Toward Key Independent Encryption Based on Q-Problem

Abdelkader Laouid*, Mostefa Kara†, Mohammad Hammoudeh‡ and Abdullah T. Al-Essa§
*LIAP Laboratory, PO Box 789, El Oued 39000, University of El Oued, Algeria.
†Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS), King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, 31261, Saudi Arabia.
‡Information & Computer Science Department, King Fahd University of Petroleum & Minerals, Academic Belt Road, Dhahran 31261, Saudi Arabia.
§Access Management & Data Protection Division, Information Protection Department, Digital & Information Technology, Saudi Aramco, Dhahran 31311, Saudi Arabia.

***Abstract*—This article defines a new post-quantum hard problem called Q-Problem. This Q-Problem is then used to design a post-quantum key-independent cryptography scheme. In the key-independent cryptography scheme, the keys of each communicating entity are hidden, and the encryption and decryption processes are performed using only random private keys.**

I. Introduction

Cryptography has to evolve in tandem with digital communication and computation to adapt to emerging threats and harness technological advancements to safeguard data integrity, confidentiality, and availability. The current array of encryption techniques, including RSA, El Gamal, and Elliptic Curve Cryptography (ECC), stands as the cornerstone of secure communication. These algorithms are secure against classical attacks due to their underlying computational complexity, e.g., integer factorization and discrete logarithms, setting high standards for security. However, the emergence of quantum computing poses a profound threat to these established encryption methods.

Quantum computing introduces a revolutionary paradigm in computation, leveraging the principles of quantum mechanics to process information in ways fundamentally different from classical computers. This emerging technology offers unprecedented computational power. Algorithms such as Shor's algorithm for integer factorization and Grover's algorithm for database search optimization were proposed to harvest the computational power of quantum computers. For instance, Shor's algorithm can factorize large integers in polynomial time, which is a task that is prohibitively time-consuming for classical computers and forms the basis of the security in RSA and similar encryption schemes. The ability of quantum computers to efficiently execute Shor's algorithm undermines the security of most traditional encryption techniques. Public-key cryptographic systems, which protect everything from Internet communications to financial transactions, could be decrypted without the private key, exposing sensitive information to quantum-powered adversaries. This imminent threat highlights the urgent need to develop quantum-resistant cryptography, also known as post-quantum cryptography, to ensure the continued security of our digital infrastructure. In response to the quantum threat to classical cryptography, researchers and industry leaders are actively developing encryption methods capable of withstanding quantum computational attacks. These quantum-resistant algorithms typically rely on mathematical problems that are considered challenging for both classical and quantum computers to solve. Promi-

Manuscript received XXXX; revised xxxxx.

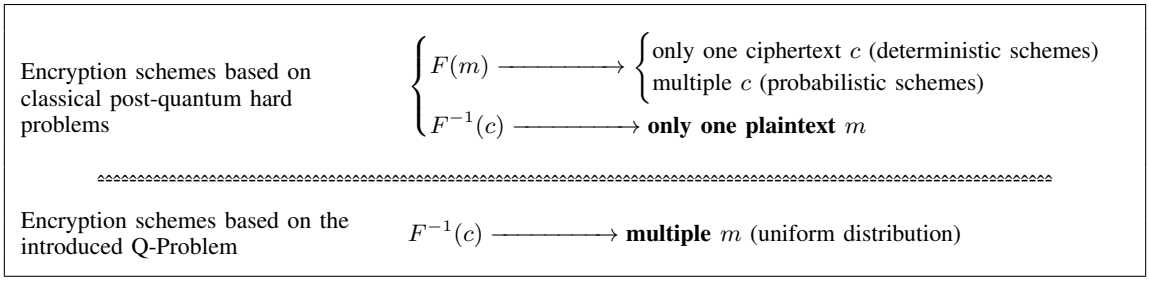


Fig. 1: Q-Problem illustration.

nent approaches include lattice-based cryptography [1], hash-based cryptography [2], and multivariate polynomial cryptography [3], each demonstrating promising quantum-resistant properties. While quantum computing presents significant risks to existing encryption techniques, the implementation and performance of most proposed post-quantum cryptographic systems introduce considerable complexity. This challenge demonstrates the delicate balance between enhancing security and maintaining efficiency in the post-quantum era.

This article presents two main contributions.

1) A new hard problem, called Q-Problem, that could be used in post-quantum encryption.
2) An innovative post-quantum key-independent cryptographic scheme.

The post-quantum, key-independent cryptographic scheme that uses the Q-Problem addresses complexity challenges and offers a novel approach to securing digital communications against quantum computing threats. The scheme's core concept is built around a newly defined challenge, known as the Q-problem, which conceals each entity's keys through encryption based only on the exchange of distinct private keys. By parameterizing complexity, the scheme allows for the dynamic adjustment of cryptographic hardness, ensuring an optimal balance between computational efficiency and robust security in both classical and post-quantum environments. This adaptable mechanism marks a significant advancement in the pursuit of robust, quantum-resilient encryption.

II. New post-quantum hard problem called Q-Problem

The emergence of quantum computers renders traditional cryptographic hard problems like factorization and discrete logarithms vulnerable, undermining their security foundations. Consequently, a new set of complex problems—such as those based on lattice structures, code theory, hash functions, multivariate polynomials, and isogenies has emerged as the basis for quantum-resistant encryption methods. These problems, despite their current robustness, rely fundamentally on their computational complexity. Thus, encryption techniques built upon them ultimately aim to provide one solution to Equation (1).

$$c = F^{-1}(m) \tag{1}$$

When quantum computers are fully realized, the aforementioned problems can easily solve this equation regardless of its complexity because each ciphertext maps to only one plaintext.

This article introduces a novel post-quantum hard problem, termed the Q-Problem, which arises when Equation (1) has multiple solutions. In this scenario, a single ciphertext corresponds to a broad set of valid plaintexts, making it difficult for an attacker, even with quantum computational power, to identify the correct plaintext from among many possibilities (see Figure 1). Q-Expressions (Qe) could take forms such as $z = x \times y \mod p$, or $z = x + y \mod p$, or $z = x^y \mod p$, where x and y are random and unknown. In each of these cases, numerous pairs of values for x and y produce the same z, with no

discernible pattern to determine which pair is the intended target.

Leveraging this Q-Problem, this article proposes a new encryption technique that offers an additional advantage, i.e., the encryption scheme is not tied to a fixed encryption key. Instead, a key is generated randomly during the encryption process and is used only once. In Q-Probleme, the attacker challenge is to find x and y from the equation $z = x \star y \mod p$, where x and y are unknown and both of them have no relation with p. Of course, without any other equation or information to help him guess x and y.

We define the Q-problem (QP) as follows:

$$\text{QP} \Leftrightarrow \begin{cases} \triangleright\ F() = \{Qe_1, Qe_2, \ldots, Qe_n\} \\ \triangleright\ Qe_i : x_i \star y_i \mod p \mid \star : +, \times, exp \\ \triangleright\ Both\ x_i\ and\ y_i\ are\ hidden \\ \triangleright\ \forall\ (Qe_i, Qe_j)\ initial\ or\ derived: \\ at\ least\ var(x_i) \neq var(x_j) \\ \ \vee\ var(y_i) \neq var(y_j) \\ \triangleright\ \forall\ Qe_i,\ (x_i, y_i)\ \theta\ p = \perp \\ \triangleright\ Given\ z, \forall\ Qe_i: \\ \#Sols_{Eq.}(\ z = x \star y \mod p) >> 1 \end{cases}$$

Var: for variable, $\perp$ means that p (or part of p) has no relation with x nor with y; x and y are unknown elements or composed arithmetic expressions of unknown elements. Qe can also be a single random value v with $v \neq v'$ for two different inputs.

The challenge in Q-problem can be defined as follows:

Q-problem: Given p ***and*** z***, find*** x ***and*** y ***where*** $x \star y \mod p = z$**.**

Initial and derived Q expression:
Let us take the following example, $F() : (Qe_1, Qe_2, Qe_3) = (x + c, y + c, x \times y)$ where c is a constant. This initial $F()$ verifies Q-Problem definition where $Qe_i \cap Qe_j = \{e\} \mid e \in \{x, y, c\}$. But if we derive new Qe from $F()$, we will find $Qe_4 = Qe_1 - Qe_2 = x - y$. Now, the pair (Qe_3, Qe_4) does not verify Q-Problem definition because $Qe_3 \cap Qe_4 = \{x, y\}$, it is ($x$ and y) and not (x or y). The attacker can get the hidden values x and y by using Qe_3 and Qe_4.

Connected/Disconnected Q-Problem (C/DQP): If we have $var(x_i) = var(x_j)$ or $var(y_i) = var(y_j)$, this is a connected Q-Problem. If $\forall\ (Qe_i, Qe_j) :\ var(x_i) \neq var(x_j)$ and $var(y_i) \neq var(y_j)$, we have a disconnected Q-Problem, i.e. there is no common elementary variable between any two Q expressions of $F()$.

Totally/Partly Q-Problem (T/PQP):
We mean the vertical change, i.e., two outputs of two different inputs. Let z_1 and z_2 be two outputs of two different inputs t_1 and t_2. By default, we have $z \neq z'$, in addition, if $(x_{t_1} \neq x_{t_2} \vee\ y_{t_1} \neq y_{t_2})\ \forall\ Qe$, it is a partly Q-Problem (PQP).

In totally Q-Problem (TQP), $z \neq z'$, and $(x_{t_1} \neq x_{t_2} \wedge\ y_{t_1} \neq y_{t_2})\ \forall\ Qe$.

Fully/Partially probabilistic Q-Problem (F/PPQP):
We mean the horizontal change i.e., two different outputs of the same input. Let z and z' be two outputs of the same input t, we have $z \neq z'$, and $(x \neq x' \vee\ y \neq y')\ \forall\ Qe$. This is a partially probabilistic Q-Problem (PPQP).

In fully probabilistic Q-Problem (FPQP), $z \neq z'$, and $(x \neq x' \wedge\ y \neq y')\ \forall\ Qe$.

In deterministic Q-Problem (DQP), $z = z'$. For operations that are not repetitive such as generating a public key, it is sufficient for x and y to be unknown.

We note that: $F/PPQP \Rightarrow T/PQP$

Perfect Q-Problem:
We call Perfect Q-Problem (FQP) if we have a fully probabilistic Q-Problem FPQP and if we decompose it, we always obtain an FPQP until the last decomposition i.e., getting a non-divisible Q-expression.

To understand this concept, let us give the following example. Let c be a constant (for example, a secret key) and r be a random variable for each execution of π with $\pi = c \times r_1 + c^{r_2}$. If we put $x = c \times r_1$ and $y = c^{r_2}$, so $\pi = x + y$ is FPQP. However, if we move down another level of decomposition, we will put $x = r_1$ and $y = c$ for the first part (previously x). At this level, $x \neq x'$, where $x' = r_1'$, but $y = y' = c$. In this level, π is only PPQP. Therefore, π is not FQP.

OTP encryption and Q-Problem:
It is known that in OTP we use a one-time encryption key. 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, this example of OTP is not fully probabilistic but a partially probabilistic Q-problem. To Convert it from PPQP to FPQP, we can for example fragment the message m randomly into two parts. Thus, $c = (m_1 \oplus k_1, m_2 \oplus k_2)$, that gives $x = m_1 \neq x' = m'_1$ and $y = k_1 \neq y' = k_2$, same think for the second part. Therefore, we need to use two keys for each message instead of one.

Examples

Let us take RSA, $c = m^e \mod n$ where e is the public key. Since e is known, RSA is not QP. Furthermore, e is 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 belongs to QP, $x = m$ (unknown) and $y = h^r$ (unknown because r is hidden). Since g is public, the second part g^r 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 belongs to the Q-Problem, if we put $x = m + 2r$ and $y = qp$, then $c = x + y$ verifying all conditions of the Q-Problem including FPAP. After decomposing $y = qp$ to $x = q$ and $y = p$, DGHV scheme is no longer FPAP.

The ideal Q-Problem scheme

The ideal Q-Problem scheme (IQP) is $DFQP$ which means a disconnect and perfect Q-Problem. An example of IQP is what we illustrated above, 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 with the same plaintext m (so it is FQP). Q expressions (ciphertexts) c_1 and c_2 have no common variable (so it is DQP).

III. Key-Independent Cryptography

This section presents the details of a new cryptographic scheme, KIE, which relies on the Q-Problem for its security. We present two implementations of KIE: one for classical computing and the other is a post-quantum cryptography.

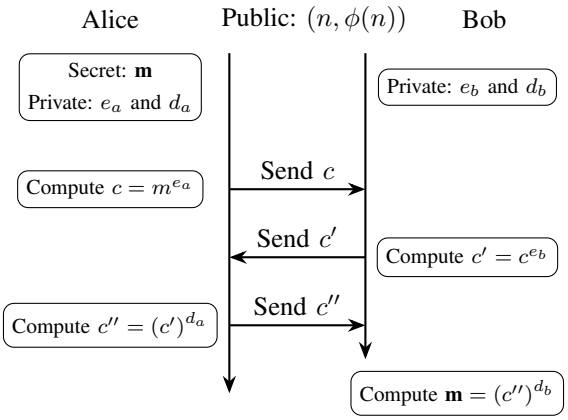


Fig. 2: Classical scheme of KIE encryption and decryption.

A. KIE for Classical Computing

Finding large prime numbers remains a challenge in several cryptography fields. In the proposed Classical KIE (C-KIE) key exchange protocol, Alice and Bob must agree on a random modulus n, which defines the finite set $\mathbb{Z}_n$, within which the message m must be an element. As depicted in Figure 2, Alice encrypts m using her secret key e_a and sends the resulting ciphertext c to Bob. Bob then encrypts c again using his secret key e_b and sends the new ciphertext c' back to Alice. At this stage, Alice decrypts c' using her private key d_a and forwards the intermediate result c'' to Bob, who finally decrypts it with his private key d_b to reveal the original message m.

The use mode of the C-KIE process is similar to RSA, as illustrated in Figure 3, where the main difference is that $\phi(n)$ is public in C-KIE, and the private keys are the randomly selected pairs (e, d). This approach facilitates secure message exchange without the need for pre-established large prime numbers or key sharing.

C-KIE is particularly efficient in low-resource environments, e.g., the Internet of Things (IoT), where encryption can be initiated by simply sharing a large random n. The ability to choose random keys and encrypt each message m_i with randomly distinct selected key pairs (e_i, d_i) enhances C-

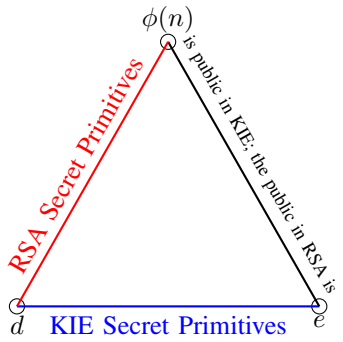


Fig. 3: RSA and KIE security triangle.

KIE's robustness compared to traditional asymmetric cryptography schemes.

B. Post-Quantum KIE Cryptography

Quantum computing poses significant threats to cryptographic systems that are based on the discrete logarithm problem. C-KIE addresses the factorization challenge by making $\phi(n)$ public while keeping (e, d) private. Consequently, the security of C-KIE depends on the practical difficulty of solving the discrete logarithm problem. As illustrated in Figure 2, the ciphertext c' is derived from c through the exponentiation $c' = c^{e_b}$, where both c and c' are known. In the presence of quantum attacks, e_b becomes vulnerable, allowing an adversary to potentially uncover the private message m from c''.

To mitigate this, a new post-quantum KIE (Q-KIE) technique, illustrated in Figure 4, is designed to obscure both the base and the exponent, thus transforming it into a more challenging discrete logarithm problem. In Figure 4, the blue color indicates a piece of shared information, the red color indicates a piece of secret information and the black indicates newly generated information. The encryption here utilizes a Secret Message Holder (SMH) approach. Bob sends the SMH $(c_1 = x^{e_3},\ c_2 = x^{e_4})$ to Alice, where x is Bob's secret random value in $\mathbb{Z}_n$ (with $x > 1$), and (e_3, e_4) are his secret random exponents in $\mathbb{Z}_{\phi(n)}$. As depicted in Figure 4, Alice encrypts her message m by raising c_1 and c_2 to her secret random exponent r, yielding c_1^r and c_2^{-r}. She then multiplies c_1^r by ${m_1}^{e_1}$ to produce c_3 and c_2^{-r} by ${m_2}^{e_2}$ to generate c_4. Upon receiving c_3 and c_4, Bob computes c_5 and c_6 and sends them back to Alice. In the final stage of this protocol, Bob extracts the messages m_1 and m_2. A simplified Python code demonstrating this scenario, along with an explanation of the running mode of Q-KIE, is hosted on https://t.ly/MQWMk. The proposed Q-KIE scheme eliminates the need for a public key sharing process and offers the flexibility of encrypting each message with distinct private primitives.

IV. KIE Security Analysis

Asymmetric cryptosystems that depend on the computational difficulty of processes such as factorization or discrete logarithms will become vulnerable in the future due to the immense computational power of quantum computers, which can effectively solve many of these problems. Shor's algorithm exemplifies such quantum capabilities. For instance, the factorization problem, which has a complexity of $O(\exp(L^{1/3}(\log L)^{2/3}))$ on classical computers, is drastically reduced to $O(L^3)$ on quantum computers for factoring non-prime integers N of L bits.

Shor's method relies on a period-finding routine on a quantum computer. A function $f : (x_1, \ldots, x_n) \longmapsto f(x_1, \ldots, x_n)$ is periodic, of period $(\omega_1, \ldots, \omega_n)$, if $f(x_1 + \omega_1, \ldots, x_n + \omega_n) = f(x_1, \ldots, x_n)$ for all tuples $(x_1, \ldots, x_n)$ in the domain of f.

Factorization problem: Given an RSA modulus $N = p \times q$, find primes p and q.

Choose a random integer $\alpha \in ZN$, without loss of generality, we assume that $gcd(\alpha, N) = 1$ otherwise, this yields the factorization of N and the factorization problem is solved.

Consider the univariate function $f(x) = a^x \mod N$. The period-finding algorithm determines a period ω such that $f(x + \omega) = f(x)$. Consequently, ω is a multiple of the order of α modulo N. Specifically, this relationship holds because $f(x + \omega) = f(x)$ if and only if $\alpha^\omega \equiv 1 \pmod N$.

If ω is a multiple of $\lambda(N)$ where $\lambda(N)$ denotes Carmichael's function, then Miller's algorithm yields the factorization of N. Otherwise, repeat the

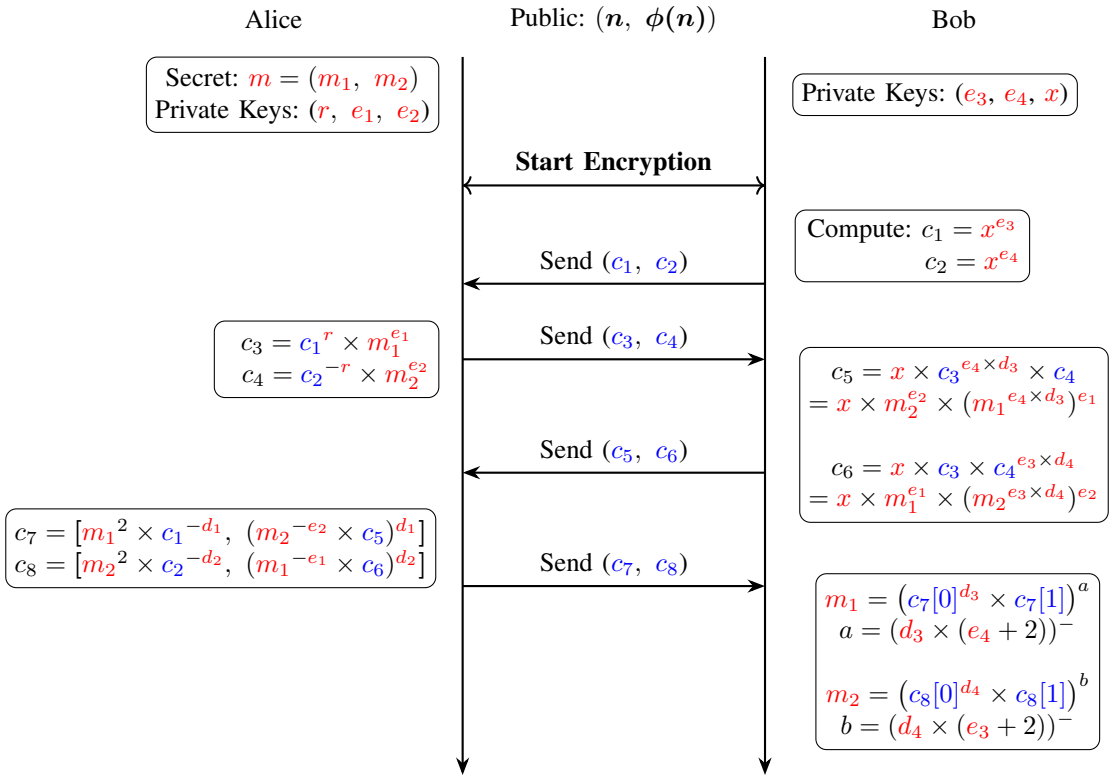


Fig. 4: Post-quantum independent key encryption scheme, private keys of both sender and receiver are randomly generated in each encryption.

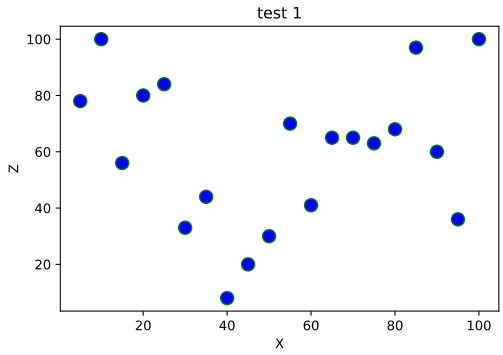
process with another α, get the period ω_α, and update ω as $\omega \longleftarrow lcm(\omega, \omega_\alpha)$, until ω is a multiple of $\lambda(N)$.

Discrete logarithm problem: Given a Diffie-Hellman modulus $g^x = y \mod p$, find x.

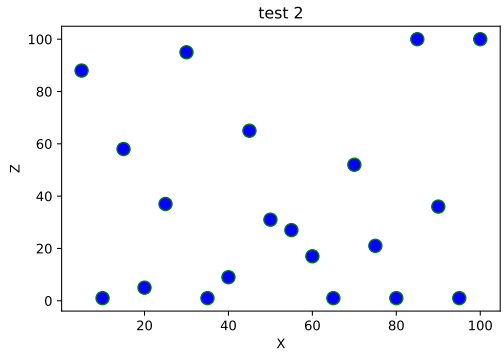
Shor's algorithm addresses the discrete logarithm problem by finding an integer x satisfying the equation $g^x = y \mod n$, where g is a generator of the multiplicative group of integers modulo n, y is an element of this group, and p is the modulus.

The process of solving the discrete logarithm problem using Shor's algorithm involves several steps, outlined as follows:

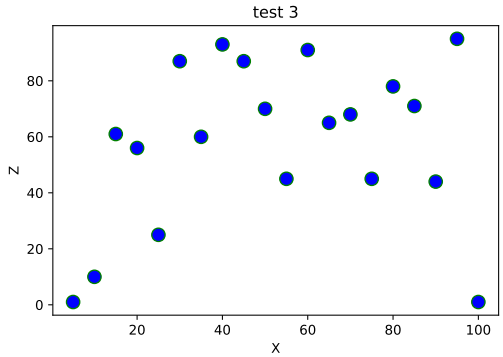
1) *Quantum Fourier Transform:* The algorithm employs the quantum Fourier transform to ascertain the period r of the function $f(a) = g^a \mod p$, where a is an arbitrary integer. The period r is the smallest positive integer for which $g^r \equiv 1 \mod p$.
2) *Period Finding:* At the heart of Shor's algorithm is the quantum computation for efficient period finding. By preparing states in a superposition and evaluating f in this superposed state, the algorithm leverages the quantum Fourier transform to extract information regarding r.
3) *Computing the Discrete Logarithm:* Given the period r, the algorithm proceeds to compute the discrete logarithm x as follows:
   a) If r is even and $g^{r/2} \not\equiv -1 \mod n$, it is possible that $g^{r/2} - 1$ and $g^{r/2} + 1$ yield clues towards finding x.
   b) Given $y = g^x$, we search for x such that $y^r \equiv (g^x)^r \equiv 1 \mod n$. If r is even, we



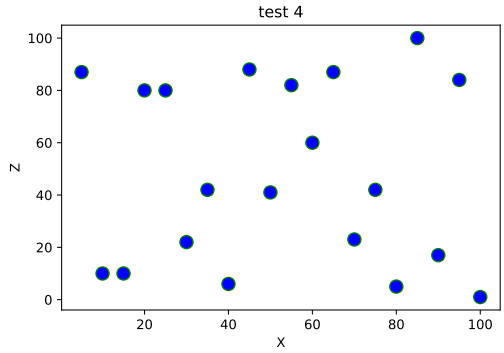
(a) First test with $p = 101$ and $x = 5, 10, 15, 20, \ldots, p - 1$.



(b) Second test with $p = 101$ and $x = 5, 10, 15, 20, \ldots, p - 1$.



(c) Third test with $p = 101$ and $x = 5, 10, 15, 20, \ldots, p - 1$.



(d) Forth test with $p = 101$ and $x = 5, 10, 15, 20, \ldots, p - 1$.

Fig. 5: Illustrating the random distribution of $z = x^y$ even if x are regular values; $p = 101, x = 5, 10, 15, 20, \ldots, p - 1$.

have $y^{r/2} \equiv \pm 1 \mod n$, which provides insights into the structure of x.

4) *Modular Exponentiation:* Efficient quantum modular exponentiation is critical for applying Shor's algorithm effectively to solve both the factoring and the discrete logarithm problems.

In essence, Shor's algorithm utilizes the quantum mechanical properties to solve the discrete logarithm problem by relating the order of y with respect to g to the period r identified by the quantum algorithm.

The Q-KIE technique is considered robust against quantum computers because it does not depend on the difficulty of the factorization problem by considering p and q already known. Furthermore, making $\phi(n)$ public shifts the challenge to another problem, namely the discrete logarithm problem.

The discrete logarithm problem involves finding x in the equation $g^x = y$ when g and y are known. Several studies prove that it is possible to solve this problem using future generations of quantum computers. This directly affects the widely used encryption techniques, such as RSA; since e is public, an attacker can choose a message m and

compute $c = m^e$, he now knows that $c^d = m$, where d is the secret key.

Q-KIE states that x and y are unknown, so the adversary knows only z in $x^y = z$. We analyze this problem in the presence of quantum computers.

Consider the bivariate function $f : (x_1, x_2) \longmapsto g^{x_1} \times y^{x_2}$.

$$g^{x_1} \times y^{x_2} = g^{x_1+\omega_1} \times y^{x_2+\omega_2} \tag{2}$$

The period finding routine finds a pair (ω_1, ω_2) such that $f(x_1+\omega_1, x_2+\omega_2) = f(x_1, x_2)$. This implies: $g^{\omega_1} \times y^{\omega_2} = 1_G \iff g^{\omega_1+k\times\omega_2} = 1_G$ and thus $\omega_1 + k \times \omega_2 \equiv 0$, or $k \times \omega_2 \equiv -\omega_1 (\mod (p-1))$. There are p pairs (ω_1, ω_2) that produce this result. If each result is equally likely, then there is only a $1/p$ probability that $(\omega_1, \omega_2) \equiv (0, 0) (\mod p)$. On the $(q-1)/q$ probability that it is not zero, the solution to the discrete logarithm problem is given by $k = -\omega_1/\omega_2 \mod (p-1)$.

Regarding Equation (2), we observe that knowing g is necessary to continue looking for x because if g is unknown, the adversary needs to choose a random value. In this case, the adversary will obtain for each chosen g a new x different from the original value.

Set $p = 11, x = 6, y = 4$, and $z = 9$ in $x^y \equiv z \mod p$, Table I shows an example of $z = 9$ for different pairs of (x, y).

TABLE I: Exp: p = 11, x = 6, y = 4, and z = 9 such $x^y \equiv z \mod p$

x \ y	2	3	**4**	5	6	7	8	9
2	4	8	5	10	9	7	3	6
3	9	5	4	1	3	9	5	4
4	5	9	3	1	4	5	9	3
5	3	4	9	1	5	3	4	9
6	3	7	**9**	10	5	8	4	2
7	5	2	3	10	4	6	9	8
8	9	6	4	10	3	2	5	7
9	4	3	5	1	9	4	3	5

By knowing only z, the adversary will get many possibilities for x and y that verify $z = x^y$; therefore, applying quantum algorithms to get a solution (x, y) is not effective even if the adversary can get all solutions (x_i, y_i) because the adversary can not check which of these pairs is the correct one.

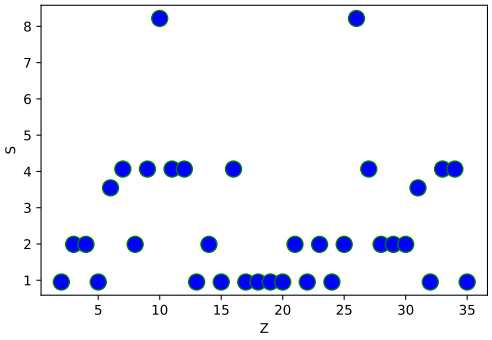


Fig. 6: Number of samples; S: # of z, $z = x^y$; $x = 2top$, $y = 2top$ for $p = 37$.

Figure 6 illustrates the distribution of samples for various values of x. For instance, with $x = 10$, we observe an average sample rate of 7.80%, indicating that approximately 7.80% of the samples correspond to values like 10^2, 10^3, and so forth, where $z_i = 10^{y_i}$ for y_i in the range $[2, p]$. For $p = 37$, the Overall Average of Samples (OAS) for any encrypted message is 4.23% (derived from $(p-2)^2$). This ratio changes with varying p; for example, the pairs (p, OAS) are as follows: (37, 2.56), (43, 2.32), (53, 1.88), (63, 1.36), and (79, 1.25).

Q-KIE involves exchanging data in one of two forms: $z = x_1^{x_2}$ or $z' = x_1 \times y_1^{x_2}$, where x_i represents unknown values and y_i represents known values. In both scenarios, regardless of the value of z, there exists a corresponding value z_1 such that $z_1 = z$ and $z_1 = x_3^{x_4}$, where $x_3 \neq x_1$ and/or $x_4 \neq x_2$. Similarly, for z', there is a z'_1 such that $z'_1 = z'$.

Lemma IV.1. *$\forall\ z \in \mathbb{Z}_p, z = x_1^{x_2}, \exists\ z_1 = z$ where $z_1 = x_3^{x_4}$ with $x_3 \neq x_1$ and/or $x_4 \neq x_2$.*

Proof. We know that $\mathbb{Z}_p$ contains exactly $\phi(p-1)$ generators (primitive roots).

Let g_1 and g_2 two different generators modulo p.

We pick a random value z, $\exists\ \alpha_1$ verifies $g_1^{\alpha_1} = z$ and $\exists\ \alpha_2$ verifies $g_2^{\alpha_2} = z$. □

Lemma IV.2. *$\forall\ z \in \mathbb{Z}_p, z' = x_1 \times y_1^{x_2}, \exists\ z'_1 = z'$ where $z' = x_3 \times y_2^{x_4}$ with $x_3 \neq x_1$ and/or $x_4 \neq x_2$*

and/or $y_2 \neq y_1$.

Proof. We know that if k is a prime number where $k < p$, k generates $\mathbb{Z}^*{}_p$ i.e., $\mathbb{Z}_p = k \times i \ \forall \ i \in \mathbb{Z}_p$ because $i = k \times i \times k^{-}$.

Let k_1 and k_2 two different prime numbers where $k_1, k_2 < p$.

We pick a random value z, $\exists\ \alpha_1$ verifies $k_1 \times \alpha_1 = z$ and $\exists\ \alpha_2$ verifies $k_2 \times \alpha_2 = z$. □

The described problem is a variant of the discrete logarithm problem, introducing additional layers of complexity by making both the base and the exponent unknown and by not restricting x to be a generator of the group.

A. Quantum Hard Logarithm Problem

Given a finite cyclic group G of order n, and an element $z \in G$, the *Inverted Discrete Logarithm Problem (IDLP)* is defined as the problem of finding all pairs of integers (x, y) for which x is not necessarily a generator of G, and the following condition is satisfied:

$$x^y \equiv z \mod n \tag{3}$$

where $x, y \in \mathbb{Z}$, $1 < x < n$, and $1 \leq y < \phi(n)$. The IDLP is characterized by:

1) **Non-Generator Base**: The base x is not restricted to generators of the group, permitting x to potentially generate a proper subgroup of G or no subgroup at all. This attribute expands the search space for solutions.
2) **Multiple Solutions**: Diverging from the traditional DLP where x is known and a unique y is sought, the IDLP entertains multiple valid (x, y) pairs satisfying the equation for a given z, attributable to the relaxed condition on x and the unknowns in both x and y.
3) **Computational Complexity**: The dual unknowns and the relaxation of x being a generator amplify the problem's complexity.

On the other hand, Shannon's theorem on perfect secrecy states that a given cryptographic system is perfectly secure if and only if every plaintext is equally likely to produce any given ciphertext. Shannon's theorem sets forth three critical conditions for perfect secrecy:

1) The key must be truly random, ensuring that there is no predictable pattern that an attacker can exploit.
2) The key must be at least as long as the message is encrypted so that the key does not repeat. Repeating keys introduce patterns that can be analyzed to break the cipher.
3) The key must never be reused in whole or part, as any reuse also introduces patterns that compromise secrecy.

Using a One-Time Pad (OTP) offers unbreakable security in message transmission. With OTP, each message is encrypted with a unique key generated specifically for that message and used only once. Consequently, even if an adversary manages to intercept and decipher one message, they gain no advantage in decrypting subsequent messages. Unlike other encryption methods where compromising a single key could potentially compromise the security of multiple messages, OTP necessitates the acquisition of each key for deciphering the specific corresponding message. This characteristic significantly amplifies the decryption complexity for any malicious actor, as they would need to obtain every unique key for every message to access the corresponding plaintext. As a result, the KIE cryptosystem ensures confidentiality and provides an added layer of protection against potential cryptographic attacks.

Figure 7 proves that a perfect OTP-based KIE encryption is achieved without any need for a pre-sharing keys process, unlike the classical OTP. In KIE, for each message, the sender and receiver generate new random numbers that are used once to hold the plaintext based on performing post-quantum encryption using two types of encapsulations: x^y and $x \times y$, where x and y are unknown giving a large number of possibilities of the used keys and encrypted message.

B. Optimal Configuration

This subsection gives the preferable Q-KIE configurations to offer robust encryption. To provide a large key pool, it is preferred to use a prime number n of the form $n = p \times 2^i + 1$ to obtain $\phi(n) = p \times 2^i$, i.e., all odd numbers are not multiple

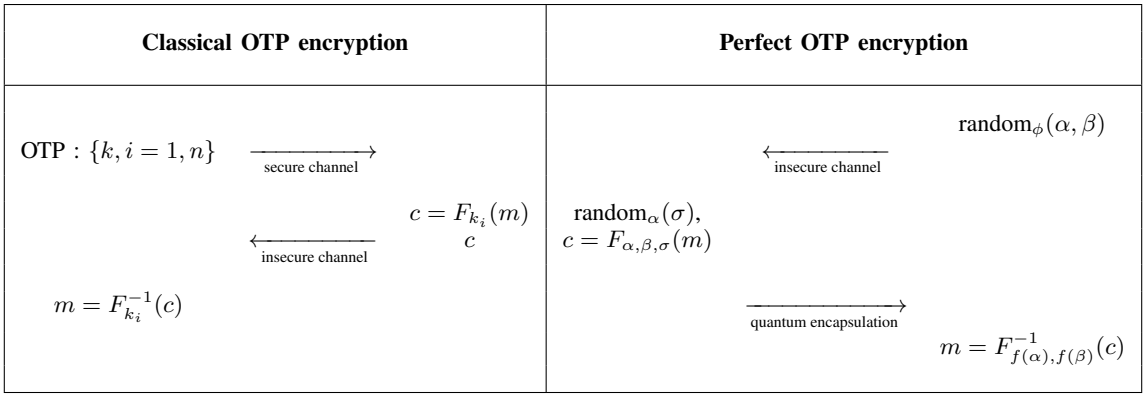


Fig. 7: Classical OTP encryption vs. perfect OTP encryption.

of p and are coprime with $\phi(n)$. Setting $p = 1$ offers the best configuration of Q-KIE to encrypt any $1 < m < n$ with a key pool equal to the half of $\phi(n)$. However, the are only five known prime numbers of the form $2^i + 1$.

Another form of n that offers a suitable configuration and avoids the challenge of finding large primes is in the form p^j. Since there are only 3, 5, 17, 257 and 65537 known prime numbers of the form $2^i + 1$, it is preferable to consider $n = (65537)^j$. In this case, $\phi(n) = (65537)^{j-1} \times 2^{16}$ and $\phi(\phi(n)) = (65537)^{j-2} \times 2^{31}$, which means that any m that is a coprime with 65537 could be encrypted and all the odd numbers coprime with 65537 are valid keys. On the other hand, we observe that Figure 4 uses the modular inverse of x and m to compute (c_3, c_4) and (c_7, c_8) respectively. Hence, using multiple primes in Q-KIE mitigates the pool's size of x and m. Setting $n = (65537)^j$ means that only m and x are not multiples of 65537 and could not be encrypted and used as an SMH, respectively.

To provide more complexity in the x^{e_i} base and exponent findings, it is preferred to use random non-generators. This is because the generators provide distinct values for each exponent i which mitigates the number of possibilities.

V. Conclusion

In light of quantum attack threats, KIE leverages the Q-Problem, a complex variant of the traditional discrete logarithm problem. This problem is based on the computational intricacies of solving for both the base and the exponent within the field of modular exponentiation. Its broader and more flexible scope permits multiple solutions and does not require x to be a generator, opening new avenues for research in cryptographic security and computational number theory.

REFERENCES

[1] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-quantum cryptography*. Springer, 2009, pp. 147–191.

[2] R. C. Merkle, *Secrecy, authentication, and public key systems.* Stanford university, 1979.

[3] T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature-verification and message-encryption," in *Advances in Cryptology—EUROCRYPT'88: Workshop on the Theory and Application of Cryptographic Techniques Davos, Switzerland, May 25–27, 1988 Proceedings 7*. Springer, 1988, pp. 419–453.