Security Notions for Random Oracle Model in Classical and Quantum Settings

Jeeun Lee* Seunghyun Lee** Kwangjo Kim*

Abstract: The advent of quantum computers and their algorithms has opened the era of post-quantum and quantum cryptography. Accordingly, new security proof tools and notions in the quantum setting need to be settled in order to prove the security of cryptographic primitives appropriately. As the random oracle model is accepted as an efficient security proof tool, it has been suggested to extend it from classical to quantum setting by allowing adversary’s access to quantum power. In this paper, we look at the background of classical, quantum-accessible, and quantum random oracle models for classical, post-quantum, and quantum cryptography, respectively, and how they are defined. Also, the security notions for random oracle model in classical and quantum settings are introduced such as IND-ATK, (IND/wqIND/qIND)-qATK, and cqIND-qATK, for ATK ∈ {CPA, CCA1, CCA2}. Finally, comparison of different cryptography eras are provided.

Keywords: classical random oracle · quantum-accessible random oracle · quantum random oracle · quantum indistinguishability · quantum attack

1 Introduction

1.1 The Advent of Quantum Computers

As more and more refined classical, i.e., non-quantum, computers are developed, several problems have been encountered such as quantum tunnelling and heat generation. Quantum tunnelling is a phenomenon where a particle tunnels through a barrier that is deemed insurmountable in the classical world. Since the number of transistors in a dense integrated circuit has doubled approximately every 1.5 years [Moo65], the gaps between transistor terminals would shrink to the classical limits at some point. Then the electrons are able to move between terminals, that is, a transistor in an off state could be unexpectedly switched on even if it is not supposed to be. Also, classical computers use logically irreversible manipulation of information by erasing a bit or merging two computation paths. This necessarily implies physical irreversibility and corresponding heat increase [Lan61].

Quantum computers have been proposed as a natural solution to circumventing the aforementioned problems since 1970s. They are based on quantum mechanics, which apply to all systems ranging from micro to macro scales, and use quantum bits, i.e., qubits, to create quantum logic gates for quantum computing. A pure qubit can be represented as a linear superposition of the basis states, |ψ⟩ = α|0⟩ + β|1⟩, where the complex numbers α and β satisfy |α|^2 + |β|^2 = 1. We may then use n qubits to represent either 2^n states, or entangled states. Besides, they use logically reversible manipulation which requires no release of heat in principle [Lan61]. For these reasons, quantum computing has attracted research interest both academically and commercially since its initial proposal.

1.2 Security Proofs in the Quantum Setting

After the publication of Deutsch’s groundbreaking paper [Deu85], many quantum algorithms have been introduced, the most famous of which are Simon’s algorithm [Sim94], Shor’s algorithm [Sho94,Sho97], and Grover’s algorithm [Gro96,Gro97]. When large-scale quantum computers are available, Shor’s algorithm could break classical asymmetric encryption and digital signature schemes based on integer factorization and discrete logarithm problems in polynomial time. Also, classical symmetric encryption schemes would not be safe due to Grover’s algorithm and Simon’s algorithm. It has been believed until recently that doubling the key size would provide security against Grover’s algorithm [CJ16,ABB15], however, widely used modes of operation for authentication and authenticated encryption have proved to be completely broken using Simon’s algorithm [KLLNP16,SS16]. For these reasons, the cryptographic community is motivated to establish new security notions and proof models in the quantum setting [BDF+11,BJ15,CHS16,Gag17,SLL16].

The most notable security proof models for provable security are the standard and random oracle models. In the standard model, existence of certain basic primitives are assumed, e.g., one-way function, based on which more complex schemes are devised. Hence, cryptographic design in the model proceeds as follows: (a) assume hardness of a computational problem concern-
ing the basic primitive, and (b) prove that an attack necessarily reduces to solving the hard problem. However, in practice, we have access to more sophisticated primitives, e.g., DES, we may readily use. In the random oracle model, these sophisticated primitives are idealized as random oracles, which are in turn used for cryptographic design and security proof. In practice, random oracles are replaced by good hash functions [BR93]. In this paper, we focus on the random oracle model as it is accepted as a more efficient and feasible model than the standard model. Also, the extension of random oracle model from classical to quantum setting is explained and the suitable security notions of indistinguishability for each model are introduced.

1.3 Organization

The rest of this paper is organized as follows. First, the classical and quantum cryptographic primitives and some security notions are briefly recalled in Section 2. Random oracle models in classical, post-quantum, and quantum settings are explained from Sections 3 to 5. Also, suitable indistinguishability notions for each model are introduced. In Section 6, we conclude the survey by comparing security notions and proof models.

2 Preliminaries

2.1 Classical Cryptographic Primitives

**Definition 2.1 (Symmetric Encryption).** A symmetric encryption scheme \( \Pi_{\text{sym}} \) is a tuple of classical probabilistic polynomial-time algorithms \((\text{KeyGen}, \text{Enc}, \text{Dec})\) and sets called key space \( K \), message space \( M \), and ciphertext space \( C \) such that

- \( k \xleftarrow{} \text{KeyGen}(1^\lambda) \): the key generation algorithm \( \text{KeyGen} \) receives a security parameter \( \lambda \) and outputs key \( k \in K \).
- \( c \xleftarrow{} \text{Enc}_k(m) \): the encryption algorithm \( \text{Enc} \) uses the key \( k \) to encrypt a message \( m \in M \) and outputs a ciphertext \( c \in C \).
- \( m \xleftarrow{} \text{Dec}_k(c) \): the decryption algorithm \( \text{Dec} \) uses the key \( k \) to decrypt a ciphertext \( c \in C \) and outputs a message \( m \) or \( \bot \) denoting \( c \) is invalid.

For any \( k \) and any \( m \), the scheme should satisfy

\[
\Pr[\text{Dec}_k(\text{Enc}_k(m)) \neq m] = \text{negl}(\lambda).
\]

**Definition 2.2 (Asymmetric Encryption).** An asymmetric encryption scheme \( \Pi_{\text{asym}} \) is a tuple of probabilistic polynomial-time algorithms \((\text{KeyGen}, \text{QEnc}, \text{QDec})\) and sets called key space \( K \), message space \( M \), and ciphertext space \( C \) such that

- \( (pk, sk) \xleftarrow{} \text{KeyGen}(1^\lambda) \): the key generation algorithm \( \text{KeyGen} \) receives a security parameter \( \lambda \) and outputs a pair of corresponding public key \( pk \in K \) and secret key \( sk \in K \).
- \( c \xleftarrow{} \text{QEnc}_{pk}(m) \): the encryption algorithm \( \text{Enc}_{pk} \) uses the public key \( pk \) to encrypt a message \( m \in M \) and outputs a ciphertext \( c \in C \).
- \( m \xleftarrow{} \text{QDec}_{sk}(c) \): the decryption algorithm \( \text{Dec} \) uses the secret key \( sk \) to decrypt a ciphertext \( c \in C \) and outputs a message \( m \) or \( \bot \) denoting \( c \) is invalid.

For any \( (pk, sk) \) and any \( m \), the scheme should satisfy

\[
\Pr[\text{Dec}_{sk}(\text{Enc}_{pk}(m)) \neq m] = \text{negl}(\lambda).
\]

2.2 Quantum Cryptographic Primitives

**Definition 2.3 (Quantum Symmetric Encryption).** A quantum symmetric encryption scheme \( \Pi_{\text{qsym}} \) is a tuple of quantum probabilistic polynomial-time algorithms \((\text{KeyGen}, \text{QEnc}, \text{QDec})\) and sets called key space \( K \), message space \( D(H_M) \), and ciphertext space \( D(H_C) \) such that

- \( k \xleftarrow{} \text{KeyGen}(1^\lambda) \): the key generation algorithm \( \text{KeyGen} \) receives a security parameter \( \lambda \) and outputs key \( k \in K \).
- \( c \xleftarrow{} \text{QEnc}_k(m) \): the quantum encryption algorithm \( \text{QEnc} \) uses the key \( k \) to encrypt a message \( m \in D(H_M) \) and outputs a ciphertext \( c \in D(H_C) \).
- \( m \xleftarrow{} \text{QDec}_k(c) \): the quantum decryption algorithm \( \text{QDec} \) uses the key \( k \) to decrypt a ciphertext \( c \in D(H_C) \) and outputs a message \( m \) or \( \bot \) denoting \( c \) is invalid.

For any \( k \) and any \( m \), the scheme should satisfy

\[
\Pr[\hat{\text{QDec}}_k(\hat{\text{QEnc}}_k(m)) \neq m] = \text{negl}(\lambda).
\]

**Definition 2.4 (Quantum Asymmetric Encryption).** A quantum asymmetric encryption scheme \( \Pi_{\text{qasym}} \) is a tuple of quantum probabilistic polynomial-time algorithms \((\text{KeyGen}, \text{QEnc}, \text{QDec})\) and sets called key space \( K \), message space \( D(H_M) \), and ciphertext space \( D(H_C) \) such that

- \( (pk, sk) \xleftarrow{} \text{KeyGen}(1^\lambda) \): the key generation algorithm \( \text{KeyGen} \) receives a security parameter \( \lambda \) and outputs a random pair of corresponding public key \( pk \in K \) and secret key \( sk \in K \).
- \( c \xleftarrow{} \text{QEnc}_{pk}(m) \): the quantum encryption algorithm \( \text{QEnc} \) uses the public key \( pk \) to encrypt a message \( m \in D(H_M) \) and outputs a ciphertext \( c \in D(H_C) \).
- \( m \xleftarrow{} \text{QDec}_{sk}(c) \): the quantum decryption algorithm \( \text{QDec} \) uses the secret key \( sk \) to decrypt a ciphertext \( c \in D(H_C) \) and outputs a message \( m \) or \( \bot \) denoting \( c \) is invalid.

For any \( (pk, sk) \) and any \( m \), the scheme should satisfy

\[
\Pr[\hat{\text{QDec}}_{sk}(\hat{\text{QEnc}}_{pk}(m)) \neq m] = \text{negl}(\lambda).
\]

The concept of quantum encryption was first introduced in [BR00]. Here, the set of all density operators on a Hilbert space \( H_n \) is denoted as \( D(H_n) \). Note that a quantum encryption scheme uses a classical bit string for a key, and arbitrary quantum states for plaintexts and ciphertexts. Also, any quantum algorithm must be
a set of unitary operations\footnote{A unitary operation, any transformation that preserves the inner product, is used to make the norm of the physical state stay fixed.}. A classical key must be established among honest parties in order to encrypt and decrypt multiple times with the same key. As in classical cryptosystems, decryption of an encrypted plaintext under the same key must recover the original plaintext with negligible error.

### 2.3 Security Notions

As one of possible security goals, \textit{indistinguishability} formalizes an adversary’s advantage to distinguish the encryptions of two plaintexts of the same length [CMS4]. As possible attack models, three different attacks are considered: \textit{chosen-plaintext attack} (CPA), \textit{non-adaptive chosen-ciphertext attack} (CCA1), and \textit{adaptive chosen-ciphertext attack} (CCA2). Under CPA, the adversary can encrypt and decryption oracle access and obtains ciphertexts for plaintexts of their choice [CMS4]. Under CCA1, the adversary can decrypt oracle access before the challenge phase only [NY90], whereas under CCA2, the adversary has encryption and decryption oracle access before and after the challenge phase [RS01]. The CCA2 adversary, however, is not allowed to query the ciphertext itself to the decryption oracle. Hence, the decryption oracle is modified as follows:

\[
\mathsf{Dec}^\mathsf{ct}(c) = \begin{cases} \\
| & \text{if } c = c_b \\
\mathsf{Dec}(c) & \text{otherwise.}
\end{cases}
\]

Also, the term \textit{adaptive} is in respect of the challenge phase, not oracle’s answers. Under all attacks the adversary is allowed to choose queries adaptively to the oracle’s answers.

### 3 Classical Cryptography

#### 3.1 Classical Random Oracle Model

The classical random oracle (CRO) model is an efficient security proof tool introduced in [BR93] in order to bridge the gap between cryptographic theory and practice. For implementation of the ideal system in the real world, the following two steps are performed. First, one designs an ideal system where all parties have an oracle access to a truly random function \(f\) and proves the security of this system. Then one replaces the random oracle with a \textit{good} hash function. This means that we consider the hash function as a black box that responds to a query. The oracle makes an independent random choice for each query, however, it records its responses and returns the same answer for the same query. A classical query algorithm with \(q\) queries can be represented as below:

\[
\text{Alg}_c := \prod_{i=1}^{q} \{G_i \circ f\},
\]

where \(G_i\) is an input generator for an oracle \(O_f\). Although there have been controversies concerning too

### 3.2 Classical Security Notions

**Definition 3.1 (IND-ATK for \(\Pi_{\text{sym}}\)).** For \(\Pi \in \{\text{CPA, CCA1, CCA2}\}\), a symmetric encryption scheme \(\Pi_{\text{sym}}\) is said to be \textit{IND-ATK secure} if the advantage of any classical probabilistic polynomial-time adversary \(\mathcal{A} = (\mathcal{A}_M, \mathcal{A}_D)\), where \(\mathcal{A}_M\) and \(\mathcal{A}_D\) are a message generator and a distinguisher, respectively, winning the game is negligible.

\[
\text{Adv}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}}(\lambda) := 2 \cdot \text{Succ}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}} - 1 = \text{negl}(\lambda),
\]

where \(\text{Succ}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}}\) is as follows:

\[
\text{Pr} \left[ \begin{aligned}
(\mathsf{pk}, \mathsf{sk}) & \xleftarrow{\$} \text{KeyGen}(1^\lambda); (m_0, m_1, \text{state}) \xleftarrow{\$} \mathcal{A}_M^\text{C}\!,

b & \xleftarrow{\$} \{0, 1\};
\mathsf{cb} & \xleftarrow{\$} \mathcal{O}_{\mathsf{Enc}}(m_b);

b' & \xleftarrow{\$} \mathcal{A}_D^\text{C}\! (\mathsf{cb}, \text{state}) : b' = b
\end{aligned} \right] \text{ for (ATK, O1, O2)} = \left\{ \begin{array}{l}
(\text{CPA, O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Enc}}^\text{C}),

(\text{CCA1, \{O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Dec}}^\text{C}, \mathcal{O}_{\mathsf{Enc}}),

(\text{CCA2, \{O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Dec}}^\text{C}, \mathcal{O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Dec}}^\text{C})).
\end{array} \right\}.
\]

**Definition 3.2 (IND-ATK for \(\Pi_{\text{sym}}\)).** For \(\Pi \in \{\text{CPA, CCA1, CCA2}\}\), an asymmetric encryption scheme \(\Pi_{\text{sym}}\) is said to be \textit{IND-ATK secure} if the advantage of any classical probabilistic polynomial-time adversary \(\mathcal{A} = (\mathcal{A}_M, \mathcal{A}_D)\), where \(\mathcal{A}_M\) and \(\mathcal{A}_D\) are a message generator and a distinguisher, respectively, winning the game is negligible.

\[
\text{Adv}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}}(\lambda) := 2 \cdot \text{Succ}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}} - 1 = \text{negl}(\lambda),
\]

where \(\text{Succ}^\text{IND-ATK}_{\mathcal{A}, \Pi_{\text{sym}}}\) is as follows:

\[
\text{Pr} \left[ \begin{aligned}
(\mathsf{pk}, \mathsf{sk}) & \xleftarrow{\$} \text{KeyGen}(1^\lambda); (m_0, m_1, \text{state}) \xleftarrow{\$} \mathcal{A}_M^\text{C}\!,

b & \xleftarrow{\$} \{0, 1\};
\mathsf{cb} & \xleftarrow{\$} \mathcal{O}_{\mathsf{Enc}}(m_b);

b' & \xleftarrow{\$} \mathcal{A}_D^\text{C}\! (\mathsf{cb}, \text{state}) : b' = b
\end{aligned} \right] \text{ for}
\]

3
(ATK, O₁, O₂) = \begin{cases} \text{CPA}(\epsilon, \epsilon) \\
\text{CCA1}(O_{\text{Dec}a}, \epsilon) \\
\text{CCA2}(O_{\text{Dec}b}, O_{\text{Dec}c}). \end{cases}

The definition of IND-ATK for \( \Pi_{\text{asym}} \) was formalized in [BDPR98, Definition 2.1].

4 Post-quantum Cryptography

4.1 Quantum-Accessible Random Oracle Model

A classical query algorithm that computes a Boolean function \( f : \{0,1\}^n \rightarrow \{0,1\} \) by using oracle queries where the input is \( X = (x_0, \ldots, x_{n-1}) \) is called a decision tree. A decision tree can be represented as a binary tree where each node represents a query, and its two children represent the two possible outcomes of the query. A leaf node represents the final answer 0 or 1. The depth of the tree, i.e., the number of queries needed to compute \( f \), is the cost of an algorithm. This query model is useful in security proof since the number of queries an adversary needs to break a scheme corresponds to the time the attack takes.

Following [BBCP98], a quantum query algorithm with \( q \) queries is a quantum analogue of a classical decision tree with \( q \) queries, where we use the power of quantum parallelism by making queries and operations in quantum superposition. This can be represented as a sequence of unitary transformations:

\[ \text{Alg}_\text{Qa} := \hat{M}U_0\hat{O}_f \cdots U_1\hat{O}_f U_0\hat{G}. \]

Here, \( \hat{G} \) is an input generator for an oracle, and \( U_j \)'s are fixed unitary transformations that do not depend on inputs. The (possibly) identical \( \hat{O}_j \)'s are unitary transformations that correspond to an oracle, and \( M \) performs a measurement.

Consider a quantum system consisting of \( m \) qubits, with each qubit having basis states \( |0\rangle \) and \( |1\rangle \), such that there are \( 2^m \) possible basis states. Let each basis state be a binary string of length \( m \) or the corresponding natural number \( 0, 1, \ldots, 2^m - 1 \). We could think of \( f \) as a function that takes \( i \) as an input, and returns the \( i \)-th bit \( x_i \) as an output. Then the oracle transformation \( \hat{O}_f \), called quantum-accessible random oracle (QRO), maps basis state \( |i, b, z\rangle \) to \( |i, b \oplus x_i, z\rangle \), where the length of query register \( i \) is \( \lceil \log n \rceil \) qubits, answer register \( b \) is one qubit, ancilla register \( z \) is an arbitrary string of \( m - \lceil \log n \rceil - 1 \) qubits, and \( \oplus \) is exclusive or. Besides the standard way which maps basis state \( |i, b, z\rangle \) to \( |i, b \oplus g(x)\rangle \) for a general function \( g : \{0,1\}^n \rightarrow \{0,1\}^m \), there can be different transformations to implement an oracle such as Fourier phase oracle \( |i, b, z\rangle \rightarrow e^{2\pi i g(x)b/n^2}|i, b\rangle \) and minimal oracle \( |i\rangle \rightarrow |g(x)\rangle \) [KKYB02]. Using standard and minimal oracles, the following quantum encryption oracles are used for constructing security notions in Section 4.2, \( \hat{O}_{\text{Enc}} \) mapping basis state \( |m, c\rangle \) to \( |m, c \oplus \text{Enc}_k(m)\rangle \) and \( \hat{O}_{\text{Enc}} \) mapping basis state \( |m\rangle \) to \( |\text{Enc}_k(m)\rangle \).

Finally, the \( \text{Alg}_\text{Qa} \) is applied to an oracle-independent initial state \( |\psi_i\rangle \), which gives an oracle-dependent final state \( |\psi_f\rangle = \text{Alg}_\text{Qa}|\psi_i\rangle \). The computation ends with some measurement or observation of the final state.

4.2 Post-quantum Security Notions

The QaRO model replaces all classical communication with quantum communication by allowing an adversary to have both quantum encryption oracle access and quantum challenge queries. In this case, the adversary and the challenger are modelled as quantum circuits sharing a certain number of qubits. For this model, one of the first attempts at defining a security notion was to extend IND-CPA to fully-quantum indistinguishability under quantum chosen-plaintext attack (fqIND-qCPA), which renames [BZ13, Definition 4.1] for consistency. This security notion is the most naturally emerging concept for an entirely quantum game, however, no symmetric encryption scheme satisfies it due to the entanglement between plaintext and ciphertext:

**Theorem 4.1 (BZ Attack [BZ13, Theorem 4.2]).**

No symmetric encryption scheme achieves fqIND-qCPA security.

**Proof.** The proof [CHSU16, Theorem 2.7] can be interpreted as follows: As shown in Figure 1, the generic adversary \( \mathcal{A} \) prepares three quantum registers: two plaintext registers \( |q_0\rangle, |q_1\rangle \) and a ciphertext register \( |q_2\rangle \).

- They are initialized as \( |0^n\rangle \) and the initial quantum state is \( |\varphi_0\rangle = |0^n\rangle|0^n\rangle|0^n\rangle \).
- To put superposition of all possible messages in the second register, the Hadamard gate acts on \( |q_1\rangle \) and state becomes \( |\varphi_1\rangle = |0^n\rangle\sum_x2^{-n/2}|x\rangle|0^n\rangle \).
- When \( \mathcal{A} \) gets quantum encryption oracle access mapping basis state \( |q_0, q_1, q_2\rangle \) to \( |q_0, q_1, q_2 \oplus \text{Enc}_k(q_0)\rangle \), then we have two cases:
  - \( |\varphi_2\rangle = \begin{cases} |0^n\rangle H^\otimes n |0^n\rangle |\text{Enc}_k(0^n)\rangle & \text{if } b = 0 \\
|0^n\rangle\sum_x2^{-n/2}|x\rangle |\text{Enc}_k(x)\rangle & \text{if } b = 1. \end{cases} \)

- Measurement on \( |q_2\rangle \) gives
  - \( |\varphi_3\rangle = \begin{cases} |0^n\rangle H^\otimes n |0^n\rangle |\text{Enc}_k(0^n)\rangle & \text{if } b = 0 \\
|0^n\rangle|x\rangle |\text{Enc}_k(x)\rangle & \text{with prob. } 2^{-n} \text{ if } b = 1. \end{cases} \)
- Acting the Hadamard on \( |q_1\rangle \) again gives
  - \( |\varphi_4\rangle = \begin{cases} |0^n\rangle |0^n\rangle |\text{Enc}_k(0^n)\rangle & \text{if } b = 0 \\
|0^n\rangle|+\rangle^n|0^n\rangle |\text{Enc}_k(x)\rangle & \text{if } b = 1. \end{cases} \)
Finally, the measurement on $|q_1\rangle$ gives

$$|\varphi_b\rangle = \begin{cases} |0^n\rangle|0^n\rangle|\text{Enc}_b(0^n)\rangle & \text{if } b = 0 \\ |0^n\rangle|i\rangle|\text{Enc}_b(x_i)\rangle & \text{for } i \in \{0,1\}^n \text{ with prob. } 2^{-n} & \text{if } b = 1. \end{cases}$$

For $b = 0$, the measurement on $|q_1\rangle$ yields $|0^n\rangle$ with probability $1$. For $b = 1$, the measurement on $|q_1\rangle$ yields $|0^n\rangle$ with probability $2^{-n}$. The $A$ outputs $b' = 0$ iff the last outcome is $|0^n\rangle$, otherwise $b' = 1$. $\square$

In order to find weaker but achievable security notions, [GHS16] analyses 16 possible candidates by spanning a binary tree. [GHS16] considers the challenger model instead of the random oracle model, in order to rule out far too powerful adversaries. In this model, the adversary and the challenger do not share the same quantum circuits. The adversary now has an access to the quantum encryption oracle provided by an external challenger who cannot be rewound, whereas in the random oracle model, the adversary has a direct access to the quantum encryption oracle. Excluding unreasonable or unachievable notions, the following definitions are left: indistinguishability under quantum ATK (IND-qATK), weak-quantum indistinguishability under quantum ATK (wqIND-qATK), and quantum indistinguishability under quantum ATK (qIND-qATK).

**Definition 4.1 (IND-qATK for $\Pi_{sym}$).** For $\Pi \in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$, a symmetric encryption scheme $\Pi_{sym}$ is said to be IND-qATK secure if the advantage of any quantum probabilistic polynomial-time adversary $A = (\mathcal{A}_M, \mathcal{A}_D)$, where $\mathcal{A}_M$ and $\mathcal{A}_D$ are a message generator and a distinguisher, respectively, winning the game is negligible.

$$\text{Adv}_{\mathcal{A}, \Pi_{sym}}^{\text{IND-qATK}}(\lambda) := 2 \cdot \text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{IND-qATK}} - 1 = \text{negl}(\lambda),$$

where $\text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{IND-qATK}}$ is as follows:

$$\Pr \left[ k \xleftarrow{\$} \text{KeyGen}(1^\lambda); (m_0, m_1, \text{state}) \xleftarrow{\$} \mathcal{A}_M^\Pi; b \xleftarrow{\$} \{0,1\}; c_b \xleftarrow{\$} \text{Enc}_b(m_b); b' \xleftarrow{\$} \mathcal{A}_D^\Pi(c_b, \text{state}) : b' = b \right]$$

for $(\mathcal{A}_K, \mathcal{O}_1, \mathcal{O}_2) = \left\{ (\text{CPA}, \hat{\text{Enc}}, \hat{\text{Enc}}), (\text{CCA1}, \hat{\text{Enc}}, \hat{\text{Dec}}), \hat{\text{Dec}}) \right\}$. The definitions of IND-qCPA and IND-qCCA were discussed in [BZ13] Definition 4.5] and [BZ13] Definition 4.6, respectively.

**Definition 4.2 (wqIND-qATK for $\Pi_{sym}$).** For $\Pi \in \{\text{CPA}, \text{CCA1}\}$, a symmetric encryption scheme $\Pi_{sym}$ is said to be wqIND-qATK secure if the advantage of any quantum probabilistic polynomial-time adversary $A = (\mathcal{A}_M, \mathcal{A}_D)$, where $\mathcal{A}_M$ and $\mathcal{A}_D$ are a message generator and a distinguisher, respectively, winning the game is negligible.

$$\text{Adv}_{\mathcal{A}, \Pi_{sym}}^{\text{wqIND-qATK}}(\lambda) := 2 \cdot \text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{wqIND-qATK}} - 1 = \text{negl}(\lambda),$$

where $\text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{wqIND-qATK}}$ is as follows:

$$\Pr \left[ k \xleftarrow{\$} \text{KeyGen}(1^\lambda); (\mathcal{D}_1, \mathcal{D}_2) \xleftarrow{\$} \mathcal{A}_M^\Pi; b \xleftarrow{\$} \{0,1\}; c_b \xleftarrow{\$} \text{Qbd}(\mathcal{D}_1(\mathcal{D}_2)); c_b \xleftarrow{\$} \hat{\text{Enc}}_b(\mathcal{D}_1); b' \xleftarrow{\$} \mathcal{A}_D^\Pi(c_b, \text{state}) : b' = b \right]$$

for $(\mathcal{A}_K, \mathcal{O}_1, \mathcal{O}_2) = \left\{ (\text{CPA}, \hat{\text{Enc}}, \hat{\text{Enc}}), (\text{CCA1}, \hat{\text{Enc}}, \hat{\text{Dec}}), \hat{\text{Dec}}) \right\}$. The definition of wqIND-qCPA was discussed in [GHS16] Definition 3.1] and [Gag17] Definition 5.26. Here, the classical description of a quantum state $\rho$, $\text{Dsc}(\rho)$, is a bit string describing a quantum circuit which outputs $\rho$. The quantum probabilistic polynomial-time algorithm $\text{Qbd}$ receives a classical description of a quantum state and outputs the quantum state $\rho$, i.e., $\rho \xleftarrow{\$} \text{Qbd}(\text{Dsc}(\rho))$. This procedure models the situation where the adversary is familiar with the message that is encrypted but the message is not generated by the adversary himself. By doing so, it prevents the adversary from generating entanglement of the plaintext with other registers.

**Definition 4.3 (qIND-qATK for $\Pi_{sym}$).** For $\Pi \in \{\text{CPA}, \text{CCA1}\}$, a symmetric encryption scheme $\Pi_{sym}$ is said to be qIND-qATK secure if the advantage of any quantum probabilistic polynomial-time adversary $A = (\mathcal{A}_M, \mathcal{A}_D)$, where $\mathcal{A}_M$ and $\mathcal{A}_D$ are a message generator and a distinguisher, respectively, winning the game is negligible.

$$\text{Adv}_{\mathcal{A}, \Pi_{sym}}^{\text{qIND-qATK}}(\lambda) := 2 \cdot \text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{qIND-qATK}} - 1 = \text{negl}(\lambda),$$

where $\text{Succ}_{\mathcal{A}, \Pi_{sym}}^{\text{qIND-qATK}}$ is as follows:

$$\Pr \left[ k \xleftarrow{\$} \text{KeyGen}(1^\lambda); (\mathcal{D}_1, \mathcal{D}_2) \xleftarrow{\$} \mathcal{A}_M^\Pi; b \xleftarrow{\$} \{0,1\}; c_b \xleftarrow{\$} \hat{\text{Enc}}_b(\mathcal{D}_1); \text{Tr}_{1-b}(\mathcal{D}_2); b' \xleftarrow{\$} \mathcal{A}_D^\Pi(c_b, \text{state}) : b' = b \right]$$

for $(\mathcal{A}_K, \mathcal{O}_1, \mathcal{O}_2) = \left\{ (\text{CPA}, \hat{\text{Enc}}, \hat{\text{Enc}}), (\text{CCA1}, \hat{\text{Enc}}, \hat{\text{Dec}}), \hat{\text{Dec}}) \right\}$. The definition of qIND-qCPA was discussed in [BZ15] Definition B.1], [GHS16] Definition 3.2], and [Gag17] Definition 5.26. Here, tracing out, $\text{Tr}_{1-b}(\mathcal{D}_2)$, is used to discard the knowledge about non-selected state since we would like to describe a particular subsystem without having to know the overall system.
For these three definitions, the security notions for \( \Pi_{\text{sym}} \) are defined similarly as in Definition 3.2. For quantum encryption oracles, IND-qATK game uses standard transformation \( \mathcal{O}_{\text{Enc}} \), where \((\mathcal{O}'_{\text{Enc}})^\dagger \neq \mathcal{O}_{\text{Dec}} \), and (wqIND/qIND)-qATK game uses minimal transformation \( \mathcal{O}'_{\text{Enc}} \), where \((\mathcal{O}'_{\text{Enc}})^\dagger = \mathcal{O}_{\text{Dec}} \). Whether an encryption device, i.e., challenger, performs standard or minimal transformations depends on its specific architecture. Therefore, it would be sufficient to be IND-qATK secure for devices using standard transformation [GHS16].

It is worth mentioning that definition of (wqIND/qIND)-qCCA2 is not as straightforward as that of CCA2 in IND game. In the definition of IND-(CCA2/qCCA2), there was a restriction that the adversary is not allowed to query the challenge ciphertext to the decryption oracle. Otherwise, the adversary would simply decrypt the challenge ciphertext and trivially win the game. Therefore, IND-(CCA2/qCCA2) was defined by modifying the decryption oracle, in Section 2.3. The classical IND game copies the challenge ciphertext \( c_0 \) and stores it in order to compare it with decryption queries and reject forbidden queries, i.e., when \( \epsilon = c_0 \). For (wqIND/qIND)-qCCA2, however, generalization of no-cloning theorem [WZ82,Die82] restricts copying the challenge ciphertext. Also, it is unclear whether the challenger can check if \( \rho_c = \rho_{c_0} \) or not without disturbing the challenge ciphertext or the query state, due to the collapse of states after measurement [GHS16].

5 Fully-quantum Cryptography

5.1 Quantum Random Oracle Model

While a classical one-way function is based on classical infeasible mathematical problems, a quantum one-way function is provably secure by a fundamental theorem of quantum information theory [GC01]. It takes a classical bit string \( k \) as an input and outputs a quantum state \( |h_k \rangle \). The mapping \( k \mapsto |h_k \rangle \) is easy to compute and verify but impossible to invert without knowing \( k \), no matter how powerful the adversary’s computers are. More explicitly, [Hol73] showed that \( n \) qubits can give at most \( n \) bits of classical information, although qubits can carry a larger amount of classical information, i.e., the amount of classical information that can be extracted from a quantum state is limited. It should be also noted that different classical inputs may lead to the same quantum outputs due to measurement. Therefore, in order to give effective security proofs of quantum cryptographic primitives based on quantum one-way functions, quantum random oracle (QRO) is introduced in [SLL16]. It is used to realize the collision-free property, so the quantum states generated by QRO are assumed to be distinguishable by its measurement. For this model, a quantum query algorithm with \( q \) queries can be represented as follows:

\[
\text{Alg}_G := \hat{M} \hat{U}_h \hat{O}_h \cdots \hat{U}_1 \hat{O}_1 \hat{U}_0 \hat{G}.
\]

Here, \( \hat{G} \) is an input generator for an oracle, \( \hat{U}_i \)'s are fixed unitary transformations that do not depend on inputs. The (possibly) identical \( \hat{O}_h \)'s are unitary transformations that correspond to an oracle, and \( \hat{M} \) performs a measurement.

5.2 Fully-quantum Security Notions

For a quantum random oracle, the computational-quantum indistinguishability under quantum ATK (cqIND-qATK) is defined as follows.

**Definition 5.1 (cqIND-qATK for \( \Pi_{\text{sym}} \)).** For ATK \( \in \{ \text{CPA}, \text{CCA1}, \text{CCA2} \} \), a quantum symmetric encryption scheme \( \Pi_{\text{sym}} \) is said to be cqIND-qATK secure if the advantage of any quantum probabilistic polynomial-time adversary \( A = (A_M, A_D) \), where \( A_M \) and \( A_D \) are a message generator and a distinguisher, respectively, winning the game is negligible.

\[
\text{Adv}^{\text{cqIND-qATK}}_{\Pi_{\text{sym}}} (\lambda) := 2 \cdot \text{Succ}^{\text{cqIND-qATK}}_{\Pi_{\text{sym}}} - 1 = \text{negl}(\lambda),
\]

where \( \text{Succ}^{\text{cqIND-qATK}}_{\Pi_{\text{sym}}} \) is as follows:

\[
\begin{align*}
\text{Pr} \left[ k \leftarrow \text{KeyGen}(t^\lambda); (\rho_{m_0}, \rho_{m_1}, \rho_{\text{state}}) \leftarrow \mathcal{O}_M^{\mathcal{G}^1}; \\
(\rho_{x_0}, 0, 1) \leftarrow \mathcal{O}_{\text{Enc}} (\rho_{m_0}); \\
\text{Tr}_{1-b} (\rho_{\text{state}}) \leftarrow \mathcal{A}_D^{\mathcal{G}^2}; \quad b' = b \right] = \text{negl}(\lambda),
\end{align*}
\]
\((\text{ATK}, \mathcal{O}_1, \mathcal{O}_2) = \left\{(\text{CPA}, \hat{O}_{\text{QEnc}}, \hat{O}_{\text{QEnc}}) \right.\\(\text{CCA1}, (\hat{O}_{\text{QEnc}}, \hat{O}_{\text{QDec}}), \hat{O}_{\text{QEnc}})\right\} \).

The definitions of cqIND-qCPA and cqIND-qCCA1 were initially introduced in \cite{Gag17, Definition 6.6} and \cite{Gag17, Definition 6.10}, respectively. The security notions for \(\Pi_{\text{sym}}\) are defined similarly as in Definition 3.2. As already discussed in Section 4.2, cqIND-qCCA2 is not yet defined.

6 Concluding Remarks

The advent of quantum computers and algorithms has threatened the current cryptographic protocols. The cryptographic community has been motivated to establish new security notions and proof models against quantum adversaries ever since. In particular, we have reviewed previous approaches to extend the classical random oracle model to the quantum setting. Accordingly, we have introduced various indistinguishability notions under different attack models and the implication among them, as shown in Table 1. Defining (wqIND/qIND/cqIND)-qCCA2 that aptly captures the CCA2 attack scenario remains an open problem, and we leave it as future work.

Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party grant funded by the Korea government (MSIT) (No. NRF-2015R1A2A2A01006812, Security & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. NRF-2015R1A2A2A01006812, Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authentication and Security Analysis of Novel Lattice-based Fully Homomorphic Signatures Robust to Quantum Computing Attack).

References


\cite{BR00} P. O. Boykin and V. Roychowdhury. Optimal encryption of quantum bits, March 2000. \url{https://arxiv.org/abs/quant-ph/0003059}


