EUF-qCMA)**.** *Let $\mathcal{MA} = (\mathcal{K}, \mathsf{MAC}, \mathsf{Ver})$ be a MAC with message set $\mathcal{X}$. Consider the following experiment with a* QPT *adversary $\mathcal{A}$:*

1. Generate a random key $\mathtt{k} \xleftarrow{\$} \mathcal{K}$.
2. $\mathcal{A}$ receives quantum oracle for $\mathsf{MAC}_{\mathtt{k}}$, makes q queries and outputs a string s.
3. $\mathcal{A}$ wins if s contains $q+1$ distinct valid classical message/tag pairs.

*We say that $\mathcal{MA}$ is existentially unforgeable under quantum chosen message attacks (*EUF-qCMA*) if no* QPT *adversary can succeed at the above experiment with better than negligible probability in λ.*

B Additional Details on Quantum Oracles

B.1 Proof of Lemma 2

We do detailed calculation of equalities claimed in the proof of Lemma 2. A similar calculation is presented in [Zha19,CMSZ19].

Let X, Y denote the adversary's registers, and R, F denote the oracle's registers.

We start with the standard oracle StO_f, which acts on the computational basis. The initial oracle state of StO_f is $\frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, |f(0;r)\| \cdots \| f(2^m-1;r)\rangle$. We write $|r\rangle\, |f(0;r)\| \ldots \| f(2^m-1;r)\rangle$ as $|r\rangle\, |D_r\rangle$. D_r can be seen as the truth table of f, where $D_r(x) := f(x;r)$. StO_f is a unitary defined (on the adversary's basis states) as:

$$\begin{aligned}\mathsf{StO}_f\, |x,y\rangle \otimes \frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, |D_r\rangle \mapsto & \frac{1}{\sqrt{2^\ell}} \sum_r |x, y \oplus D_r(x)\rangle\, |r\rangle\, |D_r\rangle \\ = & \frac{1}{\sqrt{2^\ell}} \sum_r |x, y \oplus f(x;r)\rangle\, |r\rangle\, |D_r\rangle\end{aligned}$$

Next, we consider the well-known phase oracle PhsO_f. It is obtained from StO_f by applying QFT operations to the adversary's response registers before and after the query, that is $\mathsf{PhsO}_f := \mathsf{QFT}^{\dagger Y} \circ \mathsf{StO}_f \circ \mathsf{QFT}^Y$. The initial oracle state of PhsO_f is also $\frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, |D_r\rangle$. On the adversary's basis state, it performs the map:

$$\mathsf{PhsO}_f\, |x,z\rangle \otimes \frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, |D_r\rangle \mapsto \frac{1}{\sqrt{2^\ell}} \sum_r (-1)^{z \cdot f(x;r)}\, |x,z\rangle\, |r\rangle\, |D_r\rangle\,.$$

The Fourier oracle is obtained from PhsO_f by applying QFT operations to the oracle registers before and after the query, that is $\mathsf{FourierO}_f := \mathsf{QFT}^{\dagger F} \circ \mathsf{PhsO}_f \circ \mathsf{QFT}^F$. The initial state of $\mathsf{FourierO}_f$ is:

$$\frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, \mathsf{QFT}^F\, |D_r\rangle = \frac{1}{\sqrt{2^\ell}} \sum_r |r\rangle\, \frac{1}{\sqrt{2^{n2^m}}} \sum_E (-1)^{E \cdot D_r}\, |E\rangle\,.$$

We do the computation on the basis state as:

$$\begin{aligned}
|x,z\rangle \otimes |r\rangle |E\rangle &\xrightarrow{\mathsf{QFT}^F} \frac{1}{\sqrt{2^{n2^m}}} \sum_F (-1)^{E\cdot F} |x,z\rangle |r\rangle |F\rangle \\
&\xrightarrow{\mathsf{PhsO}_f} \frac{1}{\sqrt{2^{n2^m}}} \sum_F (-1)^{E\cdot F + z\cdot F(x)} |x,z\rangle |r\rangle |F\rangle \\
&\overset{(*)}{=} \frac{1}{\sqrt{2^{n2^m}}} \sum_F (-1)^{(E\oplus P_{x,z})\cdot F} |x,z\rangle |r\rangle |F\rangle \\
&\xrightarrow{\mathsf{QFT}^{\dagger F}} |x,z\rangle |r\rangle |E\oplus P_{x,z}\rangle ,
\end{aligned}$$

where we write $F(x)$ to denote the x-th row of the truth table F, and $P_{x,z}$ is the point function that outputs z on x and 0 everywhere else. $E \oplus P_{x,z}$ is the function $(E \oplus P_{x,z})(x') \coloneqq E(x') \oplus P_{x,z}(x')$. The equality $(*)$ holds because $z \cdot F(x) = (0\|\cdots\|z\|\cdots\|0) \cdot F = P_{x,z} \cdot F$.

From this computation, the unitary $\mathsf{FourierO}_f$ can be defined (on the adversary's basis state) as:

$$\begin{aligned}
&\mathsf{FourierO}_f\, |x,z\rangle \otimes \frac{1}{\sqrt{2^\ell}} \frac{1}{\sqrt{2^{n2^m}}} \sum_r |r\rangle \sum_E (-1)^{E\cdot D_r} |E\rangle \\
&\mapsto \frac{1}{\sqrt{2^\ell}} \frac{1}{\sqrt{2^{n2^m}}} \sum_r \sum_E (-1)^{E\cdot D_r} |x,z\rangle |r\rangle |E\oplus P_{x,z}\rangle .
\end{aligned}$$

Overall, we have shown that $\mathsf{QFT}^{\dagger YF} \circ \mathsf{StO}_f \circ \mathsf{QFT}^{YF} = \mathsf{FourierO}_f$. The other equality follows directly.

B.2 Proof of Lemma 3

We consider the oracle operation handling a single query. In this setting, the database D will be a single triple (r, x, y) where $y = f(x; r)$. In the following, we denote X, Y the adversary's registers, and D^X, D^R, D^Y to denote the oracle's registers.

With this notation, we describe some local procedures acting on the database. First, let Update_x be the following unitary operation:

$$\begin{aligned}
\mathsf{Update}_x\,(|r\rangle |x,z\rangle) &= |r\rangle |x,z\rangle \quad \text{for } \ z \neq 0, \\
\mathsf{Update}_x\,(|r\rangle |x,0\rangle) &= |r\rangle |\bot,0\rangle , \\
\mathsf{Update}_x\,(|r\rangle |\bot,0\rangle) &= |r\rangle |x,0\rangle .
\end{aligned}$$

This operations is actually an involution, as $\mathsf{Update}_x \circ \mathsf{Update}_x = \mathbb{1}$.

Let U_f be the unitary implementing f, that is U_f acts on three registers: $U_f\,|r,x,u\rangle = |r,x,u \oplus f(x;r)\rangle$. Let Decomp_x be the following unitary operation:

$$\mathsf{Decomp}_x \coloneqq \mathsf{QFT}^{D^Y} \circ U_f \circ \mathsf{Update}_x .$$

Decomp is then defined as the related unitary acting on the quantum system over x, z, D states in superposition as follows.

$$\mathsf{Decomp}\,|x,z\rangle \otimes |D\rangle = |x,z\rangle \otimes \mathsf{Decomp}_x\,|D\rangle\,.$$

Let $\mathsf{Increase}$ be the procedure that initializes a new register $|0,\perp,0\rangle$. That is, $\mathsf{Increase}\,|x,z\rangle = |x,z\rangle \otimes |(0,\perp,0)\rangle$. Let $\mathsf{FourierO}'$ be unitary defined on the adversary's basis states as:

$$\begin{aligned}\mathsf{FourierO}'\,|x,z\rangle \otimes |D\rangle =&\ \mathsf{FourierO}'\,|x,z\rangle \otimes \frac{1}{\sqrt{2^\ell}}\frac{1}{\sqrt{2^n}}\sum_r\sum_u(-1)^{u\cdot f(x;r)}\,|r,x,u\rangle \\ =&\ |x,z\rangle \otimes \frac{1}{\sqrt{2^\ell}}\frac{1}{\sqrt{2^n}}\sum_r\sum_u(-1)^{u\cdot f(x;r)}\,|r,x,u\oplus z\rangle\,.\end{aligned}$$

Finally, we define the $\mathsf{CFourierO}'_f$ oracle:

$$\mathsf{CFourierO}'_f \coloneqq \mathsf{Decomp}^\dagger \circ \mathsf{FourierO}' \circ \mathsf{Decomp} \circ \mathsf{QFT}^{D^R} \circ \mathsf{Increase}.$$

We now prove the equivalence between $\mathsf{CFourierO}'_f$ and $\mathsf{FourierO}_f$.

Proposition 1. *Let $\mathcal{A}$ be any 1-query quantum oracle algorithm. Then,*

$$\Pr\left[\mathcal{A}^{\mathsf{FourierO}_f}() = 1\right] = \Pr\left[\mathcal{A}^{\mathsf{CFourierO}'_f}() = 1\right].$$

Proof. We prove through a sequence of games, which works almost the same as the proof of [Zha19, Lemma 4]. In what follows, we ambiguously denote $\mathsf{QFT}\,|f(x;r)\rangle$ by $|\eta_x\rangle$ for each $x \in \{0,1\}^m$.

Game G_0: In this game, the adversary interacts with the Fourier oracle $\mathsf{FourierO}_f$, whose initial state is $\frac{1}{\sqrt{2^\ell}}\frac{1}{\sqrt{2^{n2^m}}}\sum_r |r\rangle_{D^R}\,|\eta_0\|\cdots\|\eta_{2^m-1}\rangle_{D^Y}$.

Game G_1: In this game, we represent the oracle as a complete database $|D\rangle = \sum_r |r\rangle_{D^R}\,|(0,\eta_0)\|\cdots\|(2^m-1,\eta_{2^m-1})\rangle_{D^Y}$. The update procedure for a query is then simply $\mathsf{FourierO}'$. G_1 is identical to G_0, since we have inserted the input points $0,\ldots,2^m-1$ into the oracle's state, which is independent from the adversary's state.

Game G_2: In this game, the oracle starts out as the empty database:

$$\sum_r |r\rangle_{D^R}\,|(\perp,0)\|\cdots\|(\perp,0)\rangle_{D^Y}\,,$$

which can be prepared by applying QFT to the all-zero register D^R. Then a query is implemented as $\mathsf{Decomp}'^\dagger \circ \mathsf{FourierO}' \circ \mathsf{Decomp}'$, where Decomp' is a procedure applying Decomp_x for all x in the message space, once at a time from 0 to 2^m-1. At the beginning, Decomp' is applied to the empty database, which maps it to the complete database $\sum_r |r\rangle\,|(0,\eta_0)\|\cdots\|(2^m-1,\eta_{2^m-1})\rangle$. Note that Decomp' only affects the oracle's registers and thus commutes with any computation on the adversary's side. Therefore, G_2 is perfectly indistinguishable from G_1.

Game G_3: In this final game, we use the compressed oracle $\mathsf{CFourierO}'_f$. We note that $\mathsf{FourierO}'$ and $\mathsf{Decomp}_{x'}$ commute for any query containing $x \neq x'$. Thus, for any $x' \neq x$, we can move the computation of $\mathsf{Decomp}_{x'}$ to come after $\mathsf{FourierO}'$, thus, its applications will be cancel out. We are left with a database D whose support has at most 1 defined point after the query in G_2. The remaining $\geq 2^m - 1$ points are all $(\perp, 0)$. Therefore we can discard all but the pair on the queried point, without affecting the adversary's state. We then have:

$$\begin{aligned}\mathsf{Decomp}'^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}'(|x, z\rangle \otimes |D\rangle) &= \mathsf{Decomp}_x^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}_x(|x, z\rangle \otimes |D\rangle) \\ &= \mathsf{Decomp}^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}(|x, z\rangle \otimes |D\rangle).\end{aligned}$$

This shows that G_3 and G_2 are identical. □

The oracle $\mathsf{CFourierO}_f$ is obtained from $\mathsf{CFourierO}'_f$ by simply decompressing the database D once more time, that is $\mathsf{CFourierO}_f := \mathsf{QFT}^{*D^Y} \circ U_f \circ \mathsf{CFourierO}'_f$, where QFT^* is controlled by the D^Y registers: if D^Y is 0, QFT^* is identity, and QFT otherwise. These computations only act on the oracle's registers, thus $\mathsf{CFourierO}_f$ is also perfectly indistinguishable from $\mathsf{FourierO}_f$.

Now we give concrete computations for the quantum circuit (given in Figure 1). The intermediate states of the circuit are depicted in Figure 3.

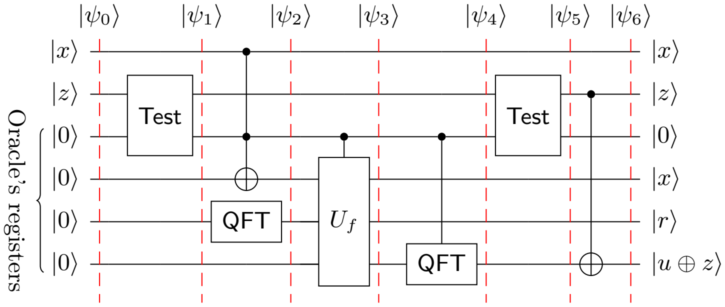

Fig. 3. Quantum circuit for $\mathsf{CFourierO}_f$ oracle.

Let us follow the states through this circuit. We denote the oracle registers as D^b, D^X, D^R, D^Y (in the order from top to bottom). The Test operation writes its output to D^b, which acts as a control bit for later computations. Recall that Test is the unitary defined as $\mathsf{Test}\,|0\rangle\,|b\rangle \mapsto |0\rangle\,|b\rangle$ and $\mathsf{Test}\,|\phi\rangle\,|b\rangle \mapsto |\phi\rangle\,|b \oplus 1\rangle$ for any $|\phi\rangle$ orthogonal to $|0\rangle$ and $b \in \{0, 1\}$. The input state is

$$|\psi_0\rangle = |x, z\rangle \otimes |0\rangle_{D^b}\,|0\rangle_{D^X}\,|0\rangle_{D^R}\,|0\rangle_{D^Y}\,. \qquad (2)$$

Now let us first consider the case $|z\rangle = |0\rangle$. We have

$$|\psi_1\rangle = |x, z\rangle \otimes |0\rangle_{D^b}\,|0\rangle_{D^X}\,|0\rangle_{D^R}\,|0\rangle_{D^Y}\,.$$

In this case, since the control bit is 0, all the controlled operations (except the last one) are just identity. We have

$$\begin{aligned}
|\psi_2\rangle &= |x,z\rangle \otimes |0\rangle_{D^b} |0\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} ,\\
|\psi_3\rangle &= |x,z\rangle \otimes |0\rangle_{D^b} |0\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} ,\\
|\psi_4\rangle &= |x,z\rangle \otimes |0\rangle_{D^b} |0\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} ,\\
|\psi_5\rangle &= |x,z\rangle \otimes |0\rangle_{D^b} |0\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} .
\end{aligned}$$

For the last operation, since $|z\rangle = |0\rangle$, it does not change the value of the register D^Y, that is

$$|\psi_6\rangle = |x,z\rangle \otimes |0\rangle_{D^b} |0\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} . \tag{3}$$

At this step, we can discard D^b, D^X, D^Y registers without affecting the joint system.

Now we consider the case $|z\rangle$ is orthogonal to $|0\rangle$. The input state is still the same as of Equation (2). We have

$$\begin{aligned}
|\psi_1\rangle &= |x,z\rangle \otimes |1\rangle_{D^b} |0\rangle_{D^X} |0\rangle_{D^R} |0\rangle_{D^Y} ,\\
|\psi_2\rangle &= |x,z\rangle \otimes |1\rangle_{D^b} |x\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |0\rangle_{D^Y} .
\end{aligned}$$

Next, the function f is evaluated using U_f acting on D^X, D^R, D^Y, giving

$$|\psi_3\rangle = |x,z\rangle \otimes |1\rangle_{D^b} |x\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} |f(x;r)\rangle_{D^Y} .$$

After the application of QFT on the register D^Y, we have

$$|\psi_4\rangle = |x,z\rangle \otimes |1\rangle_{D^b} |x\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} \frac{1}{\sqrt{2^n}} \sum_{u\in\{0,1\}^n} (-1)^{u\cdot f(x;r)} |u\rangle_{D^Y} .$$

The second application of Test would un-compute it and return D^b back to 0, thus we have

$$|\psi_5\rangle = |x,z\rangle \otimes |0\rangle_{D^b} |x\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} \frac{1}{\sqrt{2^n}} \sum_{u\in\{0,1\}^n} (-1)^{u\cdot f(x;r)} |u\rangle_{D^Y} .$$

Finally, we have

$$
\begin{aligned}
|\psi_6\rangle &= |x,z\rangle \otimes |0\rangle_{D^b} |x\rangle_{D^X} \frac{1}{\sqrt{2^\ell}} \sum_{r\in\{0,1\}^\ell} |r\rangle_{D^R} \frac{1}{\sqrt{2^n}} \sum_{u\in\{0,1\}^n} (-1)^{u\cdot f(x;r)} |u\oplus z\rangle_{D^Y} \\
&= |x,z\rangle \otimes |0\rangle_{D^b} \frac{1}{\sqrt{2^\ell}} \sum_r \sum_u (-1)^{u\cdot f(x;r)} |x,r,u\oplus z\rangle_{D^X D^R D^Y} . \qquad (4)
\end{aligned}
$$

Now we can discard the register D^b.

From Equation (3) and Equation (4), we obtain the same state as stated in Lemma 3. The correctness of the circuit follows immediately.

C One-time Symmetric-key Encryption

C.1 Definition of Security

We define two notions of quantum security for a one-time symmetric-key encryption scheme: security against passive attacks (which is one-time CPA and CCA1 security), and security against adaptive chosen ciphertext attacks (which is one-time CCA2 security).

For passive security, we define a one-time real-or-random oracle allowing at most *one* quantum query as follows.

$$
\mathcal{RR}(b) = \begin{cases} \mathsf{SymEnc}_{\mathtt{k}} & \text{if } b = 1, \\ \mathsf{SymEnc}_{\mathtt{k}} \circ \pi & \text{if } b = 0, \text{where } \pi \xleftarrow{\$} \Pi. \end{cases}
$$

Definition 10 (OT). *We define $\mathcal{A}$'s advantage by*

$$
\mathsf{Adv}^{OT}_{\mathcal{A},\mathcal{SE}}(\lambda) \coloneqq \left| \Pr\left[\mathsf{Expt}^{OT-1}_{\mathcal{SE}}(\lambda,\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}^{OT-0}_{\mathcal{SE}}(\lambda,\mathcal{A}) = 1\right] \right|,
$$

where $\mathsf{Expt}^{OT-b}_{\mathcal{SE}}(\lambda,\mathcal{A})$ is the following experiment:

```
Experiment $\mathsf{Expt}^{OT-b}_{\mathcal{SE}}(\lambda,\mathcal{A})$:
  $\mathtt{k} \xleftarrow{\$} \mathcal{K}()$
  $b' \leftarrow \mathcal{A}^{\mathcal{RR}(b)}(\lambda)$
  **return** $b'$
```

*We say $\mathcal{SE}$ is one-time passively (*OT*) secure if $\mathcal{A}$ being* QPT *implies that* $\mathsf{Adv}^{OT}_{\mathcal{A},\mathcal{SE}}(\lambda)$ *is negligible.*

An adaptive chosen-ciphertext attack is identical to a passive attack, except that *after* receiving the challenge ciphertext, $\mathcal{A}$ may query a decryption oracle any number of times. At the first sight, it seems not trivial to define one-time CCA2 security, for a simple reason that the encryption may be deterministic. In that case, our compressed technique does not work anymore. Fortunately, in one-time security, the adversary does not get to “see the secret key” until it makes a challenge query. This means that, the key need not be explicitly defined beforehand, and it can be sampled "on-the-fly". This observation allows us to consider the key as the randomness, which can be purified in the oracle's implementation. However, now the key, which is needed to answer subsequent decryption queries, is in superposition and certainly not a classical value. Thus, we will use the compressed-then-measure variation to implement the challenge query in the definition for one-time CCA2 security.

Definition 11 (OT-qCCA2)**.** *We define $\mathcal{A}$'s advantage by*

$$\mathsf{Adv}^{OT\text{-}qCCA2}_{\mathcal{A},\mathcal{SE}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{OT\text{-}qCCA2-1}_{\mathcal{SE}}(\lambda,\mathcal{A})=1\right]-\Pr\left[\mathsf{Expt}^{OT\text{-}qCCA2-0}_{\mathcal{SE}}(\lambda,\mathcal{A})=1\right]\right|,$$

where $\mathsf{Expt}^{OT\text{-}qCCA2-b}_{\mathcal{SE}}(\lambda,\mathcal{A})$ are the following experiments:

```
$\mathsf{Expt}^{OT\text{-}qCCA2-b}_{\mathcal{SE}}(\lambda,\mathcal{A})$: // $\boxed{b=1}$ $b=0$
$\boxed{\mathtt{k} \xleftarrow{\$} \mathcal{K}()}$
  $|\Phi_1\rangle = \sum_{x,y} \alpha_{x,y} |x,y,\phi_{x,y}\rangle \leftarrow \mathcal{A}_1(\lambda)$
  $\boxed{|\Phi_2\rangle \leftarrow \mathcal{O}_{\mathsf{SymEnc}_\mathtt{k}} |\Phi_1\rangle}$ $\mathtt{k}, |\Phi_2\rangle \leftarrow \mathsf{CtMStO}^{|\mathtt{k}\rangle}_{\mathsf{SymEnc}} |\Phi_1\rangle$
  $b' \leftarrow \mathcal{A}_2^{\mathsf{CInvO}_{\mathsf{SymDec}_\mathtt{k}}}(|\Phi_2\rangle)$
  return $b'$
```

*We say $\mathcal{SE}$ is one-time CCA2 (*OT-qCCA2*) secure if $\mathcal{A}$ being* QPT *implies that $\mathsf{Adv}^{OT\text{-}qCCA2}_{\mathcal{A},\mathcal{SE}}(\lambda)$ is negligible.*

C.2 Instantiation

Construction 4. *Let* qPRP *be a family of efficient classical permutations* $(\mathsf{qPRP}, \mathsf{qPRP}^{-1})$ *over a message space $\mathcal{X}$ with key space $\mathcal{K}$. We construct the following encryption scheme $\mathcal{SE} = (\mathsf{SymEnc}, \mathsf{SymDec})$ where:*

$$\mathsf{SymEnc}_\mathtt{k}(x) := \mathsf{qPRP}_\mathtt{k}(x)$$
$$\mathsf{SymDec}_\mathtt{k}(y) := \mathsf{qPRP}^{-1}_\mathtt{k}(y)$$

For security, we require qPRP to be quantum secure, i.e., secure against queries on a superposition of inputs. Zhandry [Zha16] shows how to constructs such pseudorandom permutations relying only on the existence of one-way functions.

Theorem 8. *If* qPRP *is a family of quantum-secure pseudorandom permutations, then the encryption given in Construction 4 is one-time passively secure.*

Proof Sketch. The security of the construction follows directly from the security of qPRP: we have that $\mathsf{qPRP}_\mathtt{k}$ is indistinguishable from a truly random permutation σ. Therefore, the adversary cannot distinguish an encryption of x from an encryption of $\pi(x)$. □

To build a symmetric key encryption scheme that achieves one-time security against adaptive chosen ciphertext attacks, we follow the Encrypt-then-MAC paradigm whose security follows directly from our proof in Section 4.3.

Boneh and Zhandry [BZ13a] show how to construct such one-time MACs from a family of 3-*universal hash functions.* A hash family $\{h_k\}_k$ is a 3-universal hash family if for all distinct x_1, x_2, x_3, the distribution of $(h_k(x_1), h_k(x_2), h_k(x_3))$ for a randomly chosen k is uniform.

The construction is as follows.

Construction 5. *Let $\mathcal{SE}$ be a one-time passively secure symmetric encryption, and $\mathcal{MA}$ be a one-time quantum-secure message authentication code scheme. We construct the following encryption scheme $\mathcal{SE}' = (\mathsf{SymEnc}', \mathsf{SymDec}')$ where:*

```
$\mathsf{SymEnc}'_{\mathtt{k}_1\|\mathtt{k}_2}(x)$ :
    $c \leftarrow \mathsf{SymEnc}_{\mathtt{k}_1}(x)$
    $\tau \leftarrow \mathsf{MAC}_{\mathtt{k}_2}(c)$
    return $c\|\tau$

$\mathsf{SymDec}_{\mathtt{k}_1\|\mathtt{k}_2}(c\|\tau)$ :
    if $\mathsf{Ver}_{\mathtt{k}_2}(c, \tau) = 0$ then
        return $\perp$
    $x \leftarrow \mathsf{SymDec}_{\mathtt{k}_1}(c)$
    return $x$
```

D Missing Proofs

D.1 Composability of Our Definitions

Symmetric-key Encryption.

Proof (of Theorem 1). The forward implication follows directly. For the reverse direction, we use the standard hybrid argument that uses an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with advantage ε to construct a new adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ which breaks the single-message security with advantage ε/q^2.

Define a sequence of games $G_0, \ldots, G_q$ in which $\mathcal{B}$ runs $\mathcal{A}$ and returns $\mathcal{A}$'s output as follows: For any game G_i,

1. $\mathcal{B}_1$ simulates $\mathcal{A}$'s $i-1$ first challenge queries as learning queries, that is, $\mathcal{B}$ just forwards $\mathcal{A}$'s directly to its encryption oracle.
2. $\mathcal{B}$ uses $\mathcal{A}$'s i-th challenge query as its challenge query.
3. For all $\mathcal{A}$'s other queries, $\mathcal{B}_2$ first applies a random permutation π to the plaintext registers, sends it to its encryption oracle as a learning query and applies π^{-1} to the plaintext registers before sending it back to $\mathcal{A}$.

In the case of CCA2 security, $\mathcal{B}_2$ needs to be able to record $\mathcal{A}$'s $(i+1, \ldots, q)$-th challenge queries, since it needs to simulate the decryption correctly. Though $\mathcal{B}_2$ has no control over the randomness of the encryption, it can purify the random permutation for each query, allowing it to record $\mathcal{A}$'s queries using the compressed oracle. Formally, $\mathcal{B}_2$ simulates $\mathcal{A}$'s $(i+1, \ldots, q)$-th queries using the compressed encryption oracle $\mathsf{CStO}^{|\pi\rangle}_f[x \mapsto y]$ with $f = \mathsf{SymEnc} \circ \pi$ where π is a random permutation over the message space $\mathcal{X}$, and SymEnc is the unitary implemented by $\mathcal{B}_2$'s encryption oracle. $\mathcal{B}_2$ also uses a slightly different decryption oracle in the second phase as follows. Let $\mathsf{CInvO}'_{\mathsf{SymDec}}$ be the decryption oracle of $\mathcal{B}_2$, D be its database for the challenge query, and $\mathsf{CInvO}_{\mathsf{SymDec}}$ be $\mathcal{B}_2$ simulated decryption oracle for $\mathcal{A}$. Then

$$\mathsf{CInvO}_{\mathsf{SymDec}} |y,z\rangle |D\rangle = \begin{cases} \mathsf{CInvO}'_{\mathsf{SymDec}} |y,z\rangle |D\rangle & \text{if } \mathsf{FindImage}(y,D) = (0, 0^m), \\ |y, z \oplus w\rangle |D\rangle & \text{if } \mathsf{FindImage}(y,D) = (1, w). \end{cases}$$

This oracle can be implemented identically as described in Section 3.2, except that instead of applying $U_{f^{-1}}$, it sends a decryption query on the y, z registers to $\mathsf{CInvO}'_{\mathsf{SymDec}}$.

Note that $G_0 = \mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{SE}}(\lambda, \mathcal{A})$ and $G_q = \mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{SE}}(\lambda, \mathcal{A})$. Because $\mathcal{A}$ is able to distinguish $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{SE}}$ from $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{SE}}$, there exists some $g \in [1, q]$ such that $\mathcal{A}$ distinguishes G_g from G_{g+1} with advantage at least ε/q. $\mathcal{B}$ can guess g correctly with probability $1/q$, thus $\mathcal{B}$'s overall advantage in breaking the single-message security is ε/q^2. □

Public-key Encryption.

Proof (of Theorem 4). The forward implication follows directly. For the reverse direction, we use the standard hybrid argument that uses an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with advantage ε to construct a new adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ which breaks the single-message security with advantage ε/q^2.

Define a sequence of games $G_0, \ldots, G_q$ in which $\mathcal{B}$ runs $\mathcal{A}$ and returns $\mathcal{A}$'s output as follows: For any game G_i,

1. $\mathcal{B}_1$ simulates $\mathcal{A}$'s $i-1$ first challenge queries on its own, as in the experiment $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}$.
2. $\mathcal{B}$ uses $\mathcal{A}$'s i-th challenge query as its challenge query.
3. $\mathcal{B}_2$ simulates all $\mathcal{A}$'s other queries on its own using the compressed encryption oracle, as in the experiment $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}$.

In the case of CCA2 security, $\mathcal{B}_2$ uses a slightly different decryption oracle in the second phase as follows. Let $\mathsf{CInvO}'_{\mathsf{Dec}_{\mathsf{sk}}}$ be the decryption oracle of $\mathcal{B}_2$, D be its database for the challenge query, and $\mathsf{CInvO}_{\mathsf{Dec}}$ be $\mathcal{B}_2$ simulated decryption oracle for $\mathcal{A}$. Then

$$\mathsf{CInvO}_{\mathsf{Dec}} |y,z\rangle |D\rangle = \begin{cases} (\mathsf{CInvO}'_{\mathsf{Dec}_{\mathsf{sk}}} |y,z\rangle) |D\rangle & \text{if } \mathsf{FindImage}(y,D) = (0, 0^m), \\ |y, z \oplus w\rangle |D\rangle & \text{if } \mathsf{FindImage}(y,D) = (1, w). \end{cases}$$

Note that $G_0 = \mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}(\lambda, \mathcal{A})$ and $G_q = \mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}(\lambda, \mathcal{A})$. Because $\mathcal{A}$ is able to distinguish $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}$ from $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}$, there exists some $g \in [1, q]$ such that $\mathcal{A}$ distinguishes G_g from G_{g+1} with advantage at least ε/q. $\mathcal{B}$ can guess g correctly with probability $1/q$, thus $\mathcal{B}$'s overall advantage in breaking the single-message security is ε/q^2. □

D.2 Relating Indistinguishability and Non-Malleability

Theorem 9 ($\mathsf{qIND\text{-}qCCA1} \not\Rightarrow \mathsf{qNME\text{-}qCPA}$)**.** *If there exists an encryption scheme $\mathcal{E}$ that is* $\mathsf{qIND\text{-}qCCA1}$ *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* $\mathsf{qIND\text{-}qCCA1}$ *secure but* $\mathsf{qNME\text{-}qCPA}$ *insecure.*

Proof. Assume there exists some $\mathsf{qIND\text{-}qCCA1}$ secure encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. The new encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is defined as follows.

```
$\mathsf{KeyGen}'(\lambda)$ :
    $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
    return $(\mathsf{pk}, \mathsf{sk})$
```

```
$\mathsf{Enc}'_{\mathsf{pk}}(x)$ :
    $b \xleftarrow{\$} \{0,1\}$
    $y \leftarrow \mathsf{Enc}_{\mathsf{pk}}(x)$
    return $y \| b$
```

```
$\mathsf{Dec}'_{\mathsf{sk}}(y \| b)$ :
    $x \leftarrow \mathsf{Dec}_{\mathsf{sk}}(y)$
    return $x$
```

Claim 9.1. $\mathcal{E}'$ is $\mathsf{qNME\text{-}qCPA}$ insecure.

Proof Sketch. The scheme is malleable because given a ciphertext $y \| b$ of a plaintext x, it is trivial to create another ciphertext of x by just outputting $y \| \bar{b}$. □

Claim 9.2. $\mathcal{E}'$ is $\mathsf{qIND\text{-}qCCA1}$ secure.

Proof Sketch. It is easy to see that any adversary $\mathcal{A}$ against $\mathcal{E}'$ can be used to construct an adversary $\mathcal{B}$ that attacks $\mathcal{E}$ as follows. $\mathcal{B}$ runs $\mathcal{A}$ using its own oracle $\mathcal{O}_1$, and uses $\mathcal{A}$'s challenge queries as its own challenge queries. Whenever $\mathcal{B}$ receives a challenge ciphertext, it samples a random bit b and appends it to the challenge ciphertext before forwarding it to $\mathcal{A}$. $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. One can verify that $\mathsf{Adv}_{\mathcal{B},\mathcal{E}}(\lambda) = \mathsf{Adv}_{\mathcal{A},\mathcal{E}'}(\lambda)$. Thus, the security of $\mathcal{E}'$ follows from the security of $\mathcal{E}$. □

□

Theorem 10 ($\mathsf{qNME\text{-}qCPA} \not\Rightarrow \mathsf{qIND\text{-}qCCA1}$)**.** *If there exists an encryption scheme $\mathcal{E}$ that is* $\mathsf{qNME\text{-}qCPA}$ *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* $\mathsf{qNME\text{-}qCPA}$ *secure but* $\mathsf{qIND\text{-}qCCA1}$ *insecure.*

Proof. Assume there exists some $\mathsf{qNME\text{-}qCPA}$ secure encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Fix a family $\mathsf{qPRP} = \{\mathsf{qPRP}_{\mathtt{k}} : \{0,1\}^{\ell} \to \{0,1\}^{\ell}\}$ of quantum-secure pseudorandom permutations. The new encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is defined as follows.

```
$\mathsf{KeyGen}'(\lambda)$ :
    $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
    $\mathtt{k} \xleftarrow{\$} \{0,1\}^{\lambda}$
    $\mathsf{sk}' \leftarrow \mathsf{sk} \| \mathtt{k}$
    return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Enc}'_{\mathsf{pk}}(x)$ :
    $y \leftarrow \mathsf{Enc}_{\mathsf{pk}}(x)$
    return $0 \| y$
```

```
$\mathsf{Dec}'_{\mathsf{sk}\|\mathtt{k}}(b \| y)$ :
    if $b = 0$ :
        return $\mathsf{Dec}_{\mathsf{sk}}(y)$
    else if $y = \mathsf{qPRP}_{\mathtt{k}}(0)$ :
        return $\mathsf{sk}$
    else return $\mathsf{qPRP}_{\mathtt{k}}(y)$
```

Claim 10.1. $\mathcal{E}'$ is $\mathsf{qIND\text{-}qCCA1}$ insecure.

Proof Sketch. The adversary queries $\mathsf{Dec}'_{\mathtt{sk}\|\mathtt{k}}(\cdot)$ at $1\|0$ to get $v = \mathsf{qPRP}_{\mathtt{k}}(0)$, and then queries it at the point $1\|v$ to get $\mathtt{sk}$. At this point, the adversary can obviously break the security of $\mathcal{E}'$. □

Claim 10.2. $\mathcal{E}'$ is $\mathsf{qNME\text{-}qCPA}$ secure.

Proof. Fix $\mathcal{A}$ and λ. We prove security through a sequence of games.

Game G_0: This is the standard attack game.

Game G_1: Replace qPRP with a truly random function H.

Since qPRP is a quantum-secure pseudorandom permutation, $\mathcal{A}$ cannot distinguish G_1 from G_0, except with negligible probability.

Game G_2: This is identical to G_1. The only change is to the decryption algorithm, in which instead of returning $\mathtt{sk}$ when $y = H(0)$, it returns $H(H(0))$ which is a random value independent from the secret key $\mathtt{sk}$.

Games G_1 and G_2 proceed identically unless $\mathcal{A}$ successfully outputs $H(H(0))$ with a single query. To bound the distinguishing probability, we invoke the following lemma.

Lemma 11 ([Unr15, Theorem 6.6]). *Let $\mathcal{A}$ be any quantum oracle algorithm making a single query to a random function H, with r inputs in the query. Then*

$$\Pr\Big[x = H(H(0)) : H \overset{\$}{\leftarrow} (\{0,1\}^{\ell} \to \{0,1\}^{\ell}), x \leftarrow \mathcal{A}^{H}()\Big] \leq 2^{-\Omega(\ell)} O(r).$$

This probability is negligible for polynomially-bounded r (number of inputs per query, which corresponds to the number of adversarial ciphertexts in a qNME security game).

Finally, we design an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ attacking $\mathcal{E}$ in the $\mathsf{qNME\text{-}qCPA}$ sense from the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ in this last game. $\mathcal{B}$ runs $\mathcal{A}$ as its subroutine and simulates a random oracle H itself. $\mathcal{B}_1$ and $\mathcal{B}_3$ output whatever $\mathcal{A}_1$ and $\mathcal{A}_3$ output, respectively. The algorithm $\mathcal{B}_2$ is defined as follows. $\mathcal{B}_2$ receives a vector (in superposition) of adversarial ciphertexts from $\mathcal{A}_2$.

- If the basis state is $|1\|y, z, \phi_{y,z}\rangle$, then it maps this basis state to $|\mathsf{Enc}(H(y)), z, 1\|\phi'_{y,z}\rangle$ by allocating new ancilla registers (with proper padding), computing $\mathsf{Enc}(H(y))$ and then swapping these newly created registers with the y registers. The y registers are now included in the auxiliary registers $|\phi'_{y,z}\rangle$.
- Otherwise, it keeps the basis state the same, re-organizes the state to $|y, z\rangle\,|0\|\phi_{y,z}\rangle$.

$\mathcal{B}_2$ then outputs the resulting state as its adversarial ciphertexts.

Let D be the database of $\mathcal{B}$'s challenge queries. Consider $\mathcal{B}_2$'s adversarial ciphertexts state, let Dup be the event that this state has a non-negligible weight on ciphertexts $\mathsf{Enc}(H(y))$ such that $\mathsf{Enc}(H(y)) \in D$. The simulation is indistinguishable if this happens with non-negligible probability. To see that, imagine that in the real-world experiment, $\mathcal{A}_3$ would receive exactly $H(y)$. In the random-world experiment, $\mathsf{Enc}(H(y)) \in D$ means that $H(y)$ is a random message obtained by apply a random permutation π. $\mathcal{A}_3$ would receive $\pi^{-1}(H(y))$ (by the definition of the qNME decryption oracle).

However, we show that $\Pr[\mathsf{Dup}]$ must be negligible, otherwise it would violate the security of $\mathcal{E}$ even in the qIND-qCPA. This is a standard argument, we omit the details. The security of $\mathcal{E}'$ now follows by the security of $\mathcal{E}$. □

□

Theorem 11 (qNME-$\mathsf{qCCA1} \not\Rightarrow \mathsf{qNME}$-$\mathsf{qCCA2}$)**.** *If there exists an encryption scheme $\mathcal{E}$ that is* qNME-$\mathsf{qCCA1}$ *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* qNME-$\mathsf{qCCA1}$ *secure but* qNME-$\mathsf{qCCA2}$ *insecure.*

Proof. Assume there exists some qNME-$\mathsf{qCCA1}$ secure encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Fix a family $\mathsf{qPRF} = \{\mathsf{qPRF}_{\mathtt{k}}\}$ of quantum-secure pseudorandom functions. The new encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is defined as follows.

```
$\mathsf{KeyGen}'(\lambda)$ :
$(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
$\mathtt{k} \xleftarrow{\$} \{0,1\}^{\lambda}$
$\mathtt{sk}' \leftarrow \mathtt{sk}\|\mathtt{k}$
return $(\mathtt{pk}, \mathtt{sk}')$
```

```
$\mathsf{Enc}'_{\mathtt{pk}}(x)$ :
$y \leftarrow \mathsf{Enc}_{\mathtt{pk}}(x)$
return $0\|y\|0$
```

```
$\mathsf{Dec}'_{\mathtt{sk}\|\mathtt{k}}(b\|y\|z)$ :
if $b = 0 \wedge z = 0$ :
    return $\mathsf{Dec}_{\mathtt{sk}}(y)$
else if $b = 1$ :
    return $\mathsf{qPRF}_{\mathtt{k}}(y)$
else if $b = 2 \wedge z = \mathsf{qPRF}_{\mathtt{k}}(y)$ :
    return $\mathsf{Dec}_{\mathtt{sk}}(y)$
else return $\perp$
```

Claim 11.1. $\mathcal{E}'$ is qNME-$\mathsf{qCCA2}$ insecure.

Proof Sketch. Let $0\|y\|0$ be the classical challenge ciphertext. The adversary first queries $\mathsf{Dec}'_{\mathtt{sk}\|\mathtt{k}}(\cdot)$ at $1\|y\|0$ (which is not the challenge ciphertext) to get $v = \mathsf{qPRF}_{\mathtt{k}}(y)$, and then queries it at the point $2\|y\|v$ to get the decryption of y, which is exactly the decryption of the challenge ciphertext. This helps the adversary to break the indistinguishability in the sense of qNME-$\mathsf{qCCA2}$. □

Claim 11.2. $\mathcal{E}'$ is qNME-$\mathsf{qCCA1}$ secure.

Proof. The proof is similar to that of Claim 10.2: first the pseudorandom function qPRF is replaced by a truly random function H, and for any decryption query of the form $2\|y\|z$, we return $\perp$ where y is the challenge ciphertext.

The extra step is that we need to consider the case in which the adversary happens to query to the random function involving the challenge ciphertext. However, such event is unlikely since otherwise the scheme $\mathcal{E}$ would not be secure even in the sense of qIND-$\mathsf{qCCA1}$. We formally prove the security through a sequence of games. Fix $\mathcal{A}$ and λ.

Game G_0: This is the standard attack game.

Game G_1: Replace qPRF with a truly random function H.

Since qPRF is a quantum-secure pseudorandom function, $\mathcal{A}$ cannot distinguish G_1 from G_0, except with negligible probability.

Game G_2: This is identical to G_1, except that now we will consider the encryption oracle and the decryption oracle where the random function H is involved as being implemented in the compressed oracle. Since these are equivalent to the standard oracles, these changes do not affect the adversary's success probability. We have $\Pr[G_2] = \Pr[G_1]$.

Game G_3: This is identical to G_2. Let D be the database of the challenge queries. The only change is to the decryption algorithm which is used in the last phase after the adversary has output its adversarial ciphertexts: if the ciphertext is $2\|y\|H(y)$ where $y \in D$ (in the form of $0\|y\|0$), then it returns $\perp$.

The intuition is that the adversary cannot make such a query (i.e., to put a non-negligible weight on inputs $2\|y\|H(y)$ where $y \in D$), except with negligible probability. Thus, the change is undetectable by the adversary. We formally bound the distinguishing probability between G_2 and G_3 by considering the two following events.

– Let $\mathsf{ForgeOffline}$ be the event that $\mathcal{A}_1$ (in the first phase) has a non-negligible query weight on inputs containing some $y \in D$ in its queries to H. A result of Zhandry [Zha19, Lemma 5] shows that the success probability of a quantum adversary in an standard oracle game is close to its success probability in the corresponding compressed oracle game.

  Lemma 12 ([Zha19, Lemma 5]). *Let p be the probability that an adversary making queries to a random oracle $H : \{0,1\}^m \leftarrow \{0,1\}^n$ and outputting a tuple $(\mathbf{a}, \mathbf{b}, c)$ such that $|\mathbf{a}| = |\mathbf{b}| = k$ and $H(a_i) = b_i$ for each $i \in [k]$. Let R be a collection of such tuples. Now consider running the adversary with the compressed oracle, and we measure the database D after the adversary procedures its output. Let p' be the probability that there exists a tuple $(\mathbf{a}', \mathbf{b}', c') \in R$ such that $D(a'_i) = b'_i$ for each $i \in [k]$. Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/2^n}$.*

  We now show that if $\mathsf{ForgeOffline}$ happens with non-negligible probability, we could design an adversary $\mathcal{B}$ that break $\mathcal{E}$ in the sense of $\mathsf{qIND\text{-}qCCA1}$.

  - In the first stage, $\mathcal{B}$ implements a compressed random oracle and provides a simulation of the decryption oracle of $\mathcal{A}$ using its decryption oracle. Let D^H is the database kept by $\mathcal{B}$.
  - When $\mathcal{A}$ outputs its challenge, $\mathcal{B}$ measures its database D^H and gets many pairs $D^H = \{(y, H(y))\}$. $\mathcal{B}$ then submits these y values to its decryption oracle, which are legitimately counted as decryption in the first phase, and gets back their plaintexts x. Only at this point, $\mathcal{B}$ outputs $\mathcal{A}$'s challenge as its challenge. After receiving back the challenge ciphertexts, $\mathcal{B}$ measures its challenge query, and checks if there is any value in D^H. If it does then it outputs a bit b depending on whether their plaintexts are the same, otherwise it decides by flipping a coin. Observe

that the success of $\mathcal{B}$ is exponentially close to one half the probability of $\mathsf{ForgeOffline}$ (by Lemma 12 and the standard argument).

Thus, we have $\Pr[\mathsf{ForgeOffline}]$ must be negligible.

– Let $\mathsf{ForgeOnline}$ be the event that the adversary correctly computes $H(H(y))$ for some $y \in D$ using only a single query to H in the last phase. By a similar argument to Lemma 11, we have that $\Pr[\mathsf{ForgeOnline}]$ is negligible.

Therefore, we have that

$$|\Pr[G_3] - \Pr[G_2]| \leq \Pr[\mathsf{ForgeOffline}] + \Pr[\mathsf{ForgeOnline}],$$

which is negligible.

Finally, we construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ that attacks $\mathcal{E}$ in the sense of qNME-qCCA1 from any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ of this last game. This can be argued analogously to the argument in Claim 10.2. We omit the details. □

□

D.3 A Separation Example for Public-key Encryption

Proof (of Theorem 6). Assume there exists some IND-qCCA2 secure encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Let $\mathcal{H} = \{h_k\}_k$ be a family of pairwise independent hash functions with the key space $\mathcal{K}$. The new encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is defined as follows.

```
$\mathsf{KeyGen}'(\lambda)$ :
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
return $(\mathsf{pk}, \mathsf{sk})$
```

```
$\mathsf{Enc}'_{\mathsf{pk}}(x)$ :
$r \xleftarrow{\$} \mathcal{X}, k \xleftarrow{\$} \mathcal{K}$
$c_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(r\|k)$
$c_2 \leftarrow x \oplus r$
$\sigma \leftarrow h_k(c_2)$
return $c_1\|c_2\|\sigma$
```

```
$\mathsf{Dec}'_{\mathsf{sk}}(c_1\|c_2\|\sigma)$ :
$r\|k \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_1)$
if $h_k(c_2) \neq \sigma$ then
    return $\perp$
$x \leftarrow c_2 \oplus r$
return $x$
```

The proof is completed by establishing that $\mathcal{E}'$ is IND-qCCA2 secure but vulnerable to a qIND-qCPA attack.

Lemma 13. *$\mathcal{E}'$ is IND-qCCA2 secure.*

Proof. Fix the adversary $\mathcal{A}$ and λ. For the purpose of this separation, it is sufficient to assume that $\mathcal{E}$ is perfectly correct. We prove security through a sequence of games. Let $\Pr[G_i]$ be the probability the adversary wins game G_i.

Game G_0: This is the standard attack game. Let the challenge ciphertext be (c_1^*, c_2^*, σ^*), and $K^* = (r^*, k^*)$ be the randomness used during the encryption process. Then, the decryption oracle in the second phase can be written as $\widetilde{\mathsf{Dec}}^{D_0}(\cdot)$ with $D_0 = \{(c_1^*, c_2^*, \sigma^*)\}$.

Game G_1: This is identical to G_0, except that whenever a ciphertext $(c_1^*, \cdot, \cdot) \in D_0$ is submitted to the decryption oracle in the second phase, the decryption oracle does not apply $\mathsf{Dec}_{\mathsf{sk}}(c_1^*)$, but instead uses K^* produced in the challenge phase to perform steps $2 - 5$.

This change is just conceptual, since we assume that $\mathcal{E}$ is perfectly correct. Thus, $\Pr[G_1] = \Pr[G_0]$.

Game G_2: This is identical to G_1, but now the challenger computes c_1^* by encrypting a completely random value $K^+ = (r^+, k^+)$ instead of K^*. That is, $c_1^* = \mathsf{Enc}_{\mathsf{pk}}(r^+ \| k^+)$, but $c_2^* = x \oplus r^*$ and $\sigma^* = h_{k^*}(c_2^*)$.

Notice that in games G_1 and G_2, the ciphertext c_1^* need not be submitted for decryption. We show how to turn any distinguisher $\mathcal{A}$ of games G_1 and G_2 into an adversary $\mathcal{A}'$ against the security of the underlying scheme $\mathcal{E}$: $\mathcal{A}'$ runs $\mathcal{A}$ using its oracles to answer $\mathcal{A}$, outputs (K^*, K^+) as its challenge pair. Finally, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. It is easy to see that we have:

$$|\Pr[G_2] - \Pr[G_1]| \leq \mathsf{Adv}_{\mathcal{A}',\mathcal{E}}^{\textit{ind-qcca2}}(\lambda).$$

Game G_3: We further modify G_2 and now change the oracle $\widetilde{\mathsf{Dec}}^{D_0}(\cdot)$ to be $\widetilde{\mathsf{Dec}}^{D_1}(\cdot)$ with $D_1 = \{(c_1^*, \cdot, \cdot)\}$. In other words, it rejects any ciphertext (c_1, c_2, σ) such that $c_1 = c_1^*$.

Let Forge be the event that some ciphertext is rejected in game G_3, but would not have been rejected in the game G_2. Since games G_2 and G_3 are identical until event Forge, we have $|\Pr[G_3] - \Pr[G_2]| \leq \Pr[\mathsf{Forge}]$.

Notice that in the construction of $\mathcal{E}'$, the use of pairwise independent hash functions acts as a one-time secure message authentication code, thus $\Pr[\mathsf{Forge}] = 0$.

In this final game, the component c_2^* is one-time padded of the message x_b^* using a random string r^* chosen uniformly and independently of all other variables, including b. Thus, $\Pr[G_3] = 0$.

By the security of the underlying building blocks, we have the security of $\mathcal{E}'$. □

Lemma 14. *$\mathcal{E}'$ is* qIND-qCPA *insecure.*

Proof. In the challenge phase, the adversary $\mathcal{A}$ chooses two fixed messages x_0, x_1, and prepares the following state as its challenge:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |0\rangle |+\rangle .$$

The challenge ciphertext state will be:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |x_b \oplus r\rangle |+\rangle \text{ if } b = 0,$$

or

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |\pi(x_b) \oplus r\rangle |+\rangle \text{ if } b = 1.$$

$\mathcal{A}$ then applies the Fourier sampling (as described in Section 4.2) and breaks the security of $\mathcal{E}'$ with non-negligible probability. □

□

D.4 Quantum CCA2 Security of Encrypt-then-MAC

Figure 4 describes the hybrid games in the proof of Theorem 3.

```
Experiment $\mathsf{Expt}^{qind\text{-}qcca2-b}_{\mathcal{SE}'}$ in $G_0$, $\boxed{G_1, G_2, G_3, G_4}$

  Define $\mathcal{E} = \mathsf{SymEnc}'_{\mathtt{k}}\ \boxed{\mathsf{CStO}_{\mathsf{SymEnc}'}}$
         $\mathcal{D} = \mathsf{SymDec}'_{\mathtt{k}}\ \boxed{\mathsf{CInvO}_{\mathsf{SymDec}'}}$
         $\mathcal{RR}(b) = \begin{cases} \mathsf{SymEnc}'_{\mathtt{k}}\ \boxed{\mathsf{CStO}_{\mathsf{SymEnc}'}} & \text{if } b = 1, \\ \mathsf{CStO}_{\mathsf{SymEnc}' \circ \pi} & \text{if } b = 0, \text{where } \pi \xleftarrow{\$} \Pi. \end{cases}$
  $\mathtt{k} = \mathtt{k}_1 \parallel \mathtt{k}_2 \xleftarrow{\$} \mathcal{K}'()$
  $|\phi\rangle \leftarrow \mathcal{A}_1^{\mathcal{E},\mathcal{D}}(\lambda)$
  $b' \leftarrow \mathcal{A}_2^{\mathcal{RR}(b),\mathcal{E},\mathsf{CInvO}_{\mathsf{SymDec}'}}(|\phi\rangle)$
  return $b'$

$\mathsf{SymEnc}'_{\mathtt{k}_1 \| \mathtt{k}_2}(x)$ : // $G_0, G_1, G_2$ $\boxed{G_3, G_4}$
  $\boxed{\mathtt{k}_2^* \xleftarrow{\$} \mathcal{K}_{\mathcal{MA}}()}$
  $c \leftarrow \mathsf{SymEnc}_{\mathtt{k}_1}(x)$
  $\tau \leftarrow \mathsf{MAC}_{\mathtt{k}_2}(c)\ \boxed{\mathsf{MAC}_{\mathtt{k}_2^*}(c)}$
  return $c \parallel \tau$

$\mathsf{CStO}_{\mathsf{SymEnc}'}$ // $G_0, G_1, G_2, G_3$ $\boxed{G_4}$
  Implemented as
    $\mathsf{CStO}_{\mathsf{SymEnc}'}^{|r\rangle\ \boxed{|\mathtt{k}_2^*\rangle}}[x \mapsto c \| \tau]$

$\mathsf{SymDec}'_{\mathtt{k}_1 \| \mathtt{k}_2}(c \parallel \tau)$ :
  if $\mathsf{Ver}_{\mathtt{k}_2}(c, \tau) = \bot$ then
    return $\bot$
  $x \leftarrow \mathsf{SymDec}_{\mathtt{k}_1}(c)$
  return $x$

$\mathsf{CInvO}_{\mathsf{SymDec}'}\ |c \| \tau, z, D\rangle$ // $G_0, G_1$ $\boxed{G_2, G_3, G_4}$
  if $\mathsf{FindImage}(c \| \tau, D) = (1, w)$ then
    return $|y, z \oplus w\rangle \otimes |D\rangle$
  return $|y, z \oplus \mathsf{SymDec}'_{\mathtt{k}}(y)\ \boxed{\bot}\rangle \otimes |D\rangle$
```

Fig. 4. Games for the proof of Theorem 3. In each procedure, the components inside a gray (solid) frame are only present in the games marked by a gray (solid) frame.

Quantum circuits of the unitary U_f implemented by the reduction algorithms of Claim 3.2 and Claim 3.4 (Theorem 3) are given in Figure 5. This unitary acts on three registers: the input registers, the purification registers, and the output registers.

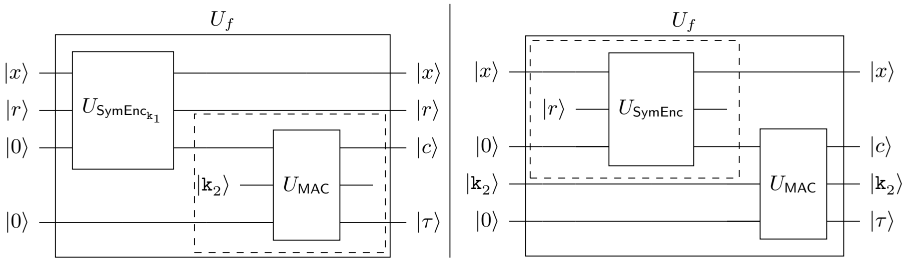

Fig. 5. Implementations of quantum circuit for the reduction algorithms of Claim 3.2 (left) and Claim 3.4 (right) in the proof of Theorem 3. The dotted box is the unitary implemented by some oracle (i.e., the challenger in the reduction algorithms), who controls all registers inside that box.

D.5 From IND-qCCA2 to qIND-qCCA2

```
Experiment $\mathsf{Expt}^{qind\text{-}qcca2-b}_{\mathcal{E}'}$ in $\colorbox{lightgray}{$G_0$}$, $\boxed{G_1, G_2, G_3, G_4}$
    Define $\mathcal{RR}(b) = \begin{cases} \colorbox{lightgray}{$\mathsf{Enc'}_{\mathtt{pk}}$}\ \boxed{\mathsf{CStO}_{\mathsf{Enc'}}} & \text{if } b = 1, \\ \mathsf{CStO}_{\mathsf{Enc'}\circ\pi} & \text{if } b = 0, \text{where } \pi \xleftarrow{\$} \Pi. \end{cases}$
    $(\mathtt{pk}, \mathtt{sk}) \xleftarrow{\$} \mathsf{KeyGen'}(\lambda)$
    $|\phi\rangle \leftarrow \mathcal{A}_1^{\mathsf{Dec'}_{\mathtt{sk}}}(\lambda)$
    $b' \leftarrow \mathcal{A}_2^{\mathcal{RR}(b), \mathsf{CInvO}_{\mathsf{Dec'}}}(|\phi\rangle)$
    return $b'$

$\mathsf{Enc'}_{\mathtt{pk}}(x)$ : // $\colorbox{lightgray}{$G_0, G_1, G_2, G_3$}$ $\boxed{G_4}$
    $\mathtt{k}^* \xleftarrow{\$} \mathcal{K}$, $\boxed{\mathtt{k}^+ \xleftarrow{\$} \mathcal{K}}$
    $c_1 \leftarrow \colorbox{lightgray}{$\mathsf{Enc}_{\mathtt{pk}}(\mathtt{k}^*)$}\ \boxed{\mathsf{Enc}_{\mathtt{pk}}(\mathtt{k}^+)}$
    $c_2 \leftarrow \mathsf{SymEnc}_{\mathtt{k}^*}(x)$
    return $c_1 \| c_2$

$\mathsf{CStO}_{\mathsf{Enc'}}$ // $\colorbox{lightgray}{$G_0, G_1, G_2$}$ $\boxed{G_3, G_4}$
    Implemented as
        $\colorbox{lightgray}{$\mathsf{CStO}^{|r\rangle}_{\mathsf{Enc'}}[x \mapsto c_1 \| c_2]$}$
        $\boxed{\mathsf{CtMStO}^{|r\rangle}_{\mathsf{Enc'}}[x \mapsto c_1 \| c_2]}$

$\mathsf{Dec'}_{\mathtt{sk}}(c_1 \| c_2)$ :
    $\mathtt{k} \leftarrow \mathsf{Dec}_{\mathtt{sk}}(c_1)$
    $x \leftarrow \mathsf{SymDec}_{\mathtt{k}}(c_2)$
    return $x$

$\mathsf{CInvO}_{\mathsf{Dec'}}\ |c_1 \| c_2, z, D\rangle$ // $\colorbox{lightgray}{$G_0, G_1$}$ $\boxed{G_2, G_3, G_4}$
    if $\mathsf{FindImage}(c_1 \| c_2, D) = (1, w)$ then
        return $|y, z \oplus \colorbox{lightgray}{$w$}\ \boxed{\mathsf{SymDec}_{\mathtt{k}^*}(c_2)}\rangle \otimes |D\rangle$
    return $|y, z \oplus \mathsf{Dec'}_{\mathtt{sk}}(c_1 \| c_2)\rangle \otimes |D\rangle$
```

Fig. 6. Games for the proof of Theorem 7. In each procedure, the components inside a gray (solid) frame are only present in the games marked by a gray (solid) frame.

Proof (of Theorem 7). We prove using hybrid games, described in Figure 6. Since our definitions are closed under composition, it is sufficient to prove for the single-message security.

Let $\mathcal{A}$ be a QPT adversary. For any game G_{index}, we denote by $\Pr[G_{\mathsf{index}}] := |\Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 1] - \Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 0]|$. Also, by event $G_{\mathsf{index}}(\mathcal{A})$, we mean the output of the experiments (defined as in Definition 3) in game G_{index} when interacting with $\mathcal{A}$.

Game G_0: This is the standard attack game. Let $\mathtt{k}^*$ denote the symmetric key used during the encryption process within the oracle.

Game G_1: This is identical to G_0, except that now the real-or-random oracle $\mathcal{RR}(b)$ will be implemented as a compressed oracle (in both cases $b = 1$ and $b = 0$). Let D be the database of the encryption oracle. Then the decryption oracle in the second phase can be written as

$$\mathsf{CInvO}^1_{\mathsf{Dec}'} |y, z\rangle |D\rangle = \begin{cases} |y, z \oplus \mathsf{Dec'}_{\mathtt{sk}}(y)\rangle |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (0,0), \\ |y, z \oplus w\rangle |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w), \end{cases}$$

where $\mathsf{FindImage}$ parses its input component y as $y = (c_1, c_2)$.

Since $\mathcal{SE}$ is perfectly correct (by definition), any decryption failure of $\mathcal{E}'$ is a decryption failure of $\mathcal{E}$. Thus $\mathcal{E}'$ is also δ-correct. By Lemma 8, we have $|\Pr[G_1] = \Pr[G_0]| \leq O(q_d \cdot \delta)$.

Game G_2: We define $\mathsf{FindImage}'$ that takes as input a tuple $(c_1, \cdot, D)$ and returns $(1, w)$ if there is any pair $(c_1, \cdot)$ in D, ignoring the second component, and $(0, 0)$ otherwise. This is identical to G_1, except that we change the decryption oracle in the second phase to

$$\mathsf{CInvO}^2_{\mathsf{Dec}'} |y, z\rangle |D\rangle = \begin{cases} |y, z \oplus \mathsf{Dec'}_{\mathtt{sk}}(y)\rangle |D\rangle & \text{if } \mathsf{FindImage}'(y, D) = (0,0), \\ |y, z \oplus \mathsf{SymDec}_{\mathtt{k}^*}(c_2)\rangle |D\rangle & \text{if } \mathsf{FindImage}'(y, D) = (1, w), \end{cases}$$

where $\mathsf{FindImage}'$ parses its input component y as $y = (c_1, c_2)$.

Let $\mathsf{DecFail}$ be the event that $\mathsf{Dec'}_{\mathtt{sk}}(\mathsf{Enc'}_{\mathtt{pk}}(x))) = x' \neq x$. Unless this event occurs, G_2 and G_1 proceed identically. We thus have

$$|\Pr[G_2] - \Pr[G_1]| \leq \Pr[\mathsf{DecFail}] \leq \delta,$$

where the last inequality follows from the definition of correctness.

Game G_3: This is identical to G_2, except that we change the implementation of the compressed oracle from $\mathsf{CStO}^{|r\rangle}[x \mapsto c_1 \| c_2]$ to $\mathsf{CtMStO}^{|r\rangle}[x \mapsto c_1 \| c_2]$. That is, we measure the purification registers (which is the randomness used in $\mathcal{E}$'s encryption) after the query.

Since the two oracle variations are equivalent, this change does not affect the adversary's success probability. We thus have $\Pr[G_3] = \Pr[G_2]$.

Notice that the same symmetric key $\mathtt{k}^*$ sampled during the encryption process within the challenger's oracle is used for all classical states of the superposition, the ciphertext state of the challenge query would be:

$$\sum_{x,y} \alpha_{x,y} \left| x, y \right\rangle \to \left| \mathsf{Enc}_{\mathtt{pk}}(\mathtt{k}^*) \right\rangle \sum_{x,y} \alpha_{x,y} \left| x, y \oplus \mathsf{SymEnc}_{\mathtt{k}^*}(x_b) \right\rangle , \qquad (5)$$

where x_b denotes the actual encrypted plaintext, depending on whether it is the real-world ($b = 1$) or the random-world ($b = 0$) (but the key $\mathtt{k}$ is independent of b). Notice that the first component c_1 of the ciphertext is a classical value.

Game G_4: This is identical to G_3, except that in the real-or-random oracle, we encrypt a complete random value $\mathtt{k}^+$ in place of the symmetric key $\mathtt{k}^*$, that is we compute $c_1 = \mathsf{Enc}_{\mathtt{pk}}(\mathtt{k}^+)$, but we still use $\mathtt{k}^*$ for symmetric encryption and decryption.

It is straightforward to see that any adversary $\mathcal{A}$ that distinguishes games G_4 from G_3 can be turned to an adversary $\mathcal{B}$ attacking the underlying scheme $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{A}$. The adversary $\mathcal{B}$ just runs the adversary $\mathcal{A}$, and uses $(\mathtt{k}^*, \mathtt{k}^+)$ as its challenge pair. Note that in both games, the challenge ciphertext of $\mathcal{B}$ is c_1, which is classical, as argued above, and that $\mathcal{B}$ never query to the decryption oracle on the challenge ciphertext, but instead uses its database to answer the query. We have

$$|\Pr[G_4] - \Pr[G_3]| \leq \mathsf{Adv}_{\mathcal{B},\mathcal{E}}^{\textit{ind-qcca2}}(\lambda).$$

Furthermore, notice the fact that in this final game, the symmetric key $\mathtt{k}^*$ is independent of the adversary's view and b, we now turn any distinguisher $\mathcal{A}$ of in this game to an adversary $\mathcal{C}$ that breaks the one-time security of $\mathcal{SE}$. $\mathcal{C}$ runs $\mathcal{A}$, when it receives the challenge query from $\mathcal{A}$, it first generates a random string k^+ and encrypt it with the public key $\mathtt{pk}$ to get c_1, and sends $\mathcal{A}$'s challenge query directly to its challenger. After receiving the answer back, $\mathcal{C}$ appends $|c_1\rangle$ to the result and forwards it to $\mathcal{A}$. In the second phase, for any decryption query that contains c_1 (from its database), $\mathcal{C}$ just forwards the query to its challenger (which can be implemented similarly as the inverse oracle described in Section 3.2).

By the security of $\mathcal{SE}$ we have $|\Pr[G_4] - \Pr[G_3]| \leq \mathsf{Adv}_{\mathcal{C},\mathcal{SE}}^{\textit{ot-qcca2}}(\lambda)$.

Putting everything together, by the security of the underlying building blocks, we have the security of $\mathcal{E}'$. □