Toward Independent Key Encryption based on Q-Problem

Abdelkader Laouid*, Mostefa Kara† and Mohammad Hammoudeh‡
*LIAP Laboratory, PO Box 789, El Oued 39000
University of El Oued, Algeria.

***Abstract*—This paper defines a post-quantum encryption scheme based on discussion cryptography by introducing a new post-quantum hard problem called Q-Problem. The idea behind this scheme is to hide the keys of each entity, and the encryption process is based on secret message holders using only random private keys.**

I. Introduction

Cybersecurity has been an ever-evolving field With recent advances in digital communication and computation, referring dynamically to new threats and leveraging technological advances to protect data integrity, confidentiality, and availability. Currently used encryption techniques, such as RSA, El Gamal, and ECC (Elliptic Curve Cryptography), are the backbone of secure communication, providing robust defense mechanisms against classical computing attacks. These cryptographic algorithms rely on the computational difficulty of problems like integer factorization or discrete logarithms, ensuring high security for current standards. However, quantum computing presents significant challenges to these established encryption methods.

Quantum computing introduces a new paradigm in computation, harnessing the principles of quantum mechanics to process information in ways fundamentally different from classical computers. This emerging technology offers unprecedented computational power, particularly through algorithms like Shor's algorithm for integer factorization and Grover's algorithm for database search optimization. Shor's algorithm, in particular, can factorize large integers in polynomial time, a prohibitively time-consuming task for classical computers based on the security of RSA and similar encryption schemes. The potential of quantum computers to execute Shor's algorithm effectively renders most traditional encryption techniques vulnerable. Public-key cryptographic systems, which secure everything from internet communications to financial transactions, could be decrypted without the private key, exposing sensitive information to quantum-enabled adversaries. This scenario underscores a pressing need to develop quantum-resistant cryptography, sometimes called post-quantum cryptography, to safeguard against the looming quantum threat.

In response to these challenges, several academic researchers and industrial companies actively provide encryption methods that can withstand quantum computational attacks. Quantum-resistant algorithms typically rely on mathematical problems that are believed to be difficult for both classical and quantum computers to solve. Lattice-based cryptography [1], hash-based cryptography [2], and multivariate polynomial cryptography [3] are among the leading approaches being explored for their quantum-resistant properties. While quantum computing poses significant risks to current encryption techniques, almost proposed post-quantum cryptographic systems introduce a notable complexity regarding their implementation and performance.

This paper defines an independent key cryptographic scheme that considers complexity factors and provides a groundbreaking approach to securing digital communication against quantum computer threats. The idea behind this scheme is to hide the keys of each entity and the encryption is based on discussion using only distinct private keys. The scheme can flexibly increase or decrease its cryptographic hardness by parameterizing complexity, ensuring an optimal balance between computational efficiency and robust security in classical or post-quantum fields.

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

With the emergence of quantum computers, the traditional hard problems, certainly factorization and discrete logarithms, become no longer safe in cryptography. Another set of hard problems, such as lattice, code-based, hash-based, multivariate algorithm, isogeny, etc., was relied upon. However, all these problems remain dependent on complexity, and all encryption techniques based on them give one and only solution to Equation (1).

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

In other words, if the quantum computers are controlled, it will become easy to solve this equation, regardless of their complexity level, because simply each ciphertext gives only one plaintext. This paper defines a new post-quantum hard problem called (Q-Problem). The problem is when Equation (1) has multiple solutions, meaning that the ciphertext gives a large set of correct plaintexts. Therefore, even with quantum computers, the attacker will obtain many valid plaintexts for the same ciphertext, with no pattern to determine which plaintext is the correct target (Figure 1). For instance, $z = x \times y \mod p$, $z = x + y \mod p$, $z = x^y \mod p$, where x and y are random and unknown, in each of these equations, there are a large number of solutions (pairs) that give the same value for z without the possibility of distinguishing between them which one is the desired target. Based on this Q-Problem, this paper provides a new encryption technique with another additional advantage, i.e., the encryption scheme is not linked to any fixed encryption key. Rather, the key is generated randomly during encryption and is used only once.

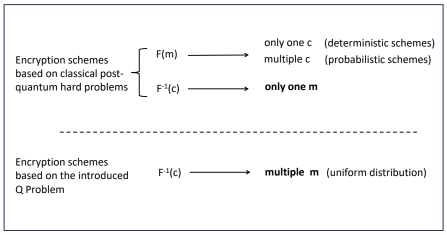

Encryption schemes based on classical post-quantum hard problems
F(m)
only one c (deterministic schemes)
multiple c (probabilistic schemes)
F-1(c)
only one m
Encryption schemes based on the introduced Q Problem
F-1(c)
multiple m (uniform distribution)

Fig. 1: Q-Problem illustration.

We can define Q-problem as follows:

$$\text{Q-problem} \Leftrightarrow \begin{cases} f(m) = x + y \ or \ = x \times y \ or \ = x^y \ \mid \\ x, y : \ variable \ or \\ \quad arithmetic \ expression \ of \ variables, \\ x, y : \ unkown, \\ x, y : \ random \ in \ each \ operation, \ i.e., \\ \qquad f(m_i) \neq f(m_j), x_i \neq x_j \wedge y_i \neq y_j \\ \qquad even \ if \ m_i = m_j. \end{cases}$$

III. Independent Key Encryption Scheme

The huge advances in quantum computation have sparked a broad spectrum of concerns, reflecting the potential for this technology to transform various sectors, from cryptography to chemical to complex system simulations. This section briefly discusses the weaknesses and complexity of the most known cryptosystems compared with the recent advances in computer technologies.

This study introduces an encryption method tailored to quantum technology's computational capabilities. Quantum computing's vast processing power makes most existing cryptographic systems, especially those relying on the large prime number factorization problem or the logarithmic discrete problem, ineffective. The Independent Key Encryption (IKE) scheme provides more flexibility to Alice and Bob, who need only sharing a random n and start exchanging confidential information. Furthermore, IKE introduces new concepts to cope with quantum computation.

A. Classical IKE scheme

Finding large prime numbers is still challenging in several security networking fields, like the Internet of Things (IoT). In the IKE, Alice and Bob must

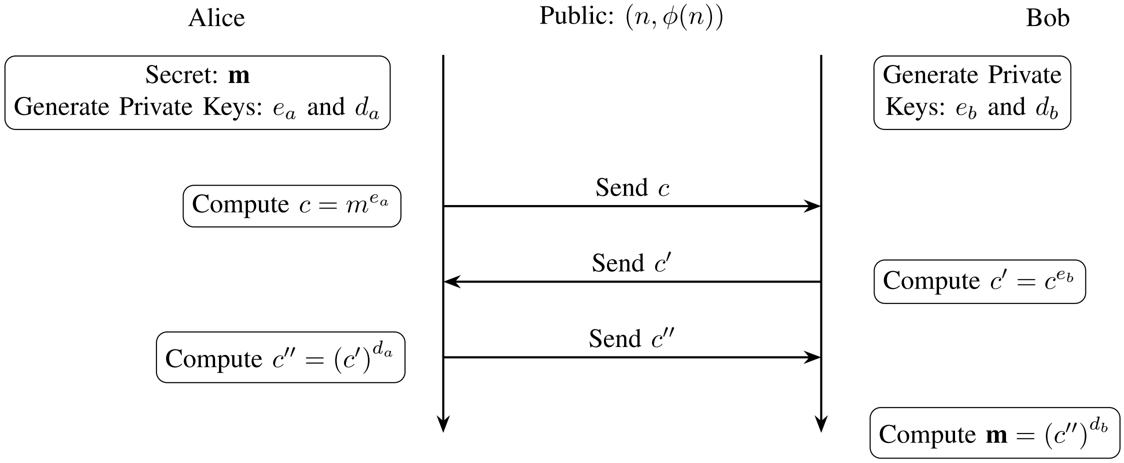

Alice
Public: $(n, \phi(n))$
Bob
Secret: $\mathbf{m}$
Generate Private Keys: e_a and d_a
Generate Private Keys: e_b and d_b
Compute $c = m^{e_a}$
Send c
Send c'
Compute $c' = c^{e_b}$
Compute $c'' = (c')^{d_a}$
Send c''
Compute $\mathbf{m} = (c'')^{d_b}$

Fig. 2: Classical Independent Key Encryption (IKE).

agree on a random modulus n, which inherently defines the finite set $\mathbb{Z}_n$, where the message m must be an element of $\mathbb{Z}_n$. In IKE encryption mode, as illustrated in Figure 2, Alice encrypts m with her secret e_a and sends the result c to Bob, which encrypts the received c again using his secret key e_b and resends the obtained result c' to Alice. In this stage, Alice removes her encryption using her secret key d_a and sends the result c'' to Bob, who reveals m using his secret key d_b. The encryption/decryption of IKE is similar to the RSA, see Figure 3, except $\phi(n)$ in IKE is public, and the private is the couple (e, d). This scenario provides a secret message exchange without any pre-requirements like finding large prime numbers or key exchanges. IKE is particularly efficient in low-resource environments where the encryption process can be started by sharing a large random n. The flexibility of choosing random keys and the possibility of encrypting each m_i with distinct random keys (e_i, d_i) renders IKE more robust than asymmetric schemes.

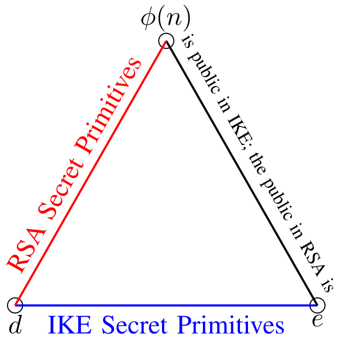


Fig. 3: RSA and IKE Security Triangle.

B. Post Quantum IKE Scheme

Quantum computation threats are also present in discrete logarithmic problems. Classical IKE resolved the factorization problem by revealing $\phi(n)$ and hiding (e, d). Hence, the security of classical IKE relies now on the practical difficulty of logarithmic discrete. We observe in Figure 2 that $c' = c^{e_b}$ such that c' and c are known. With the presence of quantum attacks, the value of e_b is considered prone to be cracked and hereafter obtain the private m from c''. Figure 4 illustrates post-quantum encryption that hides both the base and the exponent to be considered a more hard discrete logarithmic problem. The encryption here is based on the Secret Message Holder (SMH), where Bob sends the SMH $(c_1 = x^{e_3},\ c_2 = x^{e_4})$ to Alice, such that x is the Bob secret random in $\mathbb{Z}_n$ with $x > 1$, and $(e_3,\ e_4)$ are also Bob secret randoms in $\mathbb{Z}_{\phi(n)}$. As shown in Figure 4, Alice encrypts the message m by first raising c_1 and c_2 to the power of Alice secret random r, obtaining c_1^r and c_2^{-r}. She then multiplies c_1^r by ${m_1}^{e_1}$ to obtain c_3 and c_2^{-r} by ${m_2}^{e_2}$ to obtain c_4, respectively. When Bob receives c_3 and c_4 from Alice, he computes c_5 and c_6 and sends them to Alice as shown in the Figure. At the end of this discussion, Bob extracts m_1 and m_2.

IV. Cryptanalysis

In the future, several asymmetric cryptosystems will be considered unsafe because they rely on the hardness of the factorization problem or discrete logarithmic problem, which can be easily solved with the huge computation power of quantum computers. Shor's algorithm is one of these quantum solutions. For example, the factorization problem complexity on classical computers is $O(exp(L^{1/3}(logL)^{2/3}))$, whereas on the quantum computer is $O(L^3)$ for factorizing 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 $f : x \longmapsto f(x) = a^x \mod N$.

The period finding routine finds an ω such that $f(x + \omega) = f(x)$. Consequently, ω is a multiple of the order of α modulo N. Indeed, one has $f(x + \omega) = f(x) \iff \alpha^\omega \equiv 1(\mod N)$.

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 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 DLP 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 DLP 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 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 pivotal 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 DLP by relating the order of y with respect to g to the period r identified by the quantum algorithm.

Q-IKE 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,

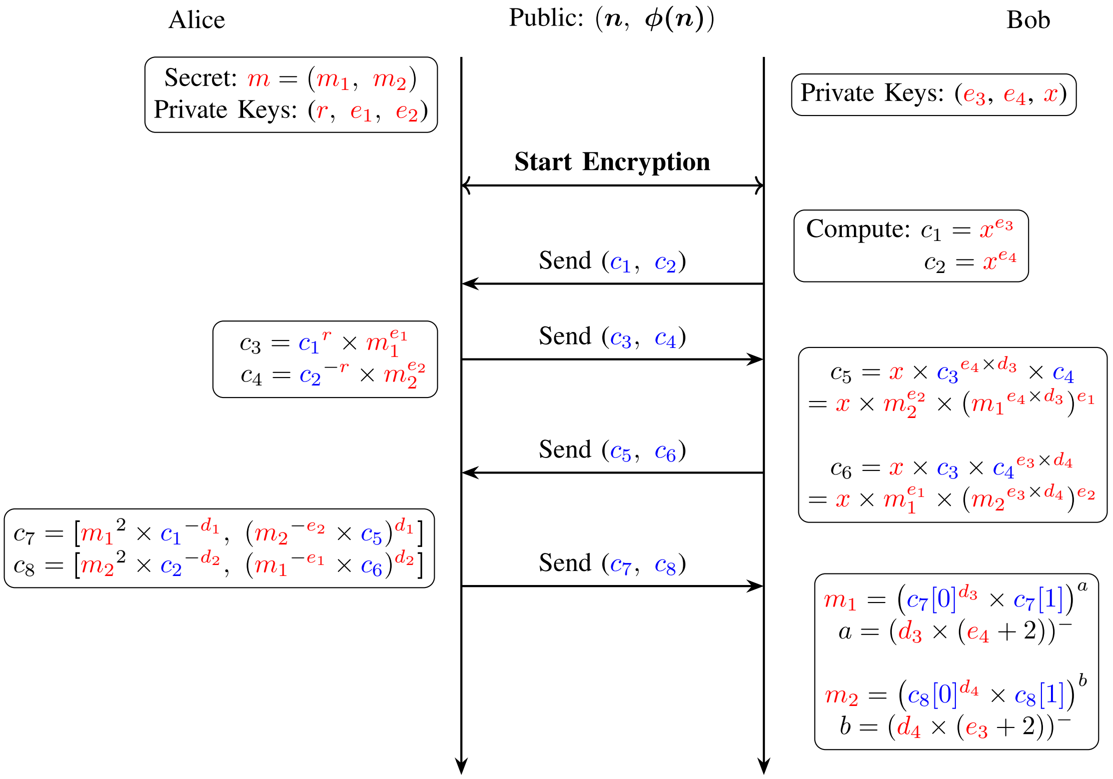

Alice
Public: $(\boldsymbol{n},\ \boldsymbol{\phi(n)})$
Bob
Secret: $m = (m_1,\ m_2)$
Private Keys: $(r,\ e_1,\ e_2)$
Private Keys: $(e_3,\ e_4,\ x)$
Start Encryption
Compute: $c_1 = x^{e_3}$
$c_2 = x^{e_4}$
Send $(c_1,\ c_2)$
$c_3 = {c_1}^{r} \times m_1^{e_1}$
$c_4 = {c_2}^{-r} \times m_2^{e_2}$
Send $(c_3,\ c_4)$
$c_5 = x \times {c_3}^{e_4 \times d_3} \times c_4$
$= x \times m_2^{e_2} \times ({m_1}^{e_4 \times d_3})^{e_1}$
$c_6 = x \times c_3 \times {c_4}^{e_3 \times d_4}$
$= x \times m_1^{e_1} \times ({m_2}^{e_3 \times d_4})^{e_2}$
Send $(c_5,\ c_6)$
$c_7 = [{m_1}^2 \times {c_1}^{-d_1},\ ({m_2}^{-e_2} \times c_5)^{d_1}]$
$c_8 = [{m_2}^2 \times {c_2}^{-d_2},\ ({m_1}^{-e_1} \times c_6)^{d_2}]$
Send $(c_7,\ c_8)$
$m_1 = \left(c_7[0]^{d_3} \times c_7[1]\right)^{a}$
$a = (d_3 \times (e_4 + 2))^{-}$
$m_2 = \left(c_8[0]^{d_4} \times c_8[1]\right)^{b}$
$b = (d_4 \times (e_3 + 2))^{-}$

Fig. 4: Post Quantum Independent Key Encryption (Q-IKE).

making $\phi(n)$ public shifts the challenge to another problem, namely the discrete logarithm problem.

DLP involves finding x in the equation $g^x = y$ when g and y are known. Several studies prove it is possible to solve this problem using future generations of quantum computers. This directly affects the most famous and widely used encryption techniques, such as RSA; since e is public, an attacker can choose a message m and compute $c = m^e$, he knows now that $c^d = m$, where d is the secret key.

Quantum IKE claims that x and y are unknown, so the adversary knows only z in $x^y = z$. We will analyze this problem in the presence of quantum computers. The famous quantum algorithm for DLP is Shor's algorithm.

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

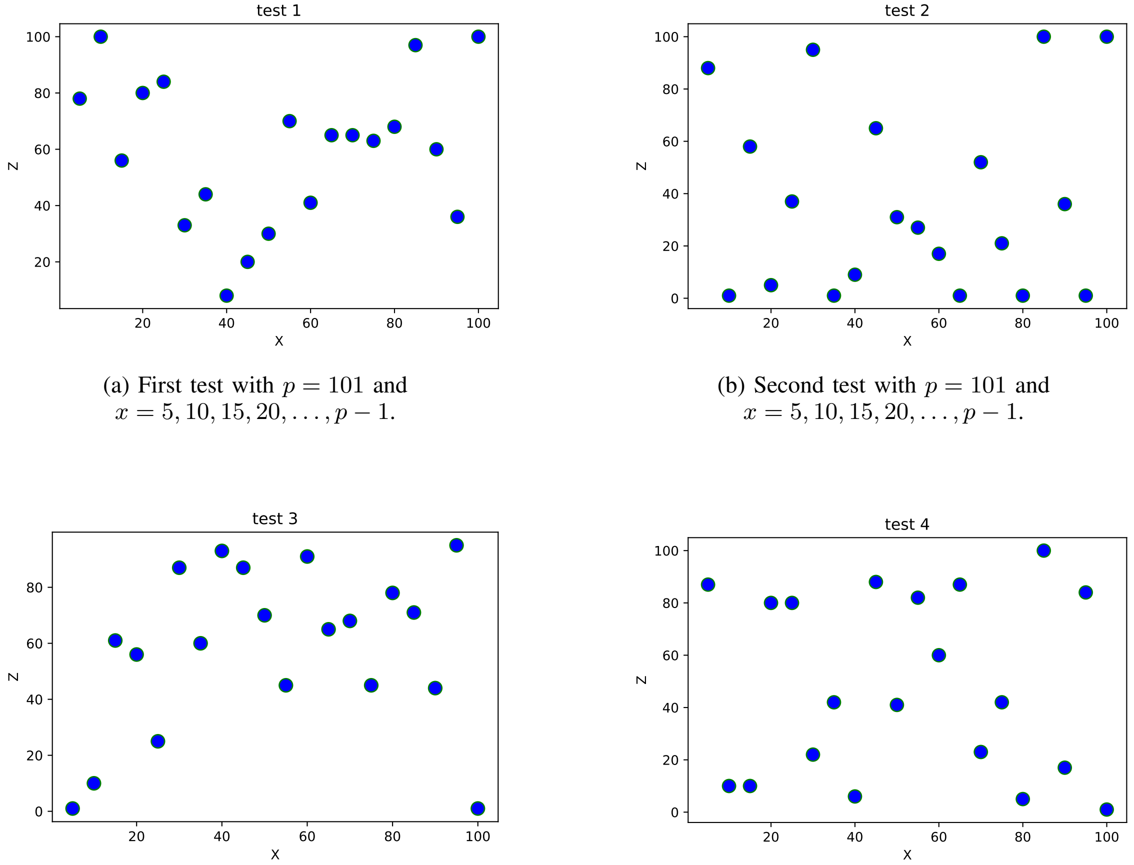
$$g^{x_1} \times y^{x_2} = g^{x_1+\omega_1} \times y^{x_2+\omega_2} \tag{2}$$

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) which 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.



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

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

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

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

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

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

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 useless even if the adversary can get all solutions (x_i, y_i) because the adversary cannot check which of these pairs is the correct one.

Figure 6 shows the number of samples. For example, if $x = 10$, we have an average of 7.80% where $z_i = 10^{y_i}(y_i = 2, p)$, meaning that we got about 7.80% samples of 10^2 and 10^3 and so on. For p = 37, the overall average of samples (oas) for any encrypted message is 4.23% (4.23% from $(p-2)^2$). We notice that when p varies, this ratio changes, for example, (p, oas) : (37, 2.56), (43, 2.32), (53, 1.88), (63, 1.36), (79, 1.25), etc.

The quantum IKE presents two types of exchanged data, either of the form $z = x_1^{x_2}$ or of the form $z' = x_1 \times y_1^{x_2}$, where x_i denotes unknown value and y_i denotes known value. In both cases, whatever the value of z is, there is $z_1 = z$ where $z_1 = x_3^{x_4}$ with $x_3 \neq x_1$ and/or $x_4 \neq x_2$;

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

y x	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

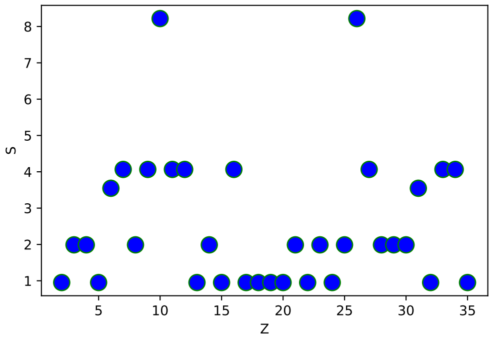


Fig. 6: Number of samples; S: # of z, $z = x^y$; $x = 2, p$, $y = 2, p$ for $p = 37$.

respectively, $z'_1 = z'$.

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 indeed an intriguing variant of the DLP, 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.

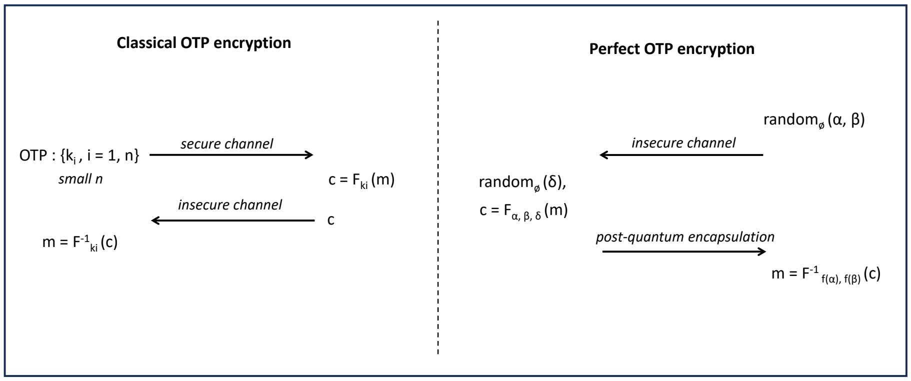


Fig. 7: Classical OTP encryption vs. proposed perfect OTP encryption

3) The key must never be reused in whole or part, as any reuse also introduces patterns that compromise secrecy.

Hence, using a one-time pad (OTP) ensures unparalleled 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 each specific 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, IKE cryptosystem ensures confidentiality and provides an added layer of protection against potential cryptographic attacks.

Figure 7 proves that a perfect OTP encryption is achieved without any need for a pre-sharing keys process, unlike the classical OTP. In IKE, 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

Q-IKE offers robust encryption, certainly, when the following preferable configurations are considered.

To provide a large key pool, it is preferred to use n a prime number of the form $n = p \times 2^i + 1$ to obtain $\phi(n) = p \times 2^i$ which means that all odd numbers not multiple of p are coprime with $\phi(n)$. Setting $p = 1$ offers the best configuration in Q-IKE to encrypt any $1 < m < n$ with a key pool equals to the half of $\phi(n)$. However, the are only five known prime numbers of the form $2^i + 1$. Another form of n offers a suitable configuration and avoids the challenge of finding large primes, n could be 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 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-IKE mitigates the pool's size of x and m. Setting $n = (65537)^j$ means that only m and x are not multiples of 65537 could not be encrypted and used as an SMH, respectively.

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

V. Conclusion

With quantum attack threats, IKE focuses on the Q-Problem that shows a complex and fascinating variant of the traditional DLP by delving into the computational intricacies of solving for both the base and the exponent within the realm of modular exponentiation. This broader and more flexible problem scope, especially the allowance for multiple solutions and the non-requirement of x being a generator, opens 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.