Post-quantum Security of the CBC, CFB, OFB, CTR, and XTS Modes of Operation

Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi,
Gelo Noel Tabia, and Dominique Unruh

University of Tartu
Estonia

Abstract. We examine the IND-qCPA security of the wide-spread block cipher modes of operation CBC, CFB. OFB, CTR, and XTS (i.e., security against quantum adversaries doing queries in superposition). We show that OFB and CTR are secure assuming that the underlying block cipher is a standard secure PRF (a pseudorandom function secure under classical queries). We give counterexamples that show that CBC, CFB, and XTS are not secure under the same assumption. And we give proofs that CBC and CFB mode are secure if we assume a quantum secure PRF (secure under queries in superposition).

Keywords. Post-quantum cryptography. Block ciphers. Modes of operation. IND-qCPA security.

1 Introduction

Block ciphers are one of the most fundamental primitives in cryptography. On its own, however, a block cipher is almost useless because it can only encrypt messages of a fixed (and usually very short) length. Therefore block ciphers are usually used in so-called "modes of operation": constructions whose goal it is to extend the message space of the block cipher, and possibly add other features or more security in the process. Since most encryption in practice uses at some level a mode of operation, the security of those modes of operation is of paramount importance for the security of many cryptographic systems.

In the light of the possible advent of quantum computers,[1] we have to ask: is existing classical cryptography also secure in the presence of attackers with quantum computers? In particular, does the security of common modes of operation break down?

In this paper, we study a number of common modes of operation, namely those listed in the 2013 ENISA[2] report on recommended encryption algorithms [10]: CBC, CFB, OFB, CTR, and XTS. We study whether those modes are secure in the quantum setting under comparable assumptions as in the classical setting, and if not, we construct counterexamples.

The aforementioned modes of operation (except ECB and XTS) are known to be IND-CPA secure in the classical setting, under the assumption that the underlying block cipher is a pseudo-random function (PRF).[3] ECB is known not to have reasonable security for most applications, while the security of XTS is an open question.

In the quantum case, there are two variants of the IND-CPA notion: "standard IND-CPA" and "IND-qCPA". While standard IND-CPA lets the quantum adversary perform only classical encryption queries, IND-qCPA (as defined by [6]) allows the adversary to perform quantum encryption queries (i.e., queries which are a superposition of different messages, to get a superposition of different ciphertexts). In other words, IND-qCPA additionally guarantees security when the encryption key is used to encrypt messages in superposition. (See below for a discussion on the relevance of this notion.)

[1] There seem to be no clear predictions as to when quantum computers will be available and strong enough to attack cryptography. But it seems daring to simply assume that they will not be available in the mid-term future, just because we do not have clear predictions.

[2] European Union Agency for Network and Information Security. We chose this list as a basis in order to investigate a practically relevant and industrially deployed set of modes of operations.

[3] If we want to be able to decrypt, then the block cipher should, of course, be a pseudo-random *permutation*. But for mere security. PRF is sufficient.

Mode of operation	Classical IND-CPA?	Standard (quantum) IND-CPA?	IND-qCPA? (with PRF)	(with qPRF)
ECB	no	no	no	no
CBC	yes[17]	yes	no[2]	yes[4]
CFB	yes[17]	yes	no[3]	yes[4]
OFB	yes[17]	yes	yes[3]	yes[3]
CTR	yes[17]	yes	yes[3]	yes[3]
XTS	unknown[11]	unknown	"no in spirit"[4]	unknown

Table 1: Summary of our results. The superscripts refer to the bibliography or to theorem numbers. "No in spirit" means that there is an attack using superposition queries that does not formally violate IND-qCPA.

Similarly, there are two variants of the notion of a classical PRF in the quantum setting: standard secure PRF and quantum secure PRF. In the first case, the function cannot be distinguished from a random function when making arbitrary classical queries to that function. In the second case, the function cannot be distinguished from random when making arbitrary quantum queries, i.e., when querying the function on a superposition of many inputs.

We can now ask the question: which variant of quantum PRFs is needed for which variant of IND-CPA. As it turns out, if we merely wish to get standard IND-CPA security, the answer is trivial: CBC, CFB, OFB, and CTR are secure assuming that the underlying block cipher is a standard PRF. In fact, the original security proofs of these schemes can be reused unmodified.[4] (We hence abstain from reproducing the original proofs in this paper and refer to the classical proofs instead.) And ECB is still trivially insecure, and for XTS we still do not know which security we achieve.

On the other hand, if we ask for IND-qCPA security, the picture changes drastically. OFB and CTR mode can be shown IND-qCPA secure based on a standard secure PRF. (The proof is relatively straightforward.)

In contrast, we prove that CBC and CFB are *not* IND-qCPA secure based when based on a standard secure PRF. In fact, for CBC and CFB we show that the adversary can even recover the secret key using quantum queries. For XTS, we show that the adversary can recover the second half of a plaintext if he can provide the first half of the plaintext (and the adversary can get half of the key). Although this does not formally contradict IND-qCPA (because IND-qCPA does not allow the challenge query to be performed in superposition), it show that XTS does not satisfy the intuitive notion of CPA security under superposition attacks.

If, however, the block cipher is a quantum secure PRF, then CBC and CFB are IND-qCPA secure. The proof of this fact, however, is quite different from the classical security proof: since the block cipher is invoked in superposition, we are in a situation similar to the analysis of quantum random oracles, which are notoriously difficult to handle in the quantum case. (Note: this refers only to the difficulties encountered in our proof. Our results are in the standard model, not in the random oracle model.)

We summarize the results in Table 1. Our counter-examples are in the quantum random oracle model, but our positive results are in the standard model (no random oracle).

On the IND-qCPA security notion. The IND-qCPA security notion [6] models passive security against adversaries that have access to the encryption of (chosen) plaintexts in superposition. The obvious question is: do we need that?

- The most obvious reason is that in the future, we might want to encrypt messages in superposition for some legitimate purpose. E.g., the encryption scheme is used as part of a

[4] Except that the set of adversaries we consider is, of course, that of quantum polynomial-time adversaries, instead of classical polynomial-time adversaries. Note that it is not always the case that a classical security proof goes through unchanged in the quantum case. (A typical example are zero-knowledge proof systems where rewinding is used in the classical proof. Rewinding-based proofs cannot be directly translated to the quantum setting [1, 13, 16].)

Fig. 1: (a) CBC mode (using a random function H instead of the block cipher). (b) Modified challenge ciphertext computation (c_1 replaced by randomness). We need to prove that replacing c_2 by a random value leads to an indistiguishable view.

quantum protocol. (That is, a protocol that actively uses quantum communication, not just a classical protocol secure against quantum adversaries.)
- A second argument (made in [7]) is that with continuing miniaturization, supposedly classical devices may enter the quantum scale, and thus "accidentally" encrypt messages in superposition. (Personally, we have doubts how realistic this case is, but we mention it for completeness.)
- There is, however, a reason why insecurity under notions such as IND-qCPA may affect the security of a purely classical system in the presence of a quantum attacker. If a classical protocol is proven secure (with respect to a quantum adversary), intermediate games in the security proof may actually contain honest parties that run in superposition. This happens in particular if zero-knowledge proof systems or similar are involved [13, 16]. For example, in [14, Section 5], the security proof of a classical protocol did not go through because the signature scheme was not secure under quantum queries (they had to change the protocol considerably instead). Encryption schemes that are not just standard IND-CPA, but IND-qCPA might help in similar situations.

1.1 Our techniques

We briefly summarize the techniques we use to prove or disprove the security of the various modes of operation.

IND-qCPA security of OFB and CTR mode using a standard PRF. Both OFB and CTR mode are stream ciphers. That is, in both cases, encryption can be represented as $\mathsf{Enc}_k(M) = G_k(|M|; r) \oplus M$, where G_k is a pseudorandom generator with key k for some randomness r. Thus, to encrypt a superposition $\sum_i \alpha_i |M_i\rangle$ of messages of length ℓ, all we need to do is to compute $c := \mathsf{Enc}_k(0) = G_k(\ell; r)$, and then to compute $\sum_i \alpha_i |\mathsf{Enc}_k(M_i; r)\rangle = \sum_i \alpha_i |M_i \oplus c\rangle$. Since computing $\mathsf{Enc}_k(0)$ can be done using a classical encryption query, it follows that superposition encryption queries can be simulated using classical encryption queries. Hence the IND-qCPA security of OFB and CTR can be directly reduced to the standard IND-CPA security of the same schemes. And standard IND-CPA security is shown exactly in the same way as in the classical setting.

IND-qCPA security of CBC and CFB mode using a quantum secure PRF. To show security of CBC and CFB mode, we cannot directly follow the classical security proof since that one relies inherently on the fact that the block cipher (the PRF) is queried only classically. Instead, we use the following techniques to prove CBC security:
- Since the block cipher is a PRF, we can assume it to be a truly random function H (to which the adversary has no access, since he does not know the key). CBC encryption is thus performed as sketched in Figure 1 (a).
- We replace the challenge encryption (i.e., the encryption query where the adversary should distinguish between $\mathsf{Enc}(m_0)$ and $\mathsf{Enc}(m_1)$) step by step by randomness. That is, we consider

a sequence of hybrid games, and in the i-th game, the first i blocks of the challenge ciphertext are replaced by uniformly random bitstrings. Once all ciphertext blocks are replaced by randomness, the probability of guessing whether m_0 or m_1 was encrypted is obviously $\frac{1}{2}$. Thus, all we need to show is that replacing one block of the challenge ciphertext by randomness leads to a negligible change in the advantage of the adversary. The situation is depicted in Figure 1 (b).

- Say we want to show that $c_2 = H(m_2 \oplus c_1)$ is indistinguishable from random (the situation in Figure 1 (b). At a first glance, this seems simple: $m_2 \oplus c_1$ is uniformly random, so the probability that it collides with other H-queries is negligible, hence $H(m_2 \oplus c_1)$ is uniformly random. However, this argument does not hold in the quantum setting: since some encryption queries are performed in superposition, it can be that H was queries on all inputs simultaneously, hence we cannot say that H was not queried at $m_2 \oplus c_1$ before. Fortunately, we can use the "One-way to Hiding (O2H) Lemma" from [15] here. This lemma basically says: for a uniformly random x, to show that $H(x)$ is indistinguishable from random, we need to show: when running the adversary, and aborting at a randomly chosen H-query, and measuring the input to that query (disturbing the superposition), then the probability that the outcome is x is negligible.
  In the present setting this means: if we measure a random H-query during the execution of the IND-qCPA game, the probability that the argument equals $m_2 \oplus c_1$ is negligible. For example, the probability that one of the h-queries before the challenge encryption equals $m_2 \oplus c_1$ is trivially negligible, because c_1 has not yet been chosen at that point.
- For the H-queries performed during the challenge query, we use the fact that H is indistinguishable from a random permutation [19]. In that case, the H-query inputs are uniformly random due to the fact that c_2 is chosen uniformly at random (remember that we replaced c_2 by a random value), hence they collide with $m_2 \oplus c_1$ only with negligible probability.
- For the H-queries performed after the challenge query, we cannot use the same argument, because those queries can be performed in superposition. However: if we only care whether the chosen H-query has input $m_2 \oplus c_1$, then, instead of just measuring the H-query input, we can measure in the computational basis all registers involved in the encryption. Then we observe that measuring all registers commutes with the operations performed during encryption, so equivalently we can assume that that measurement happens at the beginning of the encryption (and in particular measures the plaintext). And that means, for the purposes of bounding the probability of measuring H-query input $m_2 \oplus c_1$, we can assume that we encrypt a classical plaintext. From here, the argument from the previous item applies.
- Altogether, the probability of measuring $m_2 \oplus c_1$ in any H-query is negligible. Then the O2H lemma implies that the $H(m_2 \oplus c_1)$ is indistinguishable from random. And by iterating this indistinguishability, we can replace the whole challenge ciphertext by randomness. And then the adversary has only probability $\frac{1}{2}$ of guessing which challenge plaintext was encrypted.

This shows that CBC mode is IND-qCPA secure if the block cipher is a quantum secure PRF. The security of CFB mode is shown very similarly.

Insecurity of CBC and CFB mode using a standard secure PRF. To show that CBC and CFB mode are insecure using a standard secure PRF, we first construct a specific block cipher BC as follows:

$$\mathsf{BC}_k(x) := E_{H(k)}\big(\mathsf{droplastbit}\,(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))\big)$$

where E is a standard secure PRF and H refers to a random oracle. (This construction is not really a block cipher because it is not injective and hence not decryptable. The definition of BC_k can be refined to make it decryptable, we omit this technicality in this proof overview, see Section 3.1.) This block cipher has the special property of being $k\|1$-periodic: $\mathsf{BC}_k(x) = \mathsf{BC}_k(x \oplus (k\|1))$. In particular, this it cannot be a quantum secure PRF, even if E is. Namely, given superposition access to BC_k, Simon's algorithm [12] allows us to recover $k\|1$ given quantum

oracle access to BC_k.[5] This idea also allows us to break CBC mode when CBC mode uses BC_k as its underlying blockcipher. If we encrypt a single block message m using CBC, we get the ciphertext $(c_0, \mathsf{BC}_k(c_0 \oplus m))$. Although the message m is XORed with the random IV c_0, the period remains the same, namely $k\|1$. Thus, using what is basically Simon's algorithm, using superposition queries to CBC mode, we get $k\|1$ (more precisely, one bit of information about it for each superposition query). This reveals the key k completely and in particular shows that CBC is not IND-qCPA secure.

The question of course is whether BC_k is indeed a standard secure PRF. Even though the adversary has only classical access to BC_k, the proof cannot be purely classical: we use a random oracle H that the adversary can query in superposition. Instead, we use again the O2H lemma [15] mentioned above. This allows us to replace $H(k)$ by a random key y in the definition of BC_k. Now the analysis of BC_k becomes purely classical and basically amount to showing that the adversary cannot guess two inputs to BC_k that lead to the same input for E_y. (Using the actual, decryptable construction of BC_k, this proof becomes technically a bit more complex, but still follows the same ideas.)

In the case of CFB mode, the attack is similar, except that here we need to encrypt two-block messages in order to get a ciphertext that depends in a $k\|1$-periodic way on the plaintext. (Since the first message block is not fed through the block cipher in CFB mode.)

Insecurity of XTS mode using a standard secure PRF. To attack XTS, we use the same basic idea as for CBC and CFB. However, there are some additional complications. In XTS, two keys k_1, k_2 are used. Each ciphertext block is computed as $c_i := \alpha^{i-1}L \oplus \mathsf{BC}_{k_2}(\alpha^{i-1}L \oplus m_i)$. Here $L := \mathsf{BC}_{k_1}(I)$ is a secret value that is derived from a nonce I (thus L stays fixed throughout one encryption operation, but changes from ciphertext to ciphertext). If we use the block cipher constructed above (when breaking CBC), we can easily derive k_2: since BC_{k_2} is k_2-periodic, so is $\mathsf{BC}_{k_2}(\alpha^{i-1}L \oplus m_i)$. Thus with one single block encryption we would be able to retrieve one bit of k_2 using Simon's algorithm. However, retrieving k_2 does not help us in decrypting XTS mode, since we do not know k_1, and hence cannot compute the value L. Also, the fact that $\mathsf{BC}_{k_1}(I)$ is k_1-periodic does not help us to retrieve k_1 since we do not have any control over I. Instead, we use the following trick. We construct

$$\mathsf{BC}_k(x, y) := E_{H(k)}\big(\mathsf{droplastbit}\,(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))\,,\ \mathsf{droplastbit}\,(y \oplus f_k(x) \cdot \mathsf{lastbit}(x))\big)$$

where f_k is a suitable function depending on k (with the property that $\mathsf{lastbit}(\ f_k(\cdot)) = 1$). (We interpret message blocks are pairs x, y by splitting them in the middle.) Again we ignore in this proof overview that BC_k cannot be decrypted, the more involved construction given in Section 3.2 avoids this problem.

Now BC_k is k-periodic in x, and $f_k(x)$-periodic in y for fixed first input x. Using this block cipher, we can first use the attack technique described for CBC mode to recover k_2 (by encrypting a number of one block messages). The main difference is that now we create a plaintext that is a superposition in the first half of the block (x). and fixes the second block ($y := 0$). Now, instead of recovering k_1 (which seems impossible), we can recover the message L used during a given encryption query: We encrypt a message where the x-part of each block is 0, and the y-part of each block is the superposition of all messages. Since BC_{k_2} is invoked with $\alpha^{i-1}L \oplus m_i$ when encrypting m_i, we have that the first half of the input to BC_{k_2} is the first half of $\alpha^{i-1}L$. Thus BC_{k_2} is $f_{k_2}(\mathit{firsthalf}(\alpha^{i-1}L))$-periodic. Thus from message block i, using Simon's algorithm, we get one bit of $f_{k_2}(\mathit{firsthalf}(\alpha^{i-1}L))$. Since we know k_2, this reveals one bit of information about $\alpha^{i-1}L$. Thus we get a bit each about many different $\alpha^{i-1}L$ (for different i), and this allows us to compute L. If our ciphertext, in addition to the superposition-message-blocks contains parts

[5] A similar idea was already used in [18] to show that there is a standard secure PRF that is not quantum secure. However, their construction had a period with respect to $+$, not to $\oplus$, which makes it unsuitable for showing the insecurity of CBC mode.

that are unknown, we can then decrypt those using our knowledge of L and k_2. (Note that we cannot use this knowledge to decrypt another ciphertext, since each ciphertext uses a different L.) Thus, we can decrypt ciphertexts whose plaintexts are partially under our control (and in superposition), and partially unknown.

1.2 Related work

Boneh *et. al.* [4] have argued the requirement of quantum-accessible random oracle model to prove post-quantum of BR encryption scheme introduced in [2]. They have proved the CCA security of hybrid encryption scheme introduced in [2] in the quantum random oracle model. Ebrahimi and Unruh in [9] prove the CCA security of Fujisaki-Okamoto transform in the quantum random oracle model. In [5] Boneh and Zhandry construct the first message authentication codes (MACs) that are existentially unforgeable against a quantum chosen message attack and show that quantum-secure PRF leads to quantum-secure MACs. In [8], Damgård *et. al.* study secret sharing scheme and multiparty computation where the adversary make ask superposition queries. They also examine the zero knowledge protocols and use the secret sharing results to design zero knowledge proofs for all of NP in the common reference string model.

1.3 Organisation

In Section 2 we provide the various security definitions and lemmas used throughout the paper. Section 2.1 contains the definition of all the modes of operations discussed. Section 3 describes the attack on CBC, CFB, and XTS mode of operation based on standard secure PRF. In Section 4 we show how to achieve the IND-qCPA security for the OFB and CTR modes of operation. In Section 5 we show how to achieve the IND-qCPA security for the CBC and CFB modes of operation.

2 Notation and tools

Notation. By $x \leftarrow A(y)$ we denote an algorithm A that takes an input y outputs a value that is assigned to x. We write $x \leftarrow A^H(y)$ if A has access to an oracle H. By $(A \leftarrow B)$ we refer to the set of all functions from A to B. $x \xleftarrow{\$} A$ represents an x which is uniformly randomly chosen from the set A. $\{0,1\}^n$ represents the bit-strings of length n and $a\|b$ for strings a and b represents the concatenation of two strings. For two vectors a and b, $a \odot b$ denotes the dot product between two vectors. We use $\eta(t)$ to denote a function with a security parameter t. If we say a quantity is *negligible*(denoted *negl.*) we mean that it is in $o(\eta^c)$ or $1 - o(\eta^c)$ for all $c > 0$. We use the notation $A \approx B$ to say that quantity A has *negl.* difference with quantity B. For an $n-$bit string a and binary variable b, $a \cdot b = a$ if $b = 1$ otherwise $a \cdot b = 0^n$. For a string $x = x_1x_2x_3 \cdots x_n$ where x_i is the $i-th$ bit we use functions $\mathsf{lastbit}$ and $\mathsf{droplastbit}$ such that $\mathsf{lastbit}(x) = x_n$ and $\mathsf{droplastbit}(x) = x_ix_2 \cdots x_{n-1}$.

Definition 1 (IND-CPA). *A symmetric encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is indistinguishable under chosen message attack (IND-CPA secure) if no classical poly-time adversary $\mathcal{A}$ can win in the $PrivK^{CPA}_{\mathcal{A},\Pi}(t)$ game, except with probability at most 1/2 + negl:*

$PrivK^{CPA}_{\mathcal{A},\Pi}(t)$ game:

Key Gen: *The challenger picks a random key $k \leftarrow \mathsf{Gen}$ and a random bit b.*

Query: *Adversary $\mathcal{A}$ chooses two messages m_0, m_1 and sends them to the challenger. Challenger chooses $r \xleftarrow{\$} \{0,1\}^*$ and responds with $c^* = \mathsf{Enc}_k(m_b; r)$.*

Guess: *Adversary $\mathcal{A}$ produces a bit b', and wins if $b = b'$.*

Definition 2 (IND-qCPA [6]). *A symmetric encryption scheme Π = (Gen,Enc, Dec) is indistinguishable under quantum chosen message attack (IND-qCPA secure) if no efficient adversary $\mathcal{A}$ can win in the $PrivK^{qCPA}_{\mathcal{A},\Pi}(t)$ game, except with probability at most 1/2 + negl:*

$PrivK^{qCPA}_{\mathcal{A},\Pi}(t)$ game:

Key Gen: *The challenger picks a random key k and a random bit b.*

Queries

*- **Challenge Queries:** $\mathcal{A}$ sends two messages m_0, m_1 to which the challenger responds with $c^* = \mathsf{Enc}_k(m_b; r)$.*

*- **Encryption Queries:** For each such query, the challenger chooses randomness r, and encrypts each message in the superposition using r as randomness:*

$$\sum_{m,c} \psi_{m,c}|m,c\rangle \rightarrow \sum_{m,c} \psi_{m,c}\big|m, c \oplus Enc_k(m;r)\big\rangle$$

Guess: *$\mathcal{A}$ produces a bit b', and wins if $b = b'$.*

Definition 3 (Standard-security [18]). *A function PRF is a standard-secure PRF if no efficient quantum adversary $\mathcal{A}$ making classical queries can distinguish between a truly random function and a function PRF_k for a random k. That is, for every such $\mathcal{A}$, there exists a negligible function $\epsilon = \epsilon(t)$ such that*

$$\big|\Pr_{k\leftarrow\mathcal{K}}[A^{\mathsf{PRF_k}}() = 1] - \Pr_{O\leftarrow\mathcal{Y}^{\mathcal{X}}}[A^{O}() = 1]\big| < \epsilon.$$

Definition 4 (Quantum-security [18]). *A function PRF is a quantum secure PRF if no poly-time quantum adversary $\mathcal{A}$ making quantum queries can distinguish between truly random function and the function PRF_k for a random k.*

Lemma 1 (One way to hiding (O2H) [15]). *Let $H : \{0,1\}^t \rightarrow \{0,1\}^t$ be a random oracle. Consider an oracle algorithm A_{O2H} that makes at most q_{o2h} queries to H. Let B be an oracle algorithm that on input x does the following: pick $i \xleftarrow{\$} \{1,\ldots,q_{o2h}\}$ and $y \xleftarrow{\$} \{0,1\}^t$, run $A^H_{O2H}(x,y)$ until (just before) the $i-th$ query, measure the argument of the query in the computational basis, output the measurement outcome. (When A_{O2H} makes less than i queries, B outputs $\perp \notin \{0,1\}^t$.) Let,*

$$P^1_{A_{O2H}} := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^t \rightarrow \{0,1\}^t), x \xleftarrow{\$} \{0,1\}^t, b' \leftarrow A^H_{O2H}(x, H(x))],$$

$$P^2_{A_{O2H}} := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^t \rightarrow \{0,1\}^t), x \xleftarrow{\$} \{0,1\}^t, y \xleftarrow{\$} \{0,1\}^t, b' \leftarrow A^H_{O2H}(x,y)],$$

$$P_B := \Pr[x' = x : H \xleftarrow{\$} (\{0,1\}^t \rightarrow \{0,1\}^t), x \xleftarrow{\$} \{0,1\}^t, x' \leftarrow B^H(x,i)].$$

Then,

$$\big|P^1_{A_{O2H}} - P^2_{A_{O2H}}\big| \leq 2q_{o2h}\sqrt{P_B}.$$

2.1 Modes of operation

Definition 5 (ECB Scheme). *For a given permutation $E : \mathcal{K} \times \{0,1\}^t \rightarrow \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{ECB} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*

Gen*: Pick a random key $k \xleftarrow{\$} \mathcal{K}$.*

Enc*: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_1 \cdots c_n$, where $c_i = E(k, m_i)$ for $0 < i \leq n$.*

Dec*: For a given cipher-text $C = c_1 \cdots c_n$ and key k; $\hat{m}_i := E^{-1}(k, c_i)$ for $0 < i \leq n$.*

Definition 6 (CBC Scheme). *For a given permutation $E : \mathcal{K} \times \{0,1\}^t \to \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{CBC} = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ as follows:*
Gen: Pick a random key $k \xleftarrow{\$} \mathcal{K}$.
Enc: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_0 c_1 \cdots c_n$, where $c_0 \xleftarrow{\$} \{0,1\}^t$ and $c_i = E(k, m_i \oplus c_{i-1})$ for $0 < i \leq n$.
Dec: For a given cipher-text $C = c_0 c_1 \cdots c_n$ and key k; $\hat{m}_i := E^{-1}(k, c_i) \oplus c_{i-1}$ for $0 < i \leq n$.

Definition 7 (CFB Scheme). *For a given function $E : \mathcal{K} \times \{0,1\}^t \to \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{CFB} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*
Gen*: Pick a random key $k \xleftarrow{\$} \mathcal{K}$.*
Enc*: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_0 c_1 \cdots c_n$, where $c_0 \xleftarrow{\$} \{0,1\}^t$ and $c_i = E(k, c_{i-1}) \oplus m_i$ for $0 < i \leq n$.*
Dec*: For a given cipher-text $C = c_0 c_1 \cdots c_n$ and key k; $\hat{m}_i := E(k, c_{i-1}) \oplus c_i$ for $0 < i \leq n$.*

Definition 8 (OFB Scheme). *For a given function $E : \mathcal{K} \times \{0,1\}^t \to \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{OFB} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*
Gen*: Pick a random key $k \xleftarrow{\$} \mathcal{K}$.*
Enc*: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_0 c_1 \cdots c_n$, where $c_0 = r_0 \xleftarrow{\$} \{0,1\}^t$, $r_i = E(k, r_{i-1})$ and $c_i = r_i \oplus m_i$ for $0 < i \leq n$.*
Dec*: For a given cipher-text $C = c_0 c_1 \cdots c_n$ and key k; $\hat{m}_i := E(k, c_{i-1}) \oplus c_i$ for $0 < i \leq n$.*

Definition 9 (CTR Scheme). *For a given function $E : \mathcal{K} \times \{0,1\}^t \to \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{CTR} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*
Gen*: Pick a random key $k \xleftarrow{\$} \mathcal{K}$.*
Enc*: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_0 c_1 \cdots c_n$, where $c_0 \xleftarrow{\$} \{0,1\}^t$ and $c_i = E(k, c_0 + i) \oplus m_i$ for $0 < i \leq n$.*
Dec*: For a given cipher-text $C = c_0 c_1 \cdots c_n$ and key k; $\hat{m}_i := E(k, c_0 + i) \oplus c_i$ for $0 < i \leq n$.*

Definition 10 (XTS Scheme). *For a given permutation $E : \mathcal{K} \times \{0,1\}^t \to \{0,1\}^t$ we define the symmetric encryption scheme $\Pi_{XTS} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*
Gen*: Pick random keys k_1 and k_2 i.e., $k_1 \xleftarrow{\$} \mathcal{K}$ and $k_2 \xleftarrow{\$} \mathcal{K}$.*
Enc*: For a given message $M = m_1 m_2 \cdots m_n$, where n is a polynomial in t; $\mathsf{Enc}_k(M) := c_0 c_1 \cdots c_n$, where $c_i = E(k_1, m_i \oplus \Delta_i) \oplus \Delta_i$ for $0 < i \leq n$, $\Delta = \alpha^{i-1} L$, $L = E(k_2, I)$ and α is the primitive element of the field $\mathbb{F}_2^n$. Here I is a publicly known nonce that is agreed upon out of band (but that is different in different ciphertexts).*
Dec*: For a given cipher-text $C = c_1 \cdots c_n$; and key k; $\hat{m}_i := E(k, c_i \oplus \Delta_i) \oplus \Delta_i$ for $0 < i \leq n$.*

3 Quantum attacks on CBC, CFB, and XTS based on standard secure PRF

We show that CBC and CFB mode are not IND-qCPA secure in general when the underlying block cipher is only a standard secure PRF, and that XTS has a chosen-plaintext attack using superposition queries. For this, in Section 3.1 and Section 3.2 we first construct two different block ciphers that are standard secure PRFs (but are intentionally not quantum secure). Then, in Section 3.3 and **??** we show how to break CBC and CFB, respectively, when using the first of those block ciphers. And in Section 3.5 we show how to break XTS when using the second block cipher.

3.1 Construction of the block cipher for CBC

To show that a standard secure PRF is not sufficient for IND-qCPA security of CBC and XTS modes of operation we need a block cipher that is standard secure PRF but not quantum secure.

Our first step is to construct such a block cipher and prove it to be standard secure. In this section we provide two such constructions of block cipher that would be later used to show insecurity of CBC and XTS against a quantum adversary respectively.

Construction 1:

$$\mathsf{BC}_k(x) = E_{H(k)_1}\big(\mathsf{droplastbit}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))\big) \Big\| t_{H(k)_2}\big(x \oplus (k\|1) \cdot \mathsf{lastbit}(x)\big) \oplus \mathsf{lastbit}(x),$$

where, $E : \{0,1\}^{n-1} \times \{0,1\}^{n-1} \to \{0,1\}^{n-1}$ is a standard secure PRF , $t : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is a standard secure PRF, $H : \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ is a random oracle and the key $k \xleftarrow{\$} \{0,1\}^{n-1}$.

For the chosen block cipher BC we show that the given construction is a permutation. It is easy to see that E and t take an $n-1$-bit string (i.e., $\mathsf{droplastbit}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))$) and $n-$bit string (i.e.,$(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))$) respectively [6]. By appending $1-$bit output from t XORed with the last-bit of x it makes BC_k's output an n-bit string. For a given output of BC_k and key k one can decrypt E using key $H(k)_1$ to retrieve the input $\mathsf{droplastbit}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))$. This can be appended a 0-bit and can be fed into t with key $H(k)_2$ to get an output of 1-bit. This $1-$bit output when XORed with the $\mathsf{lastbit}(\mathsf{BC}_k)$ gives the $\mathsf{lastbit}(x)$. From $x \oplus (k\|1) \cdot \mathsf{lastbit}(x)$ and $\mathsf{lastbit}(x)$, we can compute x. Thus, BC is injective and if E is efficiently invertible, so is BC.

Theorem 1. *Construction 1 is a standard secure PRF for any quantum adversary $\mathcal{D}$ given classical access to BC_k and quantum access to the random oracle H.*

Proof. Let $\mathcal{D}$ be any quantum adversary that distinguishes the BC_k from a truly random function R_F with advantage δ given classical access to BC_k and quantum access to H. Without loss of generality we assume that $\mathcal{D}$ never queries its classical oracle twice with the same input. Therefore,

$$\delta = \Big|\Pr\Big[\mathcal{D}^{\underline{\mathsf{BC}_k},H} = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}, H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{2n})\Big] - \Pr\Big[\mathcal{D}^{\underline{R_F},H} = 1 : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{2n}), R_F \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^n)\Big]\Big|,$$

where, $\underline{\mathsf{BC}_k}$ denotes classical access to function BC_k.

Let,

$$G_0 := \Pr\left[\mathcal{D}^{\underline{\mathsf{BC}_k},H} = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}, H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{2n})\right],$$

$$G_1 := \Pr\left[\mathcal{D}^{\underline{R_F},H} = 1 : R_F \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^n), H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{2n})\right].$$

Thus, using the above definition we have: $\delta = |G_0 - G_1|$. The construction 1 uses a random oracle's output as a key. By the O2H lemma (Lemma 1) we have the upper bound on the probability of distinguishing between output of H and a random oracle on a common input. Hence, we define the adversaries for O2H lemma as below:

Adversary $A^H(k,w)$:

$b \leftarrow \mathcal{D}^{\underline{\tilde{\mathsf{BC}}'^k_w},H}$

return b

Adversary $B^H(k,j)$:

until the j-th H-query

$b \leftarrow \mathcal{D}^{\underline{\tilde{\mathsf{BC}}'^k_w},H}$

if $\mathcal{D}$ makes $< j$ queries

return $\perp$

$m =$ measured arg. of H

return m

[6] Notice that $(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))$ has its last bit always 0.

where,

$$\tilde{\mathsf{BC}}'^{k}_{w}(x) := E_{w_1}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x)) \quad \| \quad t_{w_2}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x)) \oplus \mathsf{lastbit}(x), \text{ and } w = w_1 \| w_2.$$

Using the above definition of the adversaries we have the following probabilities:

$$P_A^1 := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right), b' \leftarrow A^H(k, H(k))\right],$$

$$P_A^2 := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right), b' \leftarrow A^H(k, w)\right],$$

$$P_B := \Pr\left[k = k' : i \xleftarrow{\$} \{1, \ldots, q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B^H(k, i)\right]$$

where, q is the polynomial upper bound on the number of $H-$queries performed by $\mathcal{D}$.

Hence, by O2H lemma [15] (Lemma 1) we have that, $|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$. It is clear that $G_0 = P_A^1$. We now have to prove that P_B is *negligible*. Hence, we define adversaries A_2 and B_2 similar to adversary A and B except that it uses random oracles $\tilde{E}$ and $\tilde{t}$ in construction 1.

Adversary $A_2^H(k, w)$:
choose $\tilde{E}$,$\tilde{t}$ and simulate BC'_k queries using $\tilde{E}, \tilde{t}$

$b \leftarrow \mathcal{D}^{\underline{\mathsf{BC}'_k}, H}$

return b

Adversary $B_2^H(k, j)$:
choose $\tilde{E}$,$\tilde{t}$ and simulate BC'_k queries using $\tilde{E}, \tilde{t}$
until j-th $H-$ query:
$b \leftarrow \mathcal{D}^{\underline{\mathsf{BC}'_k}, H}$
if $\mathcal{D}$ makes $< j$ queries
return $\perp$
m = measured arg. of H
return m

where, $\mathsf{BC}'_k(x) := \tilde{E}(\mathsf{droplastbit}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x))) \quad \| \quad \tilde{t}(x \oplus (k\|1) \cdot \mathsf{lastbit}(x)) \oplus \mathsf{lastbit}(x)$ and $\tilde{E} : \{0,1\}^{n-1} \to \{0,1\}^{n-1}$, $\tilde{t} : \{0,1\}^n \to \{0,1\}$ are chosen randomly. We define,

$$G_2 := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right), b' \leftarrow A_2^H(k, w)\right]$$

$$P_{B_2} := \Pr\left[k = k' : j \xleftarrow{\$} \{1, \ldots, q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_2^H(k, j)\right]$$

Therefore, $P_B \leq P_{B_2} + \epsilon$, for some *negligible* ϵ as the only difference between the adversary B and B_2 is that B^H uses classical access to PRFs E_{w_1} and t_{w_2} where w_1 and w_2 are randomly chosen while B_2 uses random oracles $\tilde{E}$ and $\tilde{t}$ respectively. We define the following classical oracles $\tilde{E_1}, \tilde{t_1}, \tilde{E_2}, \tilde{t_2}$ as below:

Sampling 1:

Oracle $\tilde{E_1}$:
Upon query on x:
$\tilde{E_1}(x) = \begin{cases} \text{previous stored value}, & x \text{ queried before} \\ \text{store random element}, & x \text{ not queried before} \end{cases}$
return $\tilde{E_1}(x)$

Oracle $\tilde{t_1}$:
Upon query on x:
$\tilde{t_1}(x) = \begin{cases} \text{previous stored value}, & x \text{ queried before} \\ \text{store random element}, & x \text{ not queried before} \end{cases}$
return $\tilde{t_1}(x)$

Sampling 2:

Oracle $\tilde{E_2}$:
Upon query on x:
$\tilde{E_2}(x)$ = choose at random
if (x queried before in $\tilde{E_2}$):
bad event
return $\tilde{E_2}(x)$

Oracle $\tilde{t_2}$:
Upon query on x:
$\tilde{t_2}(x)$ = choose at random
if (x queried before in $\tilde{t_2}$):
bad event
return $\tilde{t_2}(x)$

It is easy to see that oracles $\tilde{E}_1$ and $\tilde{t}_1$ perfectly simulate oracles $\tilde{E}$ and $\tilde{t}$ respectively, whereas oracles $\tilde{E}_2$ and $\tilde{t}_2$ outputs a random string. Similar to adversary B_2 we define the adversary B_3 except that it uses oracles $\tilde{E}_2$ and $\tilde{t}_2$ instead of the oracles $\tilde{E}$ and $\tilde{t}$. Hence, we have:

$$P_{B_3} := \Pr\left[k = k' : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_3^H(k,j)\right].$$

The difference between sampling 1 and sampling 2 is only when one of the input is repeated, this we call the bad event. Let,

$$G_3 := \Pr[\text{bad event} : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_3^H(k,j)]$$

Hence, by fundamental lemma of game playing [3], we have that: $P_{B_2} \leq G_3 + P_{B_3}$.

In game G_3 and P_{B_3}, k is independent of k' because in sampling 2 oracles $\tilde{E}_2$ and $\tilde{t}_2$ gives outputs that are independent of their inputs and k is only used in the inputs of $\tilde{E}_2$ and $\tilde{t}_2$. Hence, we can replace string k in the input of adversary B_3 with a null string 0.

$$P_{B_3} = \Pr\left[k = k' : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_3^H(0,j)\right] \leq 2^{-n}.$$

Now we focus on the calculation of G_3 which denotes the probability of occurrence of bad event. From the definition of G_3 we see that bad event occurs only when same input is queried again. Let r be the total number of BC-queries and $q_1, \cdots, q_r$ are the queries to BC. Consider two queries $q_{i'}$ and $q_{j'}$ on BC'_k that leads to same query on oracles $\tilde{E}$ and $\tilde{t}$ then we have,

$$G_3 = \Pr[\exists i' \neq j' s.t. q_{i'} \oplus (k\|1) \cdot \mathsf{lastbit}(q_{i'}) = q_{j'} \oplus (k\|1) \cdot \mathsf{lastbit}(q_{j'}) : j \xleftarrow{\$} \{1,\ldots,q\}; (k\|1) \xleftarrow{\$} (\{0,1\}^{n-1}\|1); H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_3^H(0,j)].$$

From game G_3 we have $q_{i'} \oplus (k\|1) \cdot \mathsf{lastbit}(q_{i'}) = q_{j'} \oplus (k\|1) \cdot \mathsf{lastbit}(q_{j'})$. Hence, $q_{i'} \oplus q_{j'} = (k\|1) \cdot (\mathsf{lastbit}(q_{i'}) \oplus \mathsf{lastbit}(q_{j'}))$. Queries not being same must have different last bits, thereby XOR of last bit of $q_{i'}$ and $q_{j'}$ is 1 . Therefore we have,

$$G_3 = \Pr[\exists i' \neq j'; q_{i'} \oplus q_{j'} = (k\|1) : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); k' \leftarrow B_3^H(0,j)]$$

Since k does not occur in the r.h.s. of this probability, and there are $\leq \frac{r(r-1)}{2}$ pairs $i' \neq j'$

$$G_3 \leq \frac{r(r-1)}{2} 2^{-n+1}, \text{ which is } \textit{negligible}$$

We now have,

$$P_{B_2} \leq P_{B_3} + G_3 \leq 2^{-n} + \frac{r(r-1)}{2} 2^{-n+1} \approx negl.$$

Therefore,

$$P_B \leq 2^{-n} + \frac{r(r-1)}{2} 2^{-n+1} + \epsilon \approx negl.$$

Hence, $|G_0 - P_A^2| = |P_A^1 - P_A^2|$ is $negl$. Thus $\delta \leq |P_A^2 - G_1| + negl.$.

The only difference between BC'_k and $\tilde{\mathsf{BC}}'^k_w$ is that the underlying functions $\tilde{E}$ and $\tilde{t}$ of BC'_k are random oracles whereas those of $\tilde{\mathsf{BC}}^k_w$ are standard secure PRFs. Thus, by the definition of PRF $|P_A^2 - G_2| \leq \epsilon$. Using the sampling arguments as previously we define adversary A_3 similar to adversary A_2 except that the underlying function of BC'_k is $\tilde{E}_2$ and $\tilde{t}_2$.

$$G_4 = \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); b' \leftarrow A_3^H(k,w)\right],$$
$$G_5 = \Pr\left[\text{bad event} : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); b' \leftarrow A_3^H(k,w)\right]$$

Using the fundamental theorem of game playing [3] we have that $|G_2 - G_4| \leq G_5$. Output of A_3 in G_4 and G_5 does not depends on k as the oracles $\tilde{E}_2$ and $\tilde{t}_2$ output random strings. Hence, we can replace the input string k of A_3 with a null string 0.

$$G_4 = \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{2n}\right); b' \leftarrow A_3^H(0,w)\right],$$
$$G_5 = \Pr\left[\text{bad event} : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{2n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{n}\right); b' \leftarrow A_3^H(0,w)\right].$$

We have that $G_5 \leq \frac{r(r-1)}{2}2^{-n+1}$ which is *negl.* analogously to $G_3 \leq \frac{r(r-1)}{2}2^{-n+1}$. Hence, $|G_2 - G_4| \approx negl.$. Now we can see that adversary A_3 in game G_4 has completely random function instead of BC'_k as it uses random functions $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t}'_2$. Note that the function in G_4 gives different values upon queries with the same input while R_F in G_1 gives equal outputs in that case. Hence, $G_1 = G_4$. Therefore, using above results we have that:

$$\delta = |G_0 - G_1| \approx |P_A^2 - G_1| \approx |G_2 - G_1| \approx |G_4 - G_1| = negl$$

Thus, we have proved that the given construction is pseudo-random and hence a standard secure PRF.

Thus, we have proved that the given construction is pseudo-random and hence a standard secure PRF.

3.2 Construction of block cipher for XTS

Construction 2:

$$\mathsf{BC}_k(x,y) = E_{H(k)_1}(\mathsf{droplastbit}(\bar{x}), \mathsf{droplastbit}(\bar{y})) \quad \| \quad t_{H(k)_2}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(x) \quad \| \quad t'_{H(k)_3}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(y),$$

where $\bar{x} := x \oplus (k\|1) \cdot \mathsf{lastbit}(x)$ and $\bar{y} := y \oplus f_k(x) \cdot \mathsf{lastbit}(y)$ and $f_k(x) := x \oplus (0^{n-1}\|\mathsf{lastbit}(x)) \oplus (k\|1)$ and key $k \xleftarrow{\$} \{0,1\}^{n-1}$. $E : \{0,1\}^n \times \{0,1\}^{2n-2} \to \{0,1\}^{2n-2}$ and $t, t' : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}$ are standard secure PRFs. $H : \{0,1\}^{n-1} \to \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$ is a random oracle and x and y are n-bit strings.

For a chosen block cipher BC we first show that the given construction is a permutation. It is easy to see that E and t, t' take a $2n-2$-bit string (i.e, $(\mathsf{droplastbit}(\bar{x})\|\mathsf{droplastbit}(\bar{y}))$) and $2n-$bit string (i.e., $(\bar{x}\|\bar{y})$) respectively[7]. By appending 1-bit output each from t and t' XORed with the last-bit of x and last-bit of y respectively, it makes BC_k's output a $2n$-bit string. For a given output of BC_k and key k one can decrypt E with key $H(k)_1$ to retrieve the input $(\mathsf{droplastbit}(\bar{x})\|\mathsf{droplastbit}(\bar{y}))$. This can be appended a $0-$bit in each half be fed into t and t' with key $H(k)_2$ and $H(k)_3$ respectively, to get an output of $1-$bit each. These $1-$bit outputs when XORed with the last two bits of BC_k gives the $\mathsf{lastbit}(x)$ and $\mathsf{lastbit}(y)$. From $(x \oplus (k\|1) \cdot \mathsf{lastbit}(x)\|y \oplus f_k(x) \cdot \mathsf{lastbit}(y))$, $\mathsf{lastbit}(x)$ and $\mathsf{lastbit}(y)$ we can compute x and y. Thus, BC is injective, and if E is efficiently invertible, so is BC.

[7] Notice that the last bit of $(\bar{x}\|\bar{y})$ is always 0 in each half.

Theorem 2. *Construction 2 is a standard secure PRF for any quantum adversary $\mathcal{D}$ given classical oracle access BC_k and quantum access to the random oracle H.*

Proof. Let $\mathcal{D}$ be any quantum adversary that distinguishes BC_k from a truly random function R_F with advantage δ given classical access to BC_k and quantum access to H. Without loss of generality we assume that $\mathcal{D}$ never queries its classical oracle twice with the same input. Therefore,

$$\delta = \Big|\Pr\left[\mathcal{D}^{\underline{\mathsf{BC}_k},H} = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}, H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{3n})\right] - \Pr\left[\mathcal{D}^{\underline{R_F},H} = 1 : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{3n}), R_F \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^n)\right]\Big|,$$

where, $\underline{\mathsf{BC}_k}$ denotes classical access to function BC_k.

Using the following games G_0 and G_1;

$$G_0 := \Pr\left[\mathcal{D}^{\underline{\mathsf{BC}_k},H} = 1 : k \xleftarrow{\$} (\{0,1\}^{n-1}\|1), H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{3n})\right],$$
$$G_1 := \Pr\left[\mathcal{D}^{\underline{R_F},H} = 1 : H \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^{3n}), R_F \xleftarrow{\$} (\{0,1\}^n \to \{0,1\}^n)\right].$$

we have that: $\delta = \big|G_0 - G_1\big|$.

The construction 2 uses a random oracle's output as a key. By the O2H lemma (Lemma 1) we have an upper bound on the probability of distinguishing between output of H and a random oracle on a common input. Hence, we define the adversaries for O2H lemma as below:

Adversary $A^H(k,w)$:

$b \leftarrow \mathcal{D}^{\underline{\tilde{\mathsf{BC}}^k_w},H}$

return b

Adversary $B^H(k,j)$:
until the j-th H-query:
    $b \leftarrow \mathcal{D}^{\underline{\tilde{\mathsf{BC}}^k_w},H}$
if $\mathcal{D}$ makes $< j$ queries
    return $\perp$
m = measured arg. of H
return m

where,

$$\tilde{\mathsf{BC}}^k_w(x,y) := E_{w_1}(\bar{x},\bar{y}) \quad \| \quad t_{w_2}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(x) \quad \| \quad t'_{w_3}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(y), \text{ and } w = w_1\|w_2\|w_3.$$

Using the above definition of the adversaries we have the following probabilities:

$$P^1_A := \Pr\left[b' = 1 : k \xrightarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A^H(k, H(k\|1))\right],$$
$$P^2_A := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^n; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A^H(k, w)\right],$$
$$P_B := \Pr\left[k = k' : i \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B^H(k,i)\right],$$

where q is the polynomial upper bound on the number of $H-$queries performed by $\mathcal{D}$.

Hence, by O2H lemma [15] (Lemma 1) we have that, $|P^1_A - P^2_A| \leq 2q\sqrt{P_B}$. It is clear that $G_0 = P^1_A$. We now have to prove that P_B is *negligible*. Hence, we define a adversaries A_2 and B_2 similar to adversaries A and B respectively, except that it uses random oracles $\tilde{E}$, $\tilde{t}$, and $\tilde{t}'$ in construction 2.

Adversary $A_2^H(k,w)$:
choose $\tilde{E},\tilde{t},\tilde{t'}$ and simulate BC'_k queries using $\tilde{E},\tilde{t},\tilde{t'}$

$b \leftarrow \mathcal{D}^{\underline{\mathsf{BC}'_a},H}$

return b

Adversary $B_2^H(k,j)$:
choose $\tilde{E},\tilde{t},\tilde{t'}$ and simulate BC'_k queries using $\tilde{E},\tilde{t},\tilde{t'}$
until j-th, H-query:
    $b \leftarrow \mathcal{D}^{\underline{\mathsf{BC}'_k},H}$
if $\mathcal{D}$ makes $< j$ queries
    return $\perp$
m = measured arg. of H
return m

where

$$\mathsf{BC}'_k(x,y) := \tilde{E}(\mathsf{droplastbit}(\bar{x}),\mathsf{droplastbit}(\bar{y})) \quad \| \quad \tilde{t}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(x) \quad \| \quad \tilde{t'}(\bar{x},\bar{y}) \oplus \mathsf{lastbit}(y),$$

and $\tilde{E} : \{0,1\}^{2n-2} \to \{0,1\}^{2n-2}$ and $\tilde{t},\tilde{t'} : \{0,1\}^{2n} \to \{0,1\}$ are chosen randomly. We define:

$$G_2 := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{3n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right), b' \leftarrow A_2^H(k,w)\right]$$

$$P_{B_2} := \Pr\left[k = k' : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B_2^H(k,j)\right]$$

Therefore, $P_B \leq P_{B_2} + \epsilon$ for some *negligible* ϵ as the only difference between the adversary B and B_2 is that B uses classical access to standard secure PRFs E_{w_1}, t_{w_2}, and t'_{w_3} where w_1, w_2, and w_3 are randomly chosen while B_2 uses random oracles $\tilde{E}$, $\tilde{t}$, and $\tilde{t'}$. Below we simulate the following classical oracles $\tilde{E}_1$, $\tilde{t}_1$, $\tilde{t'}_1$, $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t'}_2$:

Sampling 1:

Oracle $\tilde{E}_1$:
Upon query on x:
$\tilde{E}_1(x) = \begin{cases} \text{previous stored value,} & x \text{ queried before} \\ \text{store random element,} & x \text{ not queried before} \end{cases}$
return $\tilde{E}_1(x)$

Oracle $\tilde{t}_1$:
Upon query on x:
$\tilde{t}_1(x) = \begin{cases} \text{previous stored value,} & x \text{ queried before} \\ \text{store random element,} & x \text{ not queried before} \end{cases}$
return $\tilde{t}_1(x)$

Oracle $\tilde{t'}_1$:
Upon Query on x:
$\tilde{t'}_1(x) = \begin{cases} \text{previous stored value,} & x \text{ queried before} \\ \text{store random element,} & x \text{ not queried before} \end{cases}$
return $\tilde{t'}_1(x)$

Sampling 2:

Oracle $\tilde{E}_2$:
Upon query on x:
$\tilde{E}_2(x)$ = choose at random
if (x queried before in $\tilde{E}_2$):
    bad event
return $\tilde{E}_2(x)$

Oracle $\tilde{t}_2$:
Upon Query on x:
$\tilde{t}_2(x)$ = choose at random
if (x queried before in $\tilde{t}_2$):
    bad event
return $\tilde{t}_2(x)$

Oracle $\tilde{t'}_2$:
Upon Query on x:
$\tilde{t'}_2(x)$ = choose at random
if (x queried before in $\tilde{t'}_2$):
    bad event
return $\tilde{t'}_2(x)$

It is easy to see that oracles $\tilde{E}_1$, $\tilde{t}_1$, and $\tilde{t'}_2$ perfectly simulate oracles $\tilde{E}$, $\tilde{t}$ and $\tilde{t'}$ respectively, whereas oracles $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t'}_2$ output a random string. Similar to adversary B_2 we define adversary B_3 except that it uses $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t'}_2$ instead of the functions $\tilde{E}$, $\tilde{t}$, and $\tilde{t'}$ respectively. Hence, we have:

$$P_{B_3} := \Pr\left[k = k' : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B_3^H(k,j)\right].$$

The difference between sampling 1 and sampling 2 is only when one of the inputs is repeated, this we call the bad event. Let,

$$G_3 := \Pr[\text{bad event} : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B_3^H(k,j)]$$

Hence, by fundamental lemma of game playing [3] we have that: $P_{B_2} \leq G_3 + P_{B_2}$.

In game G_3 and P_{B_3}, k is independent of k' because in sampling 2 $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t}'_2$ gives outputs that are independent of their inputs, and k is only used in the inputs of $\tilde{E}_2$, $\tilde{t}_2$ and $\tilde{t}'_2$. Hence, we can replace string k in the input of adversary B_3 with the null string 0.

$$P_{B_3} = \Pr\left[k = k' : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B_3^H(0,j)\right] \leq 2^{-n}.$$

Now we focus on the calculation of G_3 which denotes the probability of occurrence of the bad event. From the definition of G_3 we see that bad event occurs only when same input is queried again. Let there be two queries $q_{i'}$ and $q_{j'}$ on BC'_k that leads to same query on $\tilde{E}$, $\tilde{t}$, and $\tilde{t}'$, then we have,

$$G_3 = \Pr[\exists i' \neq j' s.t. \underbrace{q^1_{i'} \oplus (k\|1)\cdot \mathsf{lastbit}(q^1_{i'}) = q^1_{j'} \oplus (k\|1)\cdot \mathsf{lastbit}(q^1_{j'})}_{:=C_1} \text{ and}$$

$$\underbrace{q^2_{i'} \oplus f_k(q^1_{i'})\cdot \mathsf{lastbit}(q^2_{i'}) = q^2_{j'} \oplus f_k(q^1_{j'})\cdot \mathsf{lastbit}(q^2_{j'})}_{:=C_2} : j \xleftarrow{\$} \{1,\ldots,q\}; k \xleftarrow{\$} \{0,1\}^{n-1};$$

$$H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); k' \leftarrow B_3^H(0,j)]$$

where $q^1_{j'}$ and $q^2_{j'}$ denotes the first and second half of the query $q_{j'}$, similarly for $q_{i'}$.

From above conditions we have 16 different cases based on the last bits of $q^1_{i'}$, $q^2_{i'}$, $q^1_{j'}$, and $q^2_{j'}$. These conditions can be easily analysed and is found that in the conditions $C_1 \wedge C_2$ hold for at most one value of k. Since $q_{i'}$ and $q^1_{j'}$ are independent of k, for each i', j', $C_1 \wedge C_2$ holds with probability $\leq 2^{-n+1}$. Let r be the total number of BC queries and $q_1, \cdots, q_r$ are the queries to BC. Hence, we have:

$$G_3 = \frac{r(r-1)}{2} 2^{-n+1} \text{ which is negligible.}$$

We now have,

$$P_{B_2} \leq P_{B_2} + \epsilon \leq P_{B_3} + G_3 \leq 2^{-n} + \frac{r(r-1)}{2} 2^{-n+1} \approx negl.$$

Therefore,

$$P_B \leq 2^{-n} + \frac{r(r-1)}{2} 2^{-n+1} + \epsilon \approx negl.$$

Hence, $|G_0 - P^2_A| = |P^1_A - P^2_A|$ is *negligible*. Thus, $\delta = |P^2_A - G_1| + negl$.

The only difference between BC'_k and $\tilde{\mathsf{BC}}^k_w$ is that the underlying functions $\tilde{E}$, $\tilde{t}$, and $\tilde{t}'$ of BC'_k are random oracles whereas those of $\tilde{\mathsf{BC}}^k_w$ are standard secure PRFs. Thus, by the definition of PRF $|P^2_A - G_2| \leq \epsilon$. Using the sampling arguments as previously, we define adversary A_3 similar to adversary A_2 except that the underlying functions of BC'_k are $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t}'_2$.

$$G_4 := \Pr\left[b' = 1 : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{3n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A_3^H(k,w)\right].$$

$$G_5 := \Pr\left[\text{bad event} : k \xleftarrow{\$} \{0,1\}^{n-1}; w \xleftarrow{\$} \{0,1\}^{3n}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A_3^H(k,w)\right]$$

Using the fundamental theorem of game playing [3] we have that $|G_2 - G_4| \leq G_5$. Output of A_3 in G_4 and G_5 does not depends on k as the oracles $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t}'_2$ output random strings. Hence, we can replace the input string k of A_3 with a null string 0.

$$G_4 = \Pr\left[b' = 1 : w \xleftarrow{\$} \{0,1\}^{3n}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A_3^H(0, y)\right],$$

$$G_5 = \Pr\left[\text{bad event} : w \xleftarrow{\$} \{0,1\}^{3n}; k \xleftarrow{\$} \{0,1\}^{n-1}; H \xleftarrow{\$} \left(\{0,1\}^n \to \{0,1\}^{3n}\right); b' \leftarrow A_3^H(0, y)\right].$$

We have that $G_5 \leq \frac{r(r-1)}{2} 2^{-n+1}$ which is *negligible* analogously to $G_3 \leq \frac{r(r-1)}{2} 2^{-n+1}$. Hence, $|G_2 - G_4| \approx negl.$. Now we can see that adversary A_3 in game G_4 has completely random function instead of BC'_k as it uses random functions $\tilde{E}_2$, $\tilde{t}_2$, and $\tilde{t}'_2$. Note that the function in G_4 gives different values upon queries with the same input while R_F in G_1 gives equal outputs in that case. However, we assumed above that $\mathcal{D}$ never performs the same query twice. Hence, $G_1 = G_4$. Therefore, using above results we have that:

$$\delta = |G_0 - G_1| \approx |P_A^2 - G_1| \approx |G_2 - G_1| \approx |G_4 - G_1| = negl.$$

Thus, we have proved that the given construction is pseudo-random and hence a standard secure PRF.

3.3 Attack on CBC mode of operation

We choose a block cipher BC as in construction 1 in Section 3.1 for the construction of the Π_{CBC} scheme (Definition 6). As proved, this block cipher is a standard secure PRF (i.e., if the quantum adversary has only classical access to it).

Lemma 2. *There exists a standard secure pseudo-random function such that Π_{CBC} is not IND-qCPA secure. (In the quantum random oracle model)*

Proof. Let the Π_{CBC} scheme use the block cipher BC, we use one block message to attack the Π_{CBC} scheme. We know that the adversary has quantum access to the Π_{CBC} scheme, hence a quantum adversary can query the superposition of all messages of size equal to the block length of BC (i.e., n). The adversary prepares the quantum registers M and C to store quantum messages and receive quantum cipher-texts respectively. The adversary then stores the superposition of all one block messages in M (i.e., $\sum_m 2^{-n/2}|m\rangle$) and string $|0^{2n-1}\rangle|+\rangle$ in C respectively, and makes an encryption query. The corresponding reply is then stored in the quantum register C. The attack has been sketched in Figure 2.

After application of encryption algorithm Enc of Π_{CBC} the message and cipher-text registers contain the following data (up to normalization):

$$|M, C\rangle = \sum_m |m\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(m \oplus c_0))\rangle|+\rangle.^8$$

The adversary now XORs c_0 to the message register by using a CNOT gate. Hence, the quantum bits of the system changes to

$$\sum_m |m \oplus c_0\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(m \oplus c_0))\rangle|+\rangle.$$

Using $y = m \oplus c_0$ we have,

$$\sum_m |y\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(y))\rangle|+\rangle.$$

[8] Here, k is the key for the block cipher BC.

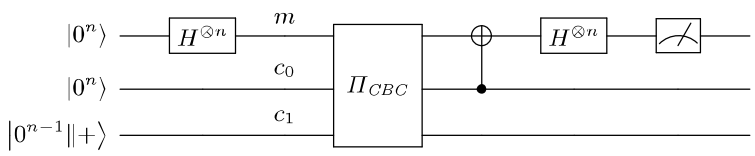

Fig. 2: Attack on 1 block CBC using Simon's algorithm

Also by construction of BC_k, this equals

$$\sum_y |y\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(y \oplus (k\|1)))\rangle|+\rangle$$
$$= \sum_y |y \oplus (k||1)\rangle|c_0\rangle|\mathsf{droplastbit}(\mathsf{BC}_k(y))\rangle|+\rangle.$$

Hence the adversary has the state (up to normalization),

$$\sum_y (|y\rangle + |y \oplus (k\|1)\rangle)|c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(y))\rangle|+\rangle.$$

We now apply n Hadamard gates (*i.e.*, $H^{\otimes n}$) giving us the following state (up to normalization):

$$\sum_y \sum_z ((-1)^{y\odot z} + (-1)^{(y\oplus(k\|1))\odot z})|z\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(y))\rangle|+\rangle$$

This is equal to

$$\sum_y \sum_z (-1)^{y\odot z}(1 + (-1)^{z\odot(k\|1)})|z\rangle \ |c_0\rangle \ |\mathsf{droplastbit}(\mathsf{BC}_k(y))\rangle|+\rangle.$$

Now if the $n-$bits of message register is measured one gets a vector z such that $z \odot (k\|1) = 0$ else the superposition collapses to 0. Hence, to retrieve k we can repeat the same attack again and again until we get n independent vectors v_i. Now using the Gaussian elimination one can retrieve the $n-1$ bits of k, thereby breaking the Π_{CBC} scheme.

3.4 Attack on CFB mode of operation

Here we provide an attack on the CFB mode of operation by using the similar attack as CBC by the use of Simon's algorithm.

Lemma 3. *There exists a standard secure pseudo-random function such that Π_{CFB} is not IND-qCPA secure. (In the quantum random oracle model.)*

Proof. Similar to the construction of Π_{CBC} scheme we choose a PRF BC as in construction 1 in Section 3.1 for the construction of Π_{CFB} scheme. We know that this PRF is secure given the quantum adversary has classical oracle access to it.

We show a key recovery attack on Π_{CFB} scheme by using encryption queries on messages with two blocks and then apply Simon's algorithm to retrieve the period a of the underlying PRF BC. It is known that the adversary has quantum access to the Π_{CBC} scheme, therefore a quantum adversary stores an equal superposition of messages and zero string of length equal to the block length of BC (i.e., n) in m_1 and m_2 blocks of register M respectively. It then initializes the quantum cipher-text register C with string $|0^{3n-1}\rangle|+\rangle$ of length $3n$. Adversary now makes an encryption query to Π_{CFB} encryption function which replies with the corresponding cipher-text in quantum register C. This attack has been sketched in the Figure 3

Hence, the registers now contain the following data (up to normalization)

$$\big|M, C\big\rangle = \sum_{m_1} \big|m_1 \big\| 0^n\big\rangle \ \big|c_0\big\rangle \ \big|\mathsf{BC}_k(c_0) \oplus m_1\big\rangle \ \big|\mathsf{droplastbit}(\mathsf{BC}_k(\mathsf{BC}_k(c_0) \oplus m_1))\big\rangle |+\rangle,$$

where m_i and c_i denotes the $i-th$ message and cipher-text blocks respectively. Also, we have that

$$\begin{aligned}\big|M, C\big\rangle &= \sum_{m_1} \big|m_1 \big\| 0^n\big\rangle \ \big|c_0\big\rangle \ \big|\mathsf{BC}_k(c_0) \oplus m_1\big\rangle \ \big|\mathsf{droplastbit}(\mathsf{BC}_k(\mathsf{BC}_k(c_0) \oplus m_1 \oplus (k\|1)))\big\rangle |+\rangle \\ &= \sum_{m_1} \big|m_1 \big\| 0^n\big\rangle \ \big|c_0\big\rangle \ \big|\mathsf{BC}_k(c_0) \oplus m_1 \oplus (k\|1))\big\rangle \ \big|\mathsf{droplastbit}(\mathsf{BC}_k(\mathsf{BC}_k(c_0) \oplus m_1)\big\rangle |+\rangle\end{aligned}$$

Let $\Upsilon = \mathsf{BC}_k(c_0) \oplus m_1$, then (up to normalization)

$$\big|M, C\big\rangle = \sum_{m_1} \big|m_1 \big\| 0^n\big\rangle \ \big|c_0\big\rangle \ \big(\big|\Upsilon\big\rangle + \big|\Upsilon \oplus (k\|1)\big\rangle\big) \ \big|\mathsf{droplastbit}(\mathsf{BC}_k(\Upsilon))\big\rangle |+\rangle$$

The adversary now applies $n-$Hadamard gates to the second block of ciphertext and gets (up to normalization):

$$\big|M, C\big\rangle = \sum_{m_1} \sum_{z} ((-1)^{\Upsilon \odot z} + (-1)^{(\Upsilon \oplus (k\|1)) \odot z}) \big|m_1 \big\| 0^n\big\rangle \ \big|c_0\big\rangle \ \big|z\big\rangle \ \big|\mathsf{droplastbit}(\mathsf{BC}_k(y))\big\rangle |+\rangle$$

Now if we measure the second block of ciphertext register we get a vector z such that $z \odot (k\|1) = 0$ else the superposition collapses to 0. Hence, to retrieve k we can repeat the same attack again and again until we get n independent vectors z_i's. Now using the gaussian elimination one can retrieve the $n-1$ bits of k, thereby breaking the Π_{CFB} scheme.

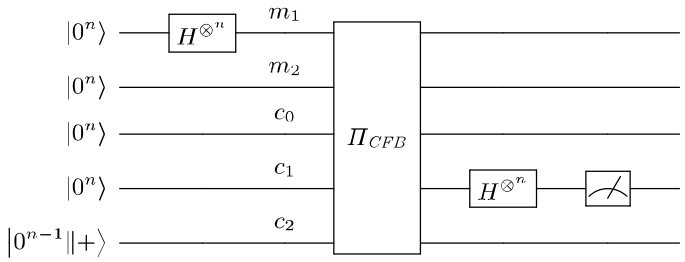

Fig. 3: Attack on 2 block CFB using Simon's algorithm

3.5 Attack on XTS mode of Operation

Lemma 4. *There exists a standard-secure pseudo-random function (in the random oracle model) such that Π_{XTS} admits an attack of the following form: The adversary first performs a number of superposition encryption queries. Then the adversary performs a superposition encryption query where the first half of the plaintext is an adversary chosen superposition of messages, and the second half is a bitstring m unknown to the adversary. Then the adversary can compute m.*

Proof. Consider the XTS scheme using the block cipher BC_k from construction 2. We first show an attack on the XTS scheme to retrieve the key k_1 as in Definition 10. It can be seen from construction 2 that a message block is $2n$-bits long. A quantum adversary prepares the message register M with all 0 bits and cipher-text register C is set such that it has the state $\big|0^{2n-2}\big\rangle |++\rangle$ in each block. Now the adversary applies n Hadamard gates (i.e., $H^{\otimes n}$) on first

half of every message block. [9] Hence the adversary possesses the $i-th$ block of message such that it contains a superposition of all $n-$bits strings in its first half and 0^n string in its 2nd half (i.e., $\sum_{m_i^1} 2^{n/2}\big|m_i^1\big\|0^n\big\rangle$). The adversary then queries on this message and receives the corresponding cipher-text (up to normalization) such that

$$\sum_M |M\rangle|C\rangle$$

where $M = (m_1^1\|0^n)(m_2^1\|0^n)\cdots(m_{p(t)}^1\|0^n)$ and
$C = \big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_1^1 \oplus L_1, L_2)\oplus L)\big\rangle\big|++\big\rangle\cdots\big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_i^1 \oplus (\alpha^{i-1}L)_1, (\alpha^{i-1}L)_2)\oplus \alpha^{i-1}L)\big\rangle\big|++\big\rangle\cdots$. As per the construction of BC_k, the first half of input to PRFs E, t, and t' is $\bar{x} = (x \oplus (k\|1)\cdot \mathsf{lastbit}(x))$ where x is the first half of input to BC. Hence, first $2n-2$ bits of the output of BC_k is periodic in the first input with period $k\|1$. We look at the $i-th$ message block and corresponding cipher-text. We have the following output on the $i-th$ message block (up to normalization)

$$\big|M, C\big\rangle := \sum_{m_i^1} \big|m_i^1\big\|0^n\big\rangle\big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_i^1 \oplus (\alpha^{i-1}L)_1, (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\big\rangle\big|++\big\rangle.$$

Hence we have (up to normalization),

$$\big|M, C\big\rangle = \sum_{m_i^1}(|m_i^1\big\|0^n\rangle + |m_i^1 \oplus (k\|1)\big\|0^n\rangle)\big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_i^1 \oplus (\alpha^{i-1}L)_1, (\alpha^{i-1}L)_2)\oplus\alpha^{i-1}L)\big\rangle\big|++\big\rangle$$

Now if we apply n Hadamard gates ($H^{\otimes n}$) on the first half of this $i-th$ message block we get the state (up to normalization):

$$\sum_{m_i}\sum_{y}(-1)^{m_i^1\odot y}(1 + (-1^{(k\|1))\odot y}))\big|y\big\|0^n\big\rangle$$
$$\big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_i^1 \oplus (\alpha^{i-1}L)_1, (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\big\rangle\big|++\big\rangle$$
$$= \sum_{m_i}\sum_{y:(k\|1)\odot y=0}(-1)^{m_i^1\odot y}\underbrace{\big|y\big\|0^n\big\rangle}_{msg\ block}\underbrace{\big|\mathsf{droplastbit}^2(\mathsf{BC}_k(m_i^1 \oplus (\alpha^{i-1}L)_1, (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\big\rangle\big|++\big\rangle}_{ciphertext\ block}.$$

Therefore, it is easy to see that if one measures the first half of the $i-th$ message block we get a uniformly random vector y such that $y \odot (k\|1) = 0$. To retrieve k we need $n-1$ independent vectors y_i's such that $y_i \odot (k\|1) = 0$ on which we can perform Gaussian elimination. These vectors can be achieved by applying the same procedure on sufficiently many message blocks.

With this we have the first key of XTS. This is not yet sufficient to break XTS as we need to retrieve L to be able to break the scheme. It is to be noted that every time a message is encrypted a new L is randomly generated. Hence, we cannot use different ciphertexts to retrieve L, if we are to use Simon's algorithm. Therefore, we will use the same ciphertext for determining L, and to contain the plaintext that we want to retrieve. For this we prepare the message register such that the $i-th$ message block will now have 0^n string set in its first half and superposition of all strings set in the second half (i.e., $m_i = \sum_{m_i^2} 2^{n/2}\big|0^n\big\|m_i^2\big\rangle$). In addition to those blocks, the message register contains a message $|m^*\rangle$ unknown to the adversary that he wants to decrypt. Thus we have that the cipher-text received will be periodic in its second half rather than first half as we did before. The received cipher-text up to normalization is

$$\sum_M |M\rangle|C\rangle \otimes |m^*\rangle$$

[9] Note that here every message block is $2n-$bits long.

where $M = (0^n \| m_1^2)(0^n \| m_2^2) \cdots (0^n \| m_{p(t)}^2)$ and
$C = \left|\mathsf{droplastbit}^2(\mathsf{BC}_k(L_1, m_1^2 \oplus L_2) \oplus L)\right\rangle \left|++\right\rangle \cdots \left|\mathsf{droplastbit}^2(\mathsf{BC}_k((\alpha^{i-1}L)_1, m_i^2 \oplus (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\right\rangle \left|++\right\rangle \cdots$. We have the following output on the $i-th$ message block (up to normalization)

$$\sum_{m_i^2} \left|0^n \| m_i^2\right\rangle \ \left|\mathsf{droplastbit}^2(\mathsf{BC}_k((\alpha^{i-1}L)_1, m_i^2 \oplus (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\right\rangle \left|+ + \right\rangle.$$

We have that the output of block cipher BC_k has period $f_k((\alpha^{i-1}L)_1)$ in its second half. Hence this equals (up to normalization):

$$\sum_{m_i^2} \left(|0^n \| m_i^2\rangle + |0^n \| m_i^2 \oplus f_k((\alpha^{i-1}L)_1)\rangle\right) \left|\mathsf{droplastbit}^2(\mathsf{BC}_k((\alpha^{i-1}L)_1, m_i^2 \oplus (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\right\rangle \left|++\right\rangle$$

Hence, for an $i-th$ block if we apply an $n-$bit Hadamard gate $(H^{\otimes n})$on the second half of $i-th$ message block we have (up to normalization)

$$\sum_{m_i^2} \sum_{y} (-1)^{m_i^2 \odot y} (1 + (-1)^{f_k((\alpha^{i-1}L)_1) \odot y}) \left|0^n \| y\right\rangle$$

$$\left|\mathsf{droplastbit}^2((\mathsf{BC}_k((\alpha^{i-1}L)_1, m_i^2 \oplus (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\right\rangle \left|+ + \right\rangle$$

$$= \sum_{m_i^2} \sum_{y : f_k((\alpha^{i-1}L)_1) \odot y = 0} (-1)^{m_i^2 \odot y} \underbrace{\left|0^n \| y\right\rangle}_{msg\ \ block} \underbrace{\left|\mathsf{droplastbit}^2(\mathsf{BC}_k((\alpha^{i-1}L)_1, m_i^2 \oplus (\alpha^{i-1}L)_2) \oplus \alpha^{i-1}L)\right\rangle \left|+ + \right\rangle}_{ciphertext\ \ block}$$

Hence if we measure the second half of the message register we get a vector y such that $y \odot f_k((\alpha^{i-1}L)_1) = 0$. By the definition of $f_k(x) = x \oplus (0^{n-1} \| \mathsf{lastbit}(x)) \oplus (k \| 1)$ in construction 2 of Section 3.1 we have,

$$y_i \odot \left((\alpha^{i-1}L)_1 \oplus (0^{n-1} \| \mathsf{lastbit}((\alpha^{i-1}L)_1)) \oplus (k \| 1)\right) = 0,$$

where y_i is the vector y obtained for the i-th block.

Let E_n be a $2n \times 2n$ matrix with 1 on the n-th diagonal element and 0 elsewhere. In the matrix notation we have,

$$\left[\ y_i \| 0^n\ \right]_R \left(\left[\ \alpha^{i-1}\ \right]_M \left[\ L\ \right]_C + \left[\ E_n\ \right]_M \left[\ \alpha^{i-1}\ \right]_M \left[\ L\ \right]_C + \begin{bmatrix} k \\ 1 \\ 0^n \end{bmatrix}_C \right) = 0,$$

where $[]_C, []_M, []_R$ represents column matrix, square matrix, and row matrix respectively. This is equivalent to

$$\left[\ y_i \| 0^n\ \right]_R \left[\ I - E_n\ \right]_M \left[\ \alpha^{i-1}\ \right]_M \left[\ L\ \right]_C = \left[\ y_i \| 0^n\ \right]_R \begin{bmatrix} k \\ 1 \\ 0^n \end{bmatrix}_C,$$

and to

$$\left[\ \mathsf{droplastbit}(y_i) \| 0^{n+1}\ \right]_R \left[\ \alpha^{i-1}\ \right]_M \left[\ L\ \right]_C = \left[\ y_i \| 0^n\ \right]_R \begin{bmatrix} k \\ 1 \\ 0^n \end{bmatrix}_C.$$

Hence, we have a matrix equation such that if we know the $2n$ independent vectors $\mathsf{droplastbit}(y_i)$ we can retrieve L using Gaussian Elimination, thereby breaking the XTS scheme. These independent vectors can be obtained by applying the above attack on sufficiently many message blocks (within the same ciphertext). Using k_1 and L, the adversary can then decrypt the ciphertext blocks corresponding to $|m^*\rangle$ and thus get m^*.

4 IND-qCPA security of OFB and CTR modes of operation

In this section, we analyze the quantum security of OFB and CTR modes of operation. Our motive is to prove the security of these schemes against the quantum adversary based on IND-qCPA definition(Definition 2) in Section 2. These two modes of operation are similar in working thence similar proofs.

We provide a generic proof for any cryptographic-system with encryption function which XOR's the message with a random pad based on the length of message and random key. This proof shows that IND-qCPA security of the scheme reduces to the fact that it is IND-CPA secure.

Lemma 5. *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme with encryption algorithm as* $\mathsf{Enc}_k(M) = G_k(|M|; r) \oplus M$*, for randomness* r*, given message* M *and key* $k \leftarrow \mathsf{Gen}$*. If* Π *is IND-CPA secure then it is IND-qCPA secure.*

Proof. Let $\Pr[PrivK^{qCPA}_{\mathcal{A}_q,\Pi}(t) = 1] = \varepsilon(t) + \frac{1}{2}$, for a poly-time quantum adversary $\mathcal{A}_q$. We construct an efficient quantum adversary $\mathcal{A}$ such that $\Pr[PrivK^{CPA}_{\mathcal{A},\Pi}(t) = 1] = \varepsilon(t) + \frac{1}{2}$. Adversary $\mathcal{A}^{\mathsf{Enc}_k}(1^t)$ works as follows:

1. $\mathcal{A}$ prepares two quantum registers M and C being message and ciphertext registers respectively.
2. runs $\mathcal{A}_q$, whenever $\mathcal{A}_q$ queries encryption oracle on superposition of messages answer the queries in the following way:
   - the quantum message and $\left|0^{|M|}\right\rangle$ are stored in M and C respectively,
   - query $s := \mathsf{Enc}_k(0^{|M|}) = G_k(|M|; r)$, where r is the randomness.
   - apply unitary operator $U$to quantum register M and C where $U\left|M, C\right\rangle := \left|M, C \oplus M \oplus s\right\rangle$.
   - send the register $\left|M, C\right\rangle$ to the adversary $\mathcal{A}_q$.
3. when $\mathcal{A}_q$ asks the challenge query send it to the challenger and send received result back to $\mathcal{A}_q$.
4. continue to answer any encryption oracle query as in step 2.
5. $\mathcal{A}_q$ outputs the result b', send b' to the challenger.

It is clear that $\Pr[PrivK^{CPA}_{\mathcal{A},\Pi}(t) = 1] = \Pr[PrivK^{qCPA}_{\mathcal{A}_q,\Pi}(t) = 1] = \frac{1}{2} + \varepsilon(t)$ and $\mathcal{A}$ is poly-time.

Theorem 3. *If* E *is a standard secure pseudo-random function then* Π_{OFB} *and* Π_{CTR} *schemes are IND-qCPA secure.*

Proof. Π_{OFB} and Π_{CTR} schemes are IND-CPA secure when E is standard secure pseudo-random function. Thus, result follows from Lemma 5.

5 IND-qCPA security of CBC and CFB mode of operation

IND-qCPA security of CBC and CFB modes of operation are conditional on the existence of quantum secure primitives. We use the One-way to Hiding Lemma [15](Lemma 1) to prove the bound for any quantum adversary that attacks the system.

We define $\mathsf{Enc}^{i,H}_{CBC}(M) := c_0 c_1 \cdots c_n$, where $c_j \xleftarrow{\$} \{0,1\}^t$ for $j \leq i$ and $c_j = H(m_j \oplus c_{j-1})$ for $i < j \leq n$. Similarly we define, $\mathsf{Enc}^{i,H}_{CFB}(M) := c_0 c_1 \cdots c_n$, where $c_j \xleftarrow{\$} \{0,1\}^t$ for $j \leq i$ and $c_j = H(c_{j-1}) \oplus m_j$ for $i < j \leq n$.

In the next lemma we prove that probability of distinguishing the output of CBC $\mathsf{Enc}^{i,H}_{CBC}$ from $\mathsf{Enc}^{i+1,H}_{CBC}$ by a quantum adversary having access to oracle $\mathsf{Enc}^{i,H}_{CBC}$ is negligible in t, where t is the security parameter. As the proof for $\mathsf{Enc}^{i,H}_{CBC}$ and $\mathsf{Enc}^{i+1,H}_{CFB}$ is similar we provide the instances for $\mathsf{Enc}^{i,H}_{CFB}$ in parentheses $[\![]\!]$ wherever there is a difference. Also, we use $\mathsf{Enc}^{i,H}$ to represent the encryption functions of $\mathsf{Enc}^{i,H}_{CBC}$ and $\mathsf{Enc}^{i,H}_{CFB}$ to generalize the proof.

Lemma 6. *For any i with $i : 0 \leq i \leq p(t) - 1$, and every quantum adversary $\mathcal{A}$ that makes at most q_A queries,*

$$\Big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}};$$
$$b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\mathsf{Enc}^{i,H}(M_b))] - \Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\};$$
$$M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\mathsf{Enc}^{i+1,H}(M_b))]\Big| \leq O\left(\frac{p(t)^2 {q_A}^2}{2^{\frac{t}{2}}}\right),$$

where $p(t)$ is the maximum number of blocks in the message M and t is the length of each message block.

Proof.

$$\varepsilon(t) = \Big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}};$$
$$b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\mathsf{Enc}^{i,H}(M_b))] - \Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\};$$
$$M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\mathsf{Enc}^{i+1,H}(M_b))]\Big|$$

For a given message $M = m_0 m_1 \cdots m_n$ let $\widetilde{\mathsf{Enc}}^i_H(M, c_0, \cdots, c_i) := \hat{c}_1 \hat{c}_2 \cdots \hat{c}_n$ where

$$\hat{c}_j = \begin{cases} c_j & 0 \leq j \leq i \\ H(\hat{c}_{j-1} \oplus m_j) \quad [\![= H(\hat{c}_{j-1}) \oplus m_j]\!] & i < j \leq n \end{cases}$$

Then we have,

$$\varepsilon(t) = \Big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}};$$
$$c_0, \ldots, c_i \xleftarrow{\$} \{0,1\}^t; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\widetilde{\mathsf{Enc}}^i_H(M_b, c_0, \ldots, c_i))] -$$
$$\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}};$$
$$c_0, \ldots, c_{i+1} \xleftarrow{\$} \{0,1\}^t; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\widetilde{\mathsf{Enc}}^{i+1}_H(M_b, c_0, \ldots, c_{i+1}))]\Big| \quad (1)$$

We put $c_i := x \oplus m_b^{i+1} [\![= x]\!]$ where m_b^{i+1} is the $(i+1)^{th}$ block of the message M_b and $x \xleftarrow{\$} \{0,1\}^t$. This means that c_i is uniformly random as x is randomly chosen. Therefore,

$$\varepsilon(t) = \Big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{Enc^{i,H}};$$
$$c_0, \ldots, c_{i-1} \xleftarrow{\$} \{0,1\}^t, x \xleftarrow{\$} \{0,1\}^t, c_i := x \oplus m_b^{i+1} [\![:= x]\!];$$
$$b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\widetilde{\mathsf{Enc}}^i_H(M_b, c_0, \ldots, c_i))] - \Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\};$$
$$M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}; c_0, \ldots, c_{i-1} \xleftarrow{\$} \{0,1\}^t, x \xleftarrow{\$} \{0,1\}^t, c_i := x \oplus m_b^{i+1} [\![c_i := x]\!],$$
$$y \xleftarrow{\$} \{0,1\}^t, c_{i+1} := y [\![:= y \oplus m_b^{i+1}]\!]; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(\widetilde{\mathsf{Enc}}^{i+1}_H(M_b, c_0, \ldots, c_{i+1}))]\Big| \quad (2)$$

By definition of $\widetilde{\mathsf{Enc}}_H^i$, we have $\widetilde{\mathsf{Enc}}_H^i(M_b, c_0, \cdots, c_i) = \widetilde{\mathsf{Enc}}_H^{i+1}(M_b, c_0, \cdots, c_{i+1})$ with $c_{i+1} := H(x) \quad [\![:= H(x) \oplus m_b^{i+1}]\!]$. Hence,

$$\begin{aligned}\varepsilon(t) = \Big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}};\\ c_0, \ldots, c_{i-1} \xleftarrow{\$} \{0,1\}^t, x \xleftarrow{\$} \{0,1\}^t, c_i := x \oplus m_b^i[\![:= x]\!], c_{i+1} := H(x)[\![:= H(x) \oplus m_b^{i+1}]\!];\\ b' \leftarrow \mathcal{A}^{Enc^{i,H}}(\widetilde{\mathsf{Enc}}_H^{i+1}(M_b, c_0, \ldots, c_{i+1}))] - \Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t),\\ b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}; c_0, \ldots, c_{i-1} \xleftarrow{\$} \{0,1\}^t, x \xleftarrow{\$} \{0,1\}^t, y \xleftarrow{\$} \{0,1\}^t,\\ c_i := x \oplus m_b^i[\![:= x]\!], c_{i+1} := y[\![:= y \oplus m_b^{i+1}]\!]; b' \leftarrow \mathcal{A}^{Enc^{i,H}}(\widetilde{\mathsf{Enc}}_H^{i+1}(M_b, c_0, \ldots, c_{i+1}))]\Big|\end{aligned}$$

We define an adversary A_{O2H} that makes oracle queries to random function $H \xleftarrow{\$} (\{0,1\}^t \to \{0,1\}^t)$. A_{O2H} with given inputs x and y does the following:

Adversary $A_{O2H}^H(x, y)$:

$M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}$
$b \xleftarrow{\$} \{0,1\}$
$c_0, \ldots, c_{i-1} \xleftarrow{\$} \{0,1\}^t; c_i = x \oplus m_b^{i+1}[\![= x]\!]; c_{i+1} = y[\![= y \oplus m_b^{i+1}]\!];$
compute $C := \widetilde{\mathsf{Enc}}_H^i(M_b, c_0, c_1, \ldots, c_{i+1})$

$b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{i,H}}(C)$

return $b' = b$

We note here that adversary A_{O2H} can answer the adversary $\mathcal{A}$'s query as it has oracle access to H. Let q_{o2h} be the number of H-queries made by $\mathcal{A}_{O2H}$, it is clear that $q_{o2h} \leq 3p(t)q_A$. Let q_1, q_2 and q_3 denote the number of queries that A_{O2H} makes to H before the challenge query, during challenge query and after challenge query respectively. [10]

It is clear that:

$$\begin{aligned}\varepsilon(t) = \Big|&\Pr[\tilde{b} = 1 : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), x \xleftarrow{\$} \{0,1\}^t, \tilde{b} \leftarrow A_{O2H}^H(x, H(x))]\\ &- \Pr[\tilde{b} = 1 : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), x \xleftarrow{\$} \{0,1\}^t, y \xleftarrow{\$} \{0,1\}^t, \tilde{b} \leftarrow A_{O2H}^H(x, y)]\Big| \quad (3)\end{aligned}$$

Let B be an oracle algorithm described in the O2H lemma(Lemma 1). Therefore, we have that $\varepsilon(t) \leq 2q_{o2h}\sqrt{P_B}$, where we have the probability P_B as

$$\begin{aligned}P_B = \Pr[x = x' : j \xleftarrow{\$} \{1, \ldots, q_{o2h}\}, x \xleftarrow{\$} \{0,1\}^t, H \xleftarrow{\$} (\{0,1\}^t \to \{0,1\}^t),\\ x' \leftarrow B^H(x, j)]\end{aligned}$$

$$= \frac{1}{q_{o2h}} \cdot \underbrace{\Pr[x = x' : x \xleftarrow{\$} \{0,1\}^t, H \xleftarrow{\$} (\{0,1\}^t \to \{0,1\}^t), x' \leftarrow B^H(x, j)]}_{:= P_B^j}$$

To evaluate P_B^j we consider three cases depending whether the j-th H-query is before, during, or after the challenge query.

[10] We can assume without loss of generality that $\mathcal{A}_{O2H}$ performs exactly q_1, q_2, q_3 queries respectively. If it performs less, we simply add dummy queries.

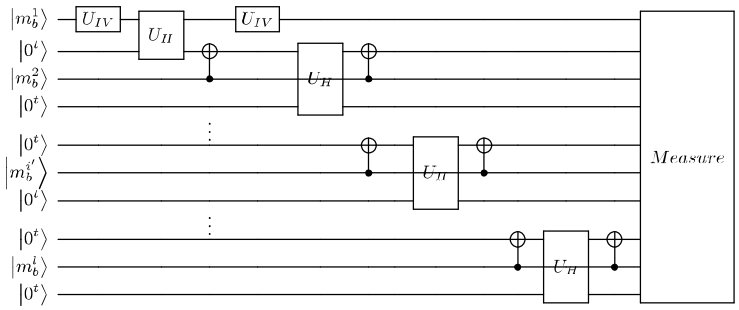

Fig. 4: Composition of Encryption Oracle using H oracle .

Case I ($j \leq q_1$):

In this case, the j-th iteration query to the oracle H is computed before the challenge query is done. So adversary $\mathcal{A}$ does not get access to x while queries are done. Therefore, adversary $\mathcal{A}$'s queries are independent of x, as it never executes challenge query and beyond. As the adversary $\mathcal{A}$ never used the x for any query we can therefore say that fixing x to be any string should not affect argument of the query. Therefore, we fix input x as the null string 0^n.

$$P_B^j = \Pr[x = x' : x \xleftarrow{\$} \{0,1\}^t, H \xleftarrow{\$} (\{0,1\}^t \to \{0,1\}^t), x' \leftarrow B^H(0,j)] \leq 2^{-t}.$$

Case II ($q_1 \leq j \leq q_1 + q_2$):

In this case the j-th iteration query to the oracle H is made during the challenge query (*i.e*, $q_1 < j \leq q_1 + q_2$). Therefore, oracle algorithm B can stop adversary $\mathcal{A}$ at any of the following queries:

$$H(m_b^{i+2} \oplus y), H(m_b^{i+3} \oplus H(m_b^{i+2} \oplus y)), \cdots, H(m_b^{p(t)} \oplus H(m_b^{p(t)-1} \oplus \cdots H(m_b^{i+2} \oplus y) \cdots))$$

$$\left[\!\left[H(y) \oplus m_b^{i+2}, H(H(y) \oplus m_b^{i+2}) \oplus m_b^{i+3}, \cdots, H(H(H(\cdots H(y) \oplus m_b^{i+2}) \cdots)) \oplus m_b^{p(t)}\right]\!\right]$$

By using result from Zhandry [19] on distinguishing a random function from a random permutation we have,

$$P_B^j \leq \Pr[x = x' : H \xleftarrow{\$} \mathsf{Perm}(), x \xleftarrow{\$} \{0,1\}^t, x' \leftarrow B^H(x,j)] + O\left(\frac{j^3}{2^t}\right)$$

Note that the argument of the j-th query is $s := m_b^{i+j-q_1+1} \oplus H(m_b^{i+j-q_1} \oplus \cdots \oplus H(m_b^{i+2} \oplus y) \cdots)$ $[\![s := H(\cdots H(H(y) \oplus m_b^{i+2}) \cdots \oplus m_b^{i+j-q_1}) \oplus m_b^{i+j-q_1+1}]\!]$. From the definition of O2H lemma we know that y is chosen independently at random from x and H. It is easy to see that for a fixed message M_b s would be assigned an output by a permutation that is independent of x but dependent on y since the input to first call to H is $m_b^{i+2} \oplus y$ $[\![y]\!]$. Therefore,

$$P_B^j \leq \Pr[x = x' : H \xleftarrow{\$} \mathsf{Perm}(), x \xleftarrow{\$} \{0,1\}^t, x' = s] + O(\frac{j^3}{2^t}) \leq \frac{1}{2^t} + O\left(\frac{j^3}{2^t}\right) \approx O\left(\frac{j^3}{2^t}\right)$$

Case III ($j \geq q_1 + q_2$):

In this case, the j-th iteration query to the oracle H is computed after the challenge query is done. We have $j > q_1 + q_2$. Adversary $\mathcal{A}$ makes many encryption oracle queries and eventually measures the argument of one of the H oracle query and stops. Say it measures in the k^{th} H oracle query of j-th encryption query.

$$P_B^j := \Pr[x = x' : x \xleftarrow{\$} \{0,1\}^t, H \xleftarrow{\$} (\{0,1\}^t \to \{0,1\}^t), x' \leftarrow B^H(x,j)]$$

The circuit diagram in Figure 4 represents the working of adversary A_{O2H}. A_{O2H} answers encryption queries using oracle access to H. Let the quantum message (possibly entangled) to be

stored in the quantum register M and the corresponding ciphertext in the quantum register C. The encryption circuit is composed of the quantum gates $U_{IV}, U_H, CNOT$ and measurements. Where $U_{IV}|M\rangle = |M \oplus IV\rangle$, $U_H|M, C\rangle = |M, C \oplus H(M)\rangle$, $CNOT|M, C\rangle = |M, C \oplus M\rangle$, and the measurements are in the computational basis of the message space. Thus, in each case I,II,III we have $P_B^j \in O\left(\frac{q_{o2h}^3}{2^{-t}}\right)$. [11]

The unitary gates used to compose the circuits are diagonal in the computational basis and hence commute with the measurements. Therefore, moving the measurements prior to the unitary operations do not affect the probability distribution of the output. Hence, we can measure the message register M before performing the unitary operations. Thus, it is similar to the Case II where we have a query on a classical message.

Therefore, we have $P_B^j = O(\frac{j^3}{2^t})$.

Hence by the definition of P_B we have, $P_B \leq O(\frac{q_{o2h}^3}{2^t})$. Therefore, we have that $\varepsilon(t) \leq q_{o2h}\sqrt{P_B} \leq q_{o2h}\sqrt{O(\frac{{q_{o2h}}^3}{2^t})} = O(\frac{{q_{o2h}}^3}{2^t})$

Theorem 4. *If the function E is a quantum secure PRF then Π_{CBC} and Π_{CFB} is IND-qCPA secure.*

Proof. For any efficient adversary $\mathcal{A}$ making q_A encryption queries using Lemma 6 and triangle inequality we have,

$$\begin{aligned}\big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow s\mathcal{A}^{\mathsf{Enc}^{0,H}}; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{0,H}}(\mathsf{Enc}^{0,H}(M_b))] \\ -\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{0,H}}; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{0,H}}(\mathsf{Enc}^{p(t),H}(M_b))]\big| \\ \leq nO\left(\frac{p(t)^3 {q_A}^3}{2^t}\right),\end{aligned}$$

One can see that $\mathsf{Enc}^{p(t),H}(M_b)$ outputs ciphertext as a completely random string. Hence, the output b' by adversary is independent of b. Therefore,

$$\begin{aligned}\big|\Pr[b = b' : H \leftarrow (\{0,1\}^t \to \{0,1\}^t), b \xleftarrow{\$} \{0,1\}; M_0, M_1 \leftarrow \mathcal{A}^{\mathsf{Enc}^{0,H}}; b' \leftarrow \mathcal{A}^{\mathsf{Enc}^{0,H}}(\mathsf{Enc}^{0,H}(M_b))] - \frac{1}{2}\big| \\ \leq p(t) \cdot O\left(\frac{p(t)^3 {q_A}^3}{2^t}\right).\end{aligned}$$

Note that $\mathsf{Enc}^{0,H}$ is indistinguishable from Enc function of Π by definition of qPRF. As the proof steps for CBC and CFB are same in Lemma 6 one could replace Enc_H^0 by Enc function of scheme Π_X where $X = \{CBC, CFB\}$. Therefore,

$$\big|\Pr[PrivK_{\mathcal{A},\Pi_X}^{qCPA}(t) = 1] - \frac{1}{2}\big| \leq O\left(\frac{p(t)^3 {q_A}^3}{2^t}\right) + negl(t).$$

as q_A is polynomial in t we deduce that,

$$\big|\Pr[PrivK_{\mathcal{A},\Pi_X}^{qCPA}(t) = 1] - \frac{1}{2}\big| \leq negl(t).$$

[11] Note that in Figure 4 we measure all registers, not only the query register. This does not change P_B^j since the additional measurements are performed on registers that are not used further.

References

1. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems (the hardness of quantum rewinding). In *FOCS 2014*, pages 474–483. IEEE, October 2014. Preprint on IACR ePrint 2014/296.
2. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73. ACM, 1993.
3. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pages 409–426, Berlin, Heidelberg, 2006. Springer-Verlag.
4. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.
5. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors. *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608. Springer, 2013.
6. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. `https://eprint.iacr.org/2013/088`, 2013. The definition of IND-qCPA only appear in this eprint, not in the conference version.
7. Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. In *ICITS 2013*, volume 8317 of *LNCS*, pages 142–161. Springer, 2014. Online version IACR ePrint 2011/421.
8. Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. In Carles Padró, editor, *Information Theoretic Security - 7th International Conference, ICITS 2013, Singapore, November 28-30, 2013, Proceedings*, volume 8317 of *Lecture Notes in Computer Science*, pages 142–161. Springer, 2013.
9. Ehsan Ebrahimi Targhi and Dominique Unruh. Quantum security of the fujisaki-okamoto transform. Technical report, Institute of Computer Science, University of Tartu, 2015. Available at http://www.cs.ut.ee/ unruh/qro.pdf.
10. European Union Agency for Network and Information Security (ENISA). Algorithms, key sizes and parameters report - 2013 recommendations. `https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report`, October 2013.
11. Phillip Rogaway. Evaluation of some blockcipher modes of operation. *Evaluation carried out for the Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan*, 2011.
12. Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.
13. Dominique Unruh. Quantum proofs of knowledge. In *Eurocrypt 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, April 2012.
14. Dominique Unruh. Everlasting multi-party computation. In *Crypto 2013*, volume 8043 of *LNCS*, pages 380–397. Springer, 2013. Preprint on IACR ePrint 2012/177.
15. Dominique Unruh. Revocable quantum timed-release encryption. *IACR Cryptology ePrint Archive*, 2013:606, 2013.
16. John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.
17. Mark Wooding. New proofs for old modes. *IACR Cryptology ePrint Archive*, 2008:121, 2008.
18. Mark Zhandry. How to construct quantum random functions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 679–687. IEEE Computer Society, 2012.
19. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.