mathematics

IMPACT FACTOR 2.2

CITESCORE 4.6

Article

A New Hard Problem for Post-Quantum Cryptography: Q-Problem Primitives

Mostefa Kara, Mohammad Hammoudeh and Sultan Alamri

MDPI

https://doi.org/10.3390/math13152410

mathematics

Article

A New Hard Problem for Post-Quantum Cryptography: Q-Problem Primitives

Mostefa Kara [1,*], Mohammad Hammoudeh [2] and Sultan Alamri [3]

[1] Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; mohammad.hammoudeh@kfupm.edu.sa
[3] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; salamri@seu.edu.sa
[*] Correspondence: mostefa.kara@kfupm.edu.sa

Abstract

This article investigates the Q-Problem, a novel theoretical framework for post-quantum cryptography. It aims to redefine cryptographic hardness by moving away from problems with unique solutions toward problems that admit multiple indistinguishable preimages. This shift is motivated by the structural vulnerabilities that quantum algorithms may exploit in traditional formulations. To support this paradigm, we define new cryptographic primitives and security notions, including Q-Indistinguishability, Long-Term Secrecy, and a spectrum of Q-Secrecy levels. The methodology formalizes the Q-Problem as a system of expressions, called Q-expressions, that must satisfy a set of indistinguishability and reduction properties. We also propose a taxonomy of its models, including Connected/Disconnected, Totally/Partly, Fully/Partially Probabilistic, Perfect, and Ideal Q-Problem variants. These models illustrate the versatility across a range of cryptographic settings. By abstracting hardness through indistinguishability rather than solvability, Q-Problem offers a new direction for designing cryptographic protocols resilient to future quantum attacks. This foundational framework provides the foundations for long-term, composable, and structure-aware security in the quantum era.

Keywords: hard mathematical problem; Information security; perfect confidentiality; post-quantum techniques; Q-Problem

MSC: 68M25; 11T71; 68P25; 94A60

check for updates

Academic Editor: Raymond Lee

Received: 26 May 2025
Revised: 16 July 2025
Accepted: 25 July 2025
Published: 26 July 2025

Citation: Kara, M.; Hammoudeh, M.; Alamri, S. A New Hard Problem for Post-Quantum Cryptography: Q-Problem Primitives. *Mathematics* **2025**, *13*, 2410. https://doi.org/10.3390/math13152410

1. Introduction

Post-quantum cryptography seeks to develop cryptographic systems that remain secure against quantum adversaries. The advent of quantum algorithms, e.g., Shor's algorithm for factoring and discrete logarithms and Grover's algorithm for unstructured search [1,2], poses a significant threat to many classical cryptographic schemes [3]. In response, researchers proposed new schemes based on assumptions believed to be quantum-resistant, including lattice-based [4], code-based [5], multivariate polynomial [6], and hash-based cryptography [7].

However, a common trait shared by most post-quantum problems is that they ultimately aim to protect a unique hidden solution. This structural assumption, while seemingly harmless, may eventually be exploited by future quantum techniques. For example, in certain code-based settings, even if the decoding problem is computationally hard, the existence of a unique solution could facilitate distinguishability or targeted inversion under quantum models [5,8,9]. Thus, post-quantum security may be compromised not by current algorithms, but by structural weaknesses in the underlying problem formulations.

To address this, we propose a new theoretical direction: the Q-Problem (QP) paradigm. Rather than protecting a single solution, QP defines problems where multiple valid preimages exist for a given instance, and the true one is computationally indistinguishable from the others. The goal is to frustrate quantum search strategies not by sheer hardness, but by deliberately removing uniqueness. QP thus introduces a new class of cryptographic hardness assumptions that shift the adversary's challenge from solving to selecting, disrupting the very structure quantum algorithms are designed to exploit.

The Q-Problem framework leads to a family of new primitives and security notions suited to long-term, post-quantum resilience. These include Q-Indistinguishability (Q-IND): a hardness notion based on indistinguishable preimage sets. LTS: a model for protecting information across decades. Q-Secrecy levels such as $Q_x nI$, quantifying degrees of entropy and indistinguishability.

Furthermore, QP enables a taxonomy of the types of theoretical problems (Table 1): connected/disconnected QP, fully/partially QP, fully/partially probabilistic QP, perfect QP, and ideal QP, allowing tailored cryptographic designs in various application scenarios [10].

Table 1. Notations table.

Notation	Description
QP	Q-Problem, the proposed cryptographic framework based on indistinguishable multi-solution hardness
LTS	Long-Term Secrecy, security model ensuring secrecy over extended time periods (Section 2)
Q-IND	Q-Indistinguishability, a security property requiring indistinguishable preimage sets (Section 2)
Q_{xnI}	Q-Secrecy level, a parameterized measure of indistinguishability and entropy (Section 2)
Qe	Q-Expression, a structural representation of hidden data in the QP model (Section 3)
CQP/DQP	Connected/Disconnected QP, classification of Qe(s) connectivity (Section 4)
TQP/PQP	Totally/Partly QP, classification instance(s) connectivity (Section 4)
FPQP/PPQP	Fully/Partially Probabilistic QP, classification instance connectivity (Section 4)
SQP	Deterministic QP, a QP variant with no probabilistic components (Section 4)
FQP	Perfect QP, a QP instance satisfying continuity of instance connectivity (Section 4)
IQP	Ideal QP, a theoretical limit where security is supposed to be maximal (Section 4)
MFOTP	Message-Fragmentation-based One-Time Pad, a proposed encryption scheme under QP (Section 4)
$\mathcal{P}$	The set of all valid preimages corresponding to a given Q-expression (Section 2)
$\mathcal{D}$	The set of all digital data structures (Appendix A.2)
$\star$	A binary operation used in Qe, it represents a transformation or interaction between digital structures, $\mathcal{D} \times \mathcal{D} \to \mathcal{D}$. (Appendix A.3)
$nI(o)$	The number of indistinguishable preimages of output o (Section 2)
SBC	Successive Breakdown of Components (Section 4)

In this work, we present the formal construction of the Q-Problem paradigm, exploring its theoretical foundation, key properties, and cryptographic potential. By redefining secrecy as a property of indistinguishability among many candidates, QP offers a new and complementary defense model for the quantum era, one that shifts focus from solving to surviving search. Illustrative examples of Q-expressions over various data types and operations are presented in Appendix A.4 to demonstrate the applicability of the QP framework across classical and modern digital domains.

2. Definitions

In this section, three new definitions are introduced.

2.1. Long-Term Secrecy (LTS)

We define LTS as the property whereby multiple indistinguishable solutions exist for the same system instance, such that only one corresponds to the intended (original) input. Crucially, no observer without prior knowledge can determine which preimage is correct.

Let $\mathcal{O}$ denote the output space, $\mathcal{I}$ the input space, and $f : \mathcal{I} \to \mathcal{O}$ be a function. For any output $o \in \mathcal{O}$, let $f^{-1}(o) \subseteq \mathcal{I}$ denote the set of preimages of o. Among these, we define $nI(o)$ as the number of indistinguishable preimages of o, i.e., those that are computationally or semantically indistinct from one another in the absence of additional information.

A system satisfies LTS if the indistinguishability of these preimages persists over time and the number of such indistinguishable preimages meets the criterion shown by (1).

$$\forall o \in \mathcal{O}, nI(o) \geq 2, \text{or} \mid f^{-}(o) \mid \geq 2I \tag{1}$$

If $nI(o) = 1$ for every $o \in \mathcal{O}$, then f is injective, i.e., one-to-one. This means that distinct inputs always map to distinct outputs, and hence each output corresponds to a unique input. This injectivity implies the absence of ambiguity, which contradicts the notion of indistinguishability that LTS aims to guarantee. In such a case, either immediately or in the future, an adversary could potentially compute the inverse $f^{-1}(o)$ and retrieve the original input, thereby breaking the LTS of the system. Therefore, this condition serves as a security requirement; it specifies when a system can be said to satisfy LTS, rather than asserting that all systems inherently meet it.

2.2. Q-Indistinguishability Assumption (Q-IND)

Let $\mathcal{D}$ be the set of all digital data structures, and let $\star$ be a generic operation over $\mathcal{D}$ (Equation (2)).

$$f(x, y) = x \star y \tag{2}$$

Suppose an oracle samples $x, y \in \mathcal{D}$ uniformly at random and outputs $a = f(x, y) = x \star y$. Then the Q-IND assumption states:

Given an instance $a \in \mathcal{D}$ and the operation $\star$, it is computationally infeasible to identify the correct input pair (x, y) from the set of all uniformly distributed valid pairs $(x_i, y_i)_{i=1,...n}$ satisfying $f(x_i, y_i) = a$, when the number of such solutions satisfies $|f^{-1}(a)| \geq 2I$.

This assumption does not assert that computing a valid preimage is hard, but rather that distinguishing the correct preimage among $\geq 2I$ uniformly distributed and computationally indistinguishable candidates is infeasible without auxiliary information.

To define the Q-IND experiment formally, let the set of all valid preimages that defined as shown in (3).

$$\mathcal{P}_a = \{(x_i, y_i)_{i=1,..n} \in \mathcal{D} \times \mathcal{D} \mid f(x_i, y_i) = a\} \tag{3}$$

Assume that $|\mathcal{P}_a| \geq 2I$ for some indistinguishability parameter I. For a quantitative interpretation of I, it could be related to a concrete security level (security parameter λ), e.g., $I = 2^\lambda$. The set of solutions (x_i, y_i) satisfying $x_i \star y_i = a$ can be viewed as a structured level set in $\mathcal{D} \times \mathcal{D}$, potentially exhibiting algebraic or geometric properties depending on $\star$.

Q-IND Game Q-IND$^{\mathcal{A}}(I)$ can be shown as follows:

1. The challenger samples $x, y \overset{\$}{\leftarrow} \mathcal{D}$ and computes $a = f(x, y)$.
2. The challenger computes the preimage set $\mathcal{P}_a$.
3. The index $j^* \in \{1, \ldots, |\mathcal{P}_a|\}$ is set such that $\mathcal{P}_a[j^*] = (x, y)$.
4. The challenger sends $(a, \mathcal{P}_a)$ to the adversary $\mathcal{A}_2$.
5. The adversary outputs an index $j' \in \{1, \ldots, |\mathcal{P}_a|\}$.
6. The adversary wins if $j' = j^*$.

Q-IND advantage is illustrated by (4).

$$\mathrm{Adv}_{\mathcal{A}}^{\text{Q-IND}}(I) := \left| \Pr[j' = j^* | \mathcal{A}_2(a, \mathcal{P}_a)] - \frac{1}{2I} \right| \tag{4}$$

The notation $\Pr[j' = j^* \mid \mathcal{A}_2(a, \mathcal{P}_a)]$ represents the probability that the adversary outputs the correct index $j' = j^*$, conditioned on having access to the values a and $\mathcal{P}_a$. This probability is taken over all internal randomness of both the adversary and the challenger, and the conditioning reflects the fact that the adversary's output depends on its input view.

The index j^* corresponds to the location of the true preimage (x, y) within the preimage set $\mathcal{P}_a$. Since $\mathcal{P}_a = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ contains all valid input pairs such that $f(x_i, y_i) = a$. Hence, the number of possible values that j^* can take is exactly $|\mathcal{P}_a|$. An adversary making a uniform random guess has a success probability of $1/|\mathcal{P}_a|$. This forms the basis of the Q-IND advantage definition, which captures how much better the adversary $\mathcal{A}_2$ performs compared to random guessing.

Q-IND assumption is defined by (5). We say that f satisfies Q-IND assumption if, for all probabilistic polynomial adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function $\varepsilon(I)$ such that:

$$\mathrm{Adv}_{\mathcal{A}}^{\text{Q-IND}}(I) \leq \varepsilon(I) \tag{5}$$

Thus, no quantum PPT adversary can do better than a negligible advantage over random guessing.

The Q-IND assumption provides a foundation for cryptographic schemes that rely on semantic ambiguity in inversion. It supports primitives where: (a) Multiple valid preimages exist for the same output. (b) Only one preimage is correct in context (e.g., authentication or identity binding). (c) An adversary cannot identify the correct preimage without additional distinguishing information. This aligns with the QP challenge and supports LTS under controlled ambiguity.

The concept of Q-Indistinguishability (Q-IND) is closely related to classical security notions such as Indistinguishability Under Chosen-Plaintext Attack (IND-CPA) and Chosen-Ciphertext Attack (IND-CCA). While IND-CPA and IND-CCA define adversarial games where the goal is to distinguish between encryptions of two chosen plaintexts, Q-IND focuses on the adversary's inability to identify the correct preimage among many indistinguishable valid candidates that map to the same output. In a classical setting, Q-IND introduces a stronger ambiguity by eliminating the notion of a unique solution, whereas IND-CPA and CCA still assume a single encryption of a specific message. In the quantum setting, Q-IND becomes even more significant, as it assumes hardness holds even when the adversary can query the function (or oracle) in superposition. This contrasts with quantum versions of IND-CPA/CCA, where encryption or decryption oracles are adapted to handle quantum queries. Therefore, Q-IND generalizes the notion of semantic security by enforcing indistinguishability over solution spaces rather than encrypted message pairs, offering a deeper level of resistance in both classical and quantum adversarial models.

2.3. Q-Secrecy Level $Q_x nI$

Let $x \in \{1, 2\}$ be the hardness level to find the preimages. The first level is $Q_1 nI$, it can be expressed as Easy to Find, Impossible to Distinguish (EFID). For example, $x + y \mod p = a$ is easy to resolve, i.e., to find the possible preimages, but without any additional information about x, y, it is impossible to guess the correct pair.

The second level is $Q_2 nI$; it can be expressed as Hard to Find, Impossible to Distinguish (HFID). For example, $x^y + y^x \mod p = a$ is not easy to resolve with large x, y, p ($O(p^2)$ with brute-force strategy). It is considered computationally hard, like a variant of the Diophantine problem modulo p.

The other factor of Q-Secrecy that can be integrated with Q_x is the number of solutions nI (# preimages), which can be referred to as the degree $n \geq 2$ (Equation (6)).

$$n = min(min|\mathcal{P}[f_i]|, |\mathcal{P}[S]|) \tag{6}$$

where $|\mathcal{P}[f_i]|$ denotes the number of solutions of a single equation i, and $|\mathcal{P}[S]|$ is the number of solutions of the entire system. Thus, the first example $x + y$ belongs to $Q_1 pI$ and the second $x^y + y^x$ belongs to $Q_2 pI$.

Supposing that $j > i$, of hardness level is as follows: $Q_1 iI$, ($Q_1 jI$, $Q_2 iI$), then $Q_2 jI$. The level between $Q_1 jI$ and $Q_2 iI$ depends on the scenario.

The hash function could be another example of HFID if the input length is large and unknown to the adversary, which will give him a large set of preimages, of course, in a context where he cannot determine which is the correct one. This example remains valid unless an efficient algorithm exists to calculate the hash preimages other than brute force, which is infeasible for large input sizes. Otherwise, it is EFID.

3. QP: Formal Presentation

QP is defined as follows:

$$\text{QP} \Leftrightarrow \begin{cases} \text{Let}QE : \{Qe_1, Qe_2, \ldots, Qe_n\} \text{ be the system} \\ 1 \triangleright Qe_i : \{x_i \star y_i; \text{and/or} z_i\} \text{ is a Q-expression where:} \\ \quad x_i, y_i, z_i \in \mathcal{D}, \text{ and } \mathcal{D} \text{ is the set of all digital data structures (digital objects);} \\ \quad \star \text{ is a generic operation over } \mathcal{D}; \\ \quad x_i \text{ and } y_i \text{ may be atomic or composed, } z_i\text{: atomic object} \\ 2 \triangleright Qe_1 \theta_1 Qe_2 \ldots \theta_{n-1} Qe_n = \perp \\ \text{: combining Q-expressions gives no information about } x_i \text{ and } y_i \\ 3 \triangleright \text{Given public parameter(s) } p, \forall Qe_i, (x_i, y_i) \theta p = \perp \\ 4 \triangleright \text{Given } p\text{, and instance} a_i = x_i \star y_i, \forall Qe_i, |\mathcal{P}_{a_i}[Qe_i]| \geq 2I \\ 5 \triangleright \text{Given QE, } |\mathcal{P}_{a_1, a_2, ..a_n}[QE]| \geq 2I \\ 6 \triangleright \text{Let} F_\theta : QE_n \xrightarrow{reducedto} QE'_{n'}, QE'_{n'} \models \text{QP}_{1-6} \\ 7 \triangleright \text{Only hidden objects are considered, i.e., coefficients/constants are ignored} \end{cases}$$

QE denotes the overall system or protocol instance, e.g., $Alice \xrightarrow{Qe_1} Bob$, and $Bob \xrightarrow{Qe_2} Alice$, so $QE : \{Qe_1, Qe_2\}$ must satisfy QP conditions, namely Points (1) through (6) as defined in the QP framework.

A QE is an abstract expression over digital structures with arbitrary operations, whose goal is to encode its operands using one or more forms of data hiding under formal indistinguishability constraints, which includes a wide spectrum of computational and information-theoretic methods; Qe could be an atomic digital object or composed with any meaningful operations between digital structures. Operands x and y may be an atomic digital object or a composed expression built from subcomponents via one or more operations, e.g., $x = x_1 \star x_2$, $x_1 = x_{11} \star x_{12}$, and so on.

In Point 2, $\perp$ states that the values of x_i and y_i remain hidden even with combining Q-expressions.

In Point 3, $\perp$ states that neither x nor y has any relation with the public parameter p that would reveal hidden values, and that the adversary cannot infer any confidential information about the secret x, y from $f(x, y, p)$.

$|\mathcal{P}_a[Qe]| \geq 2I$ means that $\#Sols(a = x \star y) \geq 2I$, it can be presented as: $|\mathcal{P}[Qe^a]| \geq 2I$.

$|\mathcal{P}_{a_1,a_2,\ldots a_n}[QE]| \geq 2I$ means that $\#Sols_{QE}(a_1 = Qe_1, a_2 = Qe_2, \ldots a_n = Qe_n) \geq 2I$ where a_i: are instances. This states that combining Q-expressions gives more than two indistinguishable solutions.

Point 6 in QP definition meant that the new system QE' derived (reduced) from combining Qe_i of the initial QE using operations θ must be, in turn, a QP, and so on (Equation (7)).

$$\{Qe_1\theta_1 Qe_2\theta_2 \ldots Qe_n\} \to \{Qe'_1, Qe'_2, \ldots, Qe'_{n'}\} \tag{7}$$

For instance, we put $QE : \{Qe_1^a : a = x + 2y + 1, Qe_2^b : b = x - z^2\}$, $QE' : \{Qe_1'^c : c = 2y - z^2 + 1\}$ is QP; but $QE : \{Qe_1^a : a = y^x, Qe_2^b : b = z^x, Qe_3^c : c = yz\}$, $QE' : \{Qe'_1 : c, Qe'_2 : c^x\}$ it is not QP because the secret variable x can be retrieved using a quantum computer.

To evaluate a Qe, only hidden objects are considered, i.e., coefficients/constants are ignored. For example, $2x, x + 1, x^2, 5hash(x), 4||x$ are elementary (atomic).

In practice, there are some conditions that must be applied to ensure the safety of Q-IND. Operation $\star$ must be free of trapdoor patterns, e.g., for multiplication $x \cdot y = a$, the adversary could try all divisors of a; if any structure leaks, the assumption might fail. Side channel attacks must be taken into account; if an adversary can learn partial bits of the inputs, e.g., via timing, Q-IND can be broken. There should not be a correlation with the external state, i.e., inputs must appear random even if the adversary knows message/ciphertext mappings.

4. QP: Models, Quantitative View, and Use Cases

We define five models of QP.

4.1. Connected and Disconnected QP (CQP, DQP)

CQP, DQP are illustrated in (8)

$$\begin{cases} \text{CQP} : \forall i, j Qe_i \cap Qe_j \neq \emptyset \\ \text{DQP} : \forall i, j Qe_i \cap Qe_j = \emptyset \end{cases} \tag{8}$$

In CQP, there exists at least one common elementary secret object shared between two or more Qe in QE. Consider these examples:

$$\begin{cases} \text{CQP} : Qe_1 : x + y \mod p = a, Qe_2 : x + z \mod p = b, \text{so } Qe_1 \cap Qe_2 = \{x\} \\ \text{DQP} : Qe_1 : x + y \mod p = a, Qe_2 : z + v \mod p = b, \text{so } Qe_1 \cap Qe_2 = \emptyset \end{cases}$$

Note that the parameter p is not considered a common object. Therefore, if $Qe_i \cap Qe_j = \{\alpha\}$ where α is public and does not contain a secret object, that does not affect the system being DQP.

4.2. Totally and Partly QP (TQP, PQP)

If CQP/DQP is about two different Qe (vertical change Qe_1 and Qe_2), TQP/PQP is about the same Qe (horizontal change Qe_1^1 and Qe_1^2), but two different inputs. Suppose that their corresponding outputs are $a_1 = x_1 \star y_1, a_2 = x_2 \star y_2$ (Equation (9)).

$$\begin{cases} \text{TQP} : \forall i \text{ for } Qe_i^{1,2}, x_1 \neq x_2 \text{ and } y_1 \neq y_2 \\ \text{PQP} : \text{otherwise} \end{cases} \tag{9}$$

Consider this example: k is the secret key for encryption, m is the plaintext. Suppose that the input is (k, m).

$$\begin{cases} Qe_1^1 : c_1 = km_1, Qe_2^1 : h_1 = hash(m_1) \\ Qe_1^2 : c_2 = km_2, Qe_2^2 : h_2 = hash(m_2) \end{cases}$$

In Qe_1, note that m (represents y) has changed but k (represents x) does not, i.e., $x_1 = x_2$. Therefore, this QE is PQP and not TQP.

4.3. Fully and Partially Probabilistic QP (FPQP, PPQP)

If TQP/PQP is a horizontal change for two different inputs, FPQP/PPQP is a horizontal change for the same input. Suppose that the corresponding outputs are $a = x \star y, a' = x' \star y'$ (Equation (10)).

$$\begin{cases} \text{FPQP} : \forall i \text{ for } Qe_i^{a,a'}, x \neq x' \text{ and } y \neq y' \\ \text{PPQP} : \text{otherwise} \end{cases} \tag{10}$$

Consider this example: k is the secret key for encryption, m is the plaintext, and r is a fresh random number. Suppose that the input is (k, m).

$$\begin{cases} \text{PPQP} : Qe_1^c : c = km + r_1, Qe_1'^{c'} : c' = km + r_2 \\ \text{FPQP} : Qe_1^c : c = m + r_1k_1 + r_2k_2, \\ Qe_1'^{c'} : c' = m + r_3k_1 + r_4k_2 \end{cases}$$

Note that in PPQP, where $x = km$ and $y = r$, y has changed but x has not in the two instances Qe_1 and Qe_1' for the same input m. In contrast, in FPQP where $x = (m + r_1k_1)$, $y = r_2k_2$, both x and y have changed ($x' = (m + r_3k_1), y' = r_4k_2$). This scheme is similar to Gentry's scheme, $c = m + r2 + qp$, where r, q are random.

In deterministic QP (SQP), $a = a'$, the same output for the same input. For operations that are not repetitive, such as generating a public key, it is sufficient for x and y to be unknown.

We emphasize that if a system satisfies $FPQP$ or $PPQP$, then it necessarily satisfies TQP or PQP, respectively.

4.4. Perfect QP (FQP)

An FQP is a fully probabilistic QP (FPQP) such that, under Successive Breakdown of Components (SBC), each resulting part remains an FPQP, down to the final, indivisible Qe (elementary, atomic, or non-decomposable Qe).

$$\text{FQP} : SBC(FPQP) \to FPQP \tag{11}$$

Let Q_e be a QP expression defined as $Q_e := x_1 \star x_2 \star \cdots \star x_n$ where each $x_i \in \mathcal{D}$. As shown in (11), SBC is a recursive process that transforms Q_e into a sequence of finer-grained expressions by recursively applying structural expansions to one or more components x_i based on their internal structure, subject to the following:

1. Initial Step: Let $Q_e^{(0)} = Q_e$.

2. Recursive Step: For each $k \geq 0$, the k-th breakdown produces a new expression $Q_e^{(k+1)}$ by replacing at least one component $x_i^{(k)} \in Q_e^{(k)}$ with a more granular expression: $x_i^{(k)} \to x_{i1}^{(k+1)} \star x_{i2}^{(k+1)} \star \cdots \star x_{im}^{(k+1)}$ where each $x_{ij}^{(k+1)} \in \mathcal{D}$.

3. Termination Condition: The process continues until all components are elementary (atomic), meaning they are no longer decomposable within the domain $\mathcal{D}$.

We denote the final expression by $Q_e^{(\ell)}$, where ℓ is the number of breakdown steps required to reach a fully resolved form $Q_e^{(\ell)} : x_1^{(\ell)} \star x_2^{(\ell)} \star \cdots \star x_m^{(\ell)}$ (elementary form).

In the context of Perfect QP (FQP), each intermediate expression $Q_e^{(k)}$ must remain a valid FPQP throughout the SBC process.

Consider the precedent example of $FPQP : Qe^c : c = m + r_1k_1 + r_2k_2$. If we put $x = (m + r_1k_1)$, $y = r_2k_2$, the first SBC on x gives $x_1 = m$, $x_2 = rk$, which is only PPQP. Therefore, this QE is not FQP.

Again, consider the precedent example of $PPQP : Qe^c : c = km + r$, so $x = km$ and $y = r$. Suppose now that the same plaintext m and the encryption secret key k have intervals. When calculating c, variables m and k are selected at random from their intervals. The original plaintext can be retrieved as: $m = c \div k^*$ where $\div$ denotes the integer division, k^* is a constant secret key, and r is a random number less than k^*. An illustrative numerical example could be: $m_1 \in [1,4], m_2 \in [5,8]$, etc., $k \in [1010, 1020]$, $k^* = 1000$. Then, $c_1 = 3 \times 1012 + 22 = 3056$, $c_1' = 4 \times 1015 + 107 = 4167$, and $m = 3056 \div 1000 = 3$, $m' = 4167 \div 1000 = 4$. Subexpression y is atomic, the SBC on x is $x_1 = m, x_2 = k$. Note that $x_1 \neq x_1'$ and $x_2 \neq x_2'$ for the same plaintext. On the other hand, each elementary Q-expression is changed for two instances of the same input. Accordingly, this scheme is FQP.

4.5. Ideal QP (IQP)

IQP is $DFQP$, which means a Disconnect and a Perfect QP. An example of IQP is introduced: a Message-Fragmentation-based OTP encryption scheme (MFOTP). In this scheme, $c_m = (c_1, c_2) = (m_1 \oplus k_1, m_2 \oplus k_2)$ where m_1 and m_2 are two random fragments of m, $k_1 \neq k_2$ even if the same plaintext m is encrypted again.

Why is the suggested MFOTP an IQP?

Let QE be a system, $QE : \{Qe_1, Qe_2\}$, $Qe_1^{c_1} : c_1 = m_1 \oplus k_1, Qe_2^{c_2} : c_2 = m_2 \oplus k_2$, this is presented as $x \oplus y$. Both x and y are hidden, $\mid \mathcal{P}_c \mid \geq 2I$ is satisfied due to the existence of many indistinguishable (m, k) where $m \oplus k = c$. Thus, QE is QP.

Elements x and y are atomic, and $(m_1 \oplus k_1) \cap (m_2 \oplus k_2) = \emptyset$, so QE is DQP. For two inputs m_1, m_2: $m_1 : Qe_{11}^{c_{11}} : c_{11} = m_{11} \oplus k_{11}, Qe_{12}^{c_{12}} : c_{12} = m_{12} \oplus k_{12}$; and $m_2 : Qe_{21}^{c_{21}} : c_{21} = m_{21} \oplus k_{21}, Qe_{22}^{c_{22}} : c_{22} = m_{22} \oplus k_{22}$.

Noting that $\forall i \; for \; Qe_i, x_{m_1} \neq x_{m_2}$ and $y_{m_1} \neq y_{m_2}$, so QE is TQP. In addition, for a new m, a new fragmentation will be used, i.e., $m_{ij} \neq m_{ij}'$, and new keys are used ($k_{ij} \neq k_{ij}'$). Every object in QE is changed even with the same input; therefore, this scheme is FPQP. Now, MFOTP is DQP and FPQP, which means that MFOTP is an IQP.

4.6. OTP and Q-Problem

It is known that in OTP, a one-time encryption key is used. For example, if $Enc(m) : c = m \oplus k_1$ and $c' = m \oplus k_2$ for the same input m, we find that $x = x' = m$. Therefore, the basic version of OTP is not fully probabilistic but a partially probabilistic QP.

To convert it from PPQP to FPQP, random fragmentation of the message m into two parts can be used. Thus, the first encryption of m is $c = (m_1 \oplus k_1, m_2 \oplus k_2)$, the second encryption of m is $c = (m'_1 \oplus k'_1, m'_2 \oplus k'_2)$. That gives $x = m_1 \neq x' = m'_1$ and $y = k'_1 \neq y' = k'_1$, the same thing for the second part of m i.e., $x = m_2 \neq x' = m'_2$ and $y = k_2 \neq y' = k'_2$. Therefore, using two keys for each message instead of one is needed. Since it is easy to obtain valid solutions (m, k) in $c = m \oplus k$ for a given c, basic OTP and MFOTP are Q_1nI. Figure 1 summarizes the presented Q-Problem models.

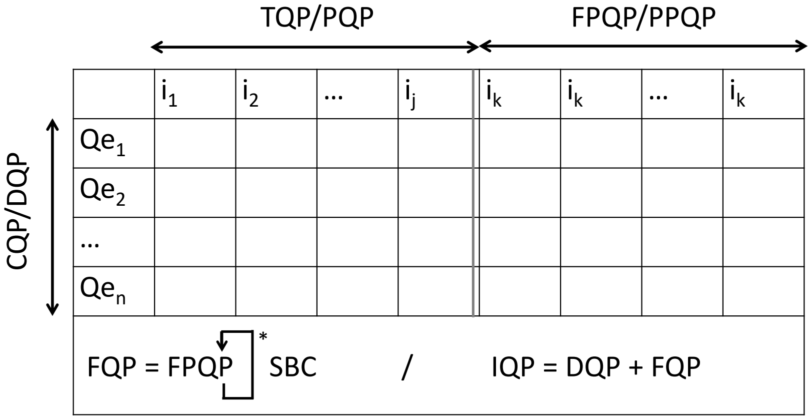

Figure 1. Illustration of Q-Problem taxonomy, "*" indicates repeating the SBC operation.

4.7. Quantitative View of QP Models

Although the QP taxonomy is defined structurally, each class can be approximated quantitatively in terms of the number of indistinguishable preimages ($|\mathcal{P}_a[Qe]|$) and the associated adversarial advantage. For example, in an Ideal QP (IQP), each output is derived from a disconnected and perfectly decomposable expression, producing a large set of indistinguishable preimages and minimizing the adversary's ability to recover the true origin. In contrast, classes such as PPQP or PQP may provide only partial variation or structural overlap. Table 2 summarizes the distinctions across QP levels, allowing practitioners to select appropriate configurations based on entropy goals and attack models.

Table 2 shows a qualitative and quantitative comparison of QP model classes, where $x_1 > x_2 > x_3 > x_4 > x_5 > x_6$ and $y_1 > y_2$ are entropy estimates for indistinguishable sets, and Adv. success $\leq 1/2^{x_i}$ (respectively, $\leq 1/2^{y_i}$). We can not compare CQP/DQP with other classes because CQP/DQP is inter-Qe and the others are intra-Qe, except IQP, which relies on two dimensions.

Table 2. Qualitative and quantitative comparison of QP model classes.

QP Class	Distinction Type	$\lvert\mathcal{P}_a[Qe]\rvert$	SBC
IQP	Disconnected + Perfect	2^{x_1}	Fully
FQP	Fully Decomposable FPQP	2^{x_2}	Fully
FPQP	Probabilistic over same input	2^{x_3}	Partially
PPQP	Partial randomness (same input)	2^{x_4}	Varies
TQP	Different inputs, all parts vary	2^{x_5}	N/A
PQP	Different inputs, partial overlap	2^{x_6}	N/A
DQP	Qe_i are disjoint	2^{y_1}	N/A
CQP	Qe_i share at least one term	2^{y_2}	N/A

4.8. QP Use Cases

This point explains how the belonging of schemes to QP is analyzed.

Let us take RSA, $c = m^e \mod n$ where e is the public key. Since values c and e are given, and the only hidden variable is m, RSA's expression is in the form of x^y where y is known, so $x^y = a$ has only one solution. Therefore, point 4 in the formal definition is not satisfied, and RSA is not QP. Furthermore, e depends on n where $e \times d \equiv 1 \mod \phi(n)$.

Let us take ElGamal, $c = (m \times h^r, g^r)$ where $h = g^k$ is the public key and k is the secret key. The first part $m \times h^r$ belongs to QP, $x = m$ (unknown) and $y = h^r$ (unknown because r is hidden). Since g is public, the expression of the second part g^r is in the form of x^y where x is known, so ElGamal does not belong to QP.

Let us take Gentry's FHE scheme (DGHV) that is written as $c = m + 2r + qp$ where r and q are random for each encryption and p is a secret key. It can be considered that this scheme is QP, if we put $x = m + 2r$ and $y = qp$, then $c = x + y$, this form verifies all conditions of QP. Since it has only one Qe, it is DQP; for two different inputs, x, y will be changed, which makes it TQP; for two different instances of the same input, x, y will be changed, so it is FPQP; we note that not each elementary variable, e.g., m, will be changed for two instances of the same input, so it is not FQP; consequently, the scheme is not IQP. Given c, the possible solutions in $c = m + 2r + qp$ are easy to find, so Gentry's scheme is $Q_1 nI$.

As for Kyber-NIST, Key Encapsulation Mechanism, the public key is (A, t) where $t = As + e$, (s, e) is the secret key. The encapsulation function computes $(u, v) = (Ar + e_1, tr + e_2)$ where r, e_1, r_2 are random. The system $QE : Qe_1 : As + e, Qe_2 : Ar + e_1, Qe_3 : tr + e_2$ satisfies QP definition, for example, s, e are hidden in Qe_1, r, e_1 are hidden in Qe_2, r, e_2 are hidden in Qe_3. We note that $Qe_1 \cap Qe_3 = \{e\}$, and $Qe_2 \cap Qe_3 = \{r\}$, so it is not DQP but CQP. To check TQP/PQP (respectively, FPQP/PPQP, FQP), Qe_1 is not considered because it concerns the generation of the public key, and it is computed once. If we put $x = Ar, y = e_1$ (respectively, tr, e_2), terms x, y will be changed for two different inputs, so it is TQP, also for two different instances of the same inputs (new encryption of the same plaintext), so it is FPQP. Kyber is not FQP because not every elementary variable will be changed. After SBC, $x_1 = x'_1 = A$ (or t), for two instances, does not change. Consequently, Kyber is not IQP. Given t in $t = As + e$ and (u, v) in $(u = Ar + e_1, v = tr + e_2)$, the possible solutions are easy to find, so Kyber is $Q_1 nI$.

The same thing for FrodoKEM-NIST, Key Encapsulation Mechanism, it uses the public key (A, B) where $B = As + e$, the encapsulation is: $(B', C) = (As' + e', B^T s' + e'')$.

In Dilithium-NIST, Digital Signature Algorithm, the public key is (A, t) where $t = As_1 + s_2$, (s_1, s_2) is the secret key. The signature function computes $(z, r) = (y + Hash(Ay, m)s_1, Hash(Ay, m)s_2)$ where y is random.

The system, $QE : Qe_1 : As_1 + s_2, Qe_2 : y + H(Ay, m)s_1, Qe_3 : H(Ay, m)s_2$, satisfies QP definition. We note that $Qe_1 \cap Qe_2 = \{s_1\}$, and $Qe_1 \cap Qe_3 = \{s_2\}$, so it is not DQP but CQP. The common variable A is not considered because it is public. In Qe_3, we put $y = s_2$; y will not be changed for two different inputs, so the scheme is PQP, also for two different instances of the same inputs, so it is PPQP. Consequently, Dilithium is not FQP nor IQP. With given (z, r) in $(z = y + Hash(Ay, m)s_1, r = Hash(Ay, m)s_2)$, the possible solutions for (m, y, s_1, s_2) are not easy to find, so Dilithium is $Q_2 nI$.

As for McEliece-NIST public-key encryption, the public key is G' where $G' = SGP$, (S, G, P) is the secret key. The encryption function computes $c = mG' + e$ where e is random. The system is $QE : Qe_1 : SGP, Qe_2 : mG' + e$. We note that $Qe_1 \cap Qe_2 = \{G'\}$, so the system is not DQP but CQP. In Qe_2, we put $x = mG', y = e$; x, y will be changed for two different inputs, so it is TQP, not for two different instances of the same inputs, so it is PPQP. In SBC, we put $x_1 = m, x_2 = G'$; G' will not be changed for two instances of the same input, so it is not FQP; consequently, McEliece is not IQP. With a given (G', c) in $(G' = SGP, c = mG' + e)$, the possible solutions for (S, G, P, m, e) are easy to find, so Dilithium is $Q_1 nI$.

Table 3 summarizes this analysis and classification.

The QP framework establishes a novel structural foundation based on indistinguishability of preimages, aiming primarily to resist quantum adversaries. However, QP compliance, defined by satisfying the indistinguishability conditions such as Q-IND, does not automatically imply security against classical attacks. For instance, the expression $c = m \cdot k \bmod p$ may satisfy the QP requirement of having multiple indistinguishable (m, k) pairs that map to the same ciphertext c. However, this basic scheme is vulnerable to classical attacks, such as known-plaintext or chosen-plaintext attacks, especially if the key k is reused or low-entropy inputs are involved. In such cases, an adversary can trivially recover m or k by computing modular inverses or exploiting statistical patterns. This illustrates that not every QP-compliant scheme is secure against classical threats. As in traditional cryptography, QP-based designs must be fortified with adequate primitives and resistance to adaptive attacks to achieve robust classical and quantum security. Therefore, while QP offers a strong post-quantum abstraction, practical schemes must still undergo rigorous classical cryptanalysis.

Table 3. Classification of cryptographic schemes under QP. FQP is FPQP*, and IQP is DQP and FQP.

Scheme	QP	CQP, DQP	TQP, PQP	FPQP, PPQP	FQP	IQP	Degree
RSA	N	-	-	-	-	-	-
ElGamal	N	-	-	-	-	-	-
Gentry's FHE	Y	DQP	TQP	FPQP	N	N	$Q_1 nI$
Kyber	Y	CQP	TQP	FPQP	N	N	$Q_1 nI$
FrodoKEM	Y	CQP	TQP	FPQP	N	N	$Q_1 nI$
Dilithium	Y	CQP	PQP	PPQP	N	N	$Q_2 nI$
McEliece	Y	CQP	TQP	PPQP	N	N	$Q_1 nI$
OTP	Y	DQP	TQP	PPQP	N	N	$Q_1 nI$
MFOTP	Y	DQP	TQP	FPQP	Y	Y	$Q_1 nI$

5. Conclusions

This article introduced the QP as a new theoretical framework in cryptography and privacy-preserving, centered on the definition of novel primitives and security notions such as Q-IND and LTS. The formalism captures how digital expressions can hide information structurally, across a wide class of operations and data types, by enforcing indistinguishability as a foundational property. As an initial demonstration, we focused on specific "expression forms, data structure, and operations", illustrating the core components of the framework. While this version did not explore the full diversity of digital data structures, QP lays the groundwork for future exploration.

Although the present work focuses on classical digital structures, extending the QP framework to operate directly on quantum states, such as qubits, remains an open direction. Such an extension would require redefining Q-expressions in the context of quantum information theory and exploring indistinguishability under quantum superposition and entanglement.

Author Contributions: Conceptualization: M.K.; methodology: M.K., and M.H.; validation: M.K., M.H., and S.A.; formal analysis: M.K., and S.A.; writing—original draft preparation: M.K.; writing—review and editing: M.K., M.H., and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: The study is supported by the grant no. CRPG-25-3092 under the Cybersecurity Research and Innovation Pioneers Grant, provided by the National Cybersecurity Authority NCA, in the Kingdom of Saudi Arabia.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We gratefully acknowledge the support of the National Cybersecurity Authority (NCA), Kingdom of Saudi Arabia, through the Cybersecurity Research and Innovation Pioneers Grant.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Digital Structures, Operations, and Examples in the QP Framework

Appendix A.1. General Forms of Quantum Expressions (QEs)

Each Q-expression $Qe \in QE$ is defined as a data-hiding transformation of the form $Qe = \{x \star y, z\}$, where $x, y, z \in \mathcal{D}$, and $\star$ is an operation over digital structures. The design of Qe can leverage a wide range of hiding mechanisms, generalized as follows:

- Encrypted QEs: data are transformed using a cryptographic key to ensure confidentiality.
- Obfuscated QEs: program or function logic is restructured to preserve behavior but hide structure.
- Transformed QEs: inputs are encoded, compressed, reshaped, or structurally reformatted.
- Masked/Blinded QEs: sensitive data are hidden via randomness or masking terms.
- Anonymized QEs: identifiable attributes are removed or abstracted to preserve privacy.
- Homomorphic QEs: enables operations on encoded data without revealing underlying inputs.
- Steganographic QEs: hidden data are embedded within unrelated digital carriers.
- Zero-Knowledge QEs: encodes a proof of validity without revealing the underlying secret.

Appendix A.2. The Domain of Digital Structures $\mathcal{D}$

Let $\mathcal{D}$ denote the set of all digital data structures that may serve as operands in a Qe. Elements of $\mathcal{D}$ can include the following categories:

- Sequential: strings, arrays, binary files
- Hierarchical: JSON, XML, trees, nested objects
- Graph-based: knowledge graphs, dependency graphs, neural networks
- Tabular: relational databases, CSV files, matrix tables
- Geometric: vectors, meshes, CAD models, coordinate structures
- Encoded Media: JPEG, PNG, MP3, MP4, and other compressed formats
- Encrypted Formats: ciphertexts, MACs, key blobs, wrapped keys
- Executable Structures: bytecode, compiled binaries, interpretable code blocks

This abstraction enables QP to capture a wide variety of data types relevant to cryptographic modeling and information hiding.

Appendix A.3. Examples of Operations $\star$ over Digital Structures

The binary operation $\star$, used in Qe as $x \star y$, represents a transformation or interaction between digital structures $\star : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$. It serves as a structural and/or cryptographic combiner that binds two elements into a secure, indistinguishable form. It supports key properties like obfuscation, composability, and preimage ambiguity in the system. Examples of such operations include:

- Bitwise Operations: AND, OR, XOR, NOT, shift left/right, rotate
- Arithmetic Operations: addition, subtraction, multiplication, division, modulo, exponentiation
- Logical/Boolean Operations: conjunction, disjunction, implication, equivalence
- Structural Operations: concatenation, slicing, padding, alignment, encoding

- Cryptographic Primitives: hashing, encryption/decryption (symmetric/asymmetric), digital signatures, MACs
- Information-Theoretic Operations: entropy measures, compression, error correction
- Machine Learning-Related Operations: tensor reshaping, embedding transformations, model-layer mappings
- Data Structure Operations: graph traversal, tree pruning, set union/intersection
- Mathematical Transforms: logarithm, discrete Fourier transform, matrix multiplication, normalization
- Specialized/Domain-Specific: image convolution, audio mixing, geometric transformations, natural language tokenization

These categories illustrate the generality and expressive power of the $\star$ operator within QEs, allowing the QP framework to apply across domains ranging from classical computation to modern AI systems.

Appendix A.4. Illustrative Qe Use Cases Across Digital Structures

This appendix presents some representative use cases where the QP framework models scenarios beyond the capabilities of traditional cryptographic paradigms, using diverse digital structures and operations defined in the paper.

1. Obfuscated Bytecode Fragment: Data type: executable bytecode. Operation: code obfuscation. Transmitting a protected license-checking routine. The logic is embedded in obfuscated bytecode, defined as:

$$Qe : \text{Obfuscate}(f(x)) = a$$

where x is the secret license, f is the checking logic, and a is the resulting bytecode. Q-IND: the QP framework ensures that even if the adversary decompiles a, multiple valid preimages of f exist (e.g., semantically equivalent but syntactically distinct logic trees), making the original logic indistinguishable. QP models indistinguishability over code semantics, not just ciphertexts. Traditional encryption or obfuscation fails to formalize such structural ambiguity.

2. Steganographic Image Transmission: Data type: encoded media (image). Operation: LSB steganographic embedding. Hiding a secret message x inside an image y using least significant bit (LSB) encoding:

$$Qe : \text{Embed}_{\text{LSB}}(x, y) = a$$

The output a is visually identical to y, but carries hidden content. Q-IND: the set of all valid messages that could be embedded into a is large. The Q-IND notion captures the indistinguishability of the true x among all plausible message preimages. Unlike traditional steganography, Q-IND formally ensures that the adversary cannot computationally distinguish the actual message from others.

3. Chained Hashing and Format-Preserving Encryption: Data type: structured strings (e.g., credit card numbers, UUIDs). Operations: hashing $\circ$ format-preserving encryption. Storing credit card tokens by applying a multi-stage transformation. The expression is modeled as:

$$Qe : \text{FPE}(\text{Hash}(x \| r)) = a$$

where x is the secret plaintext (e.g., a credit card number), r is a session-specific salt or nonce, $\text{Hash}(x \| r)$ is cryptographic hash to destroy structure, $\text{FPE}(\cdot)$ is format-preserving encryption to retain expected format, a is the final token (e.g., 16-digit numeric string). Q-IND: a has multiple indistinguishable preimages.

4. Logic-Based Condition Masking: Data type: policy rules. Operations: $\wedge, \Rightarrow$, and $\equiv$. Encoding multiple access policies using Q-expressions:

$$Qe_1 : ((x_1 \wedge x_2) \Rightarrow y_1)$$
$$Qe_2 : ((x_3 \Rightarrow x_4) \equiv y_2)$$

Q-IND: many logically equivalent forms lead to the same behavior a_i, and their union provides no leakage beyond behaviorally equivalent systems.

5. Layered Tensor Transformation: Data type: input tokens and image features. Operations: reshape, embed, and map. Consider two Q-expressions from different input sources:

$$Qe_1 : \text{LayerMap}_1(\text{Embed}(\text{Reshape}(x_1))) \rightarrow a_1$$
$$Qe_2 : \text{LayerMap}_2(\text{Embed}(\text{Reshape}(x_2))) \rightarrow a_2$$

Q-IND: the adversary cannot determine the exact path (reshape + embed + layer map) or even whether $x_1 = x_2$, due to the multi-layer indistinguishability and compositional entropy.

6. Graph Query Obfuscation: Data Type: graph nodes and relations. Operations: traverse, filter, and intersection. Suppose two separate queries are issued over a graph:

$$Qe_1 : \text{Filter}_1(\text{Traverse}(x_1)) \cap y_1 \rightarrow a_1$$
$$Qe_2 : \text{Filter}_2(\text{Traverse}(x_2)) \cap y_2 \rightarrow a_2$$

Q-IND: the adversary sees both output sets, but cannot infer the origin nodes x_i or filters applied. Graph redundancy and non-injective traversal allow multiple plausible preimage paths for each a_i.

7. Multi-Modal Privacy Workflow: Data types: structured logs, images, and graphs. Operations: logical masking, pixel transformation, and graph pruning. A privacy-preserving analytics system processes diverse input types using separate, specialized Q-expressions:

$$\begin{aligned} Qe_1 &: (\text{flag} \wedge \neg\text{status}) \Rightarrow \text{anomaly}, && \text{boolean condition over log data} \\ Qe_2 &: \text{Downscale}(\text{Scramble}(\text{Crop}(x))), && \text{composed image filter} \\ Qe_3 &: \text{Prune}(\text{Traverse}(G), \text{degree} < 3), && \text{graph-based operation} \end{aligned}$$

Each Q-expression is applied to a distinct modality: a structured log entry, a surveillance image, and a social or sensor graph. Qe_1 expresses a logic-obfuscated policy; Qe_2 hides image structure through a composed filter; Qe_3 obscures graph topology via traversal + pruning. Q-IND: the preimage sets $\mathcal{P}_{a_i}[Qe_i]$ are large and indistinguishable.

These examples demonstrate how the QP framework unifies a wide range of hiding mechanisms under a single indistinguishability-based model. Whether applied to logic obfuscation, covert channels, or data anonymization. While some of these use cases can also be addressed using traditional cryptographic primitives, e.g., masking and steganography, the QP framework provides a unified formalism based on preimage indistinguishability. This allows one to reason uniformly across various data-hiding methods, especially under post-quantum threat models that exploit structural uniqueness or algebraic predictability.

References

1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
2. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.

3. Aljumaiah, O.; Jiang, W.; Addula, S.R.; Almaiah, M.A. Analyzing cybersecurity risks and threats in IT infrastructure based on NIST framework. *J. Cyber Secur. Risk Audit.* **2025**, *2025*, 12–26. [CrossRef]
4. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **2009**, *56*, 1–40. [CrossRef]
5. Malygina, E.S.; Kutsenko, A.V.; Novoselov, S.A.; Kolesnikov, N.S.; Bakharev, A.O.; Khilchuk, I.S.; Shaporenko, A.S.; Tokareva, N.N. Post-quantum cryptosystems: Open problems and current solutions. Isogeny-based and code-based cryptosystems. *J. Appl. Ind. Math.* **2024**, *18*, 103–121. [CrossRef]
6. Ding, J.; Petzoldt, A.; Schmidt, D.S. Multivariate cryptography. In *Multivariate Public Key Cryptosystems*; Springer: New York, NY, USA, 2020; pp. 7–23.
7. Bernstein, D.J.; Hopwood, D.; Hülsing, A.; Lange, T.; Niederhagen, R.; Papachristodoulou, L.; Schneider, M.; Schwabe, P.; Wilcox-O'Hearn, Z. SPHINCS: Practical stateless hash-based signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 368–397.
8. Bavdekar, R.; Chopde, E.J.; Agrawal, A.; Bhatia, A.; Tiwari, K. Post quantum cryptography: A review of techniques, challenges and standardizations. In Proceedings of the 2023 International Conference on Information Networking (ICOIN), Bangkok, Thailand, 11–14 January 2023; pp. 146–151.
9. Qiu, D.; Luo, L.; Xiao, L. Distributed Grover's algorithm. *Theor. Comput. Sci.* **2024**, *993*, 114461. [CrossRef]
10. Kara, M.; Karampidis, K.; Panagiotakis, S.; Hammoudeh, M.; Felemban, M.; Papadourakis, G. Lightweight and Efficient Post Quantum Key Encapsulation Mechanism Based on Q-Problem. *Electronics* **2025**, *14*, 728. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.