Design and Analysis of Four-State Quantum Public-Key Encryption Scheme

Rahul Kumar[1] and Vikas Srivastava[1*]

[1*]Department of Mathematics, National Institute of Technology Warangal, Telangana, 506004, India.

*Corresponding author(s). E-mail(s): vsv@nitw.ac.in;
Contributing authors: rk25mar2r14@student.nitw.ac.in;

Abstract

Quantum public-key encryption (QPKE) is an important direction for secure communication in the presence of quantum adversaries. In this paper, we analyze the four-state QPKE scheme of Liu et al. and show that its ciphertext structure leaks information about computational-basis plaintexts. We present a ciphertext-leakage attack in which an adversary, without knowing the private key, measures the quantum ciphertext component and combines the result with the exposed classical correction bit to recover the plaintext. To overcome this limitation, we propose sQPKE, a simple quantum public-key encryption scheme. The sQPKE scheme uses only elementary operations such as XOR, parity computation, CNOT, Hadamard, Pauli-$\boldsymbol{Y}$ gates, and computational-basis measurements. We prove correctness, analyze security against ciphertext-leakage, eavesdropping, and distinguishing attacks, and validate the attack and proposed construction through Qiskit implementation and resource estimation.

Keywords: Quantum public-key encryption, Quantum cryptography, Ciphertext-leakage attack, Four-state quantum encryption, Quantum cryptanalysis

MSC Classification: 81P94 , 94A60 , 68P25 , 68P27 , 81P65

1 Introduction

Public-key encryption (PKE) [1] is an asymmetric cryptographic primitive in which a pair of mathematically related keys is used for secure communication. The public key is used for encryption, while the corresponding private key is used for decryption. This

mechanism allows two parties to exchange confidential information without first sharing a secret key through a secure channel. Due to this property, public-key encryption has become a fundamental component of modern secure communication systems, including HTTPS, e-commerce, digital payments, electronic voting, and secure key distribution [2–5].

Classical public-key cryptosystems are usually based on computational hardness assumptions such as integer factorization problem and discrete logarithmic problem. However, the security of such schemes is threatened by quantum computation. Shor's algorithm [6] can solve integer factorization and discrete logarithm problems in polynomial time on a sufficiently powerful quantum computer. Therefore, the development of cryptographic schemes that remain secure in the presence of quantum adversaries has become an important research direction.

Quantum cryptography [7] provides a different approach to public-key encryption by using quantum states as part of the cryptographic mechanism. In 2000, Okamoto et al. [8] introduced the notion of quantum public-key cryptosystems. Gottesman [9] introduced the first quantum public-key encryption protocol. Yang et al. [10] proposed a QPKE protocol with information-theoretic security based on Einstein–Podolsky–Rosen pairs. The construction includes bit-oriented and two-bit-oriented encryption schemes, which can be extended to multi-bit encryption through a suitable public-key algorithmic structure. Nikolopoulos [11] proposed a QPKE scheme based on a quantum trapdoor function constructed from single-qubit rotations. In this scheme, integer values are mapped to quantum states of a qubit, and the difficulty of inversion is related to fundamental limitations imposed by quantum mechanics, such as Holevo's theorem. Nikolopoulos et al. [12] analyzed deterministic quantum public-key encryption and introduced the notion of a forward-search attack in this setting. They showed that deterministic quantum-public-key bit-encryption schemes are vulnerable to such attacks. However, they also showed that a deterministic scheme can be used as a building block for constructing a randomized bit-encryption scheme that avoids this weakness.

Liang et al. [13] proposed a QPKE protocol for classical messages that can also provide authentication for quantum messages. Their scheme is information-theoretically secure with respect to encryption, and the probability of successful tampering decreases exponentially with the security parameter. Wu et al. [14] proposed a bit-oriented quantum public-key encryption scheme in which Boolean functions are used as private keys and randomly changed pairs of quantum states and classical strings are used as public keys. Later, Wu et al. [15] proposed a QPKE scheme based on Bell states, in which the public key is used for encryption and the private key is used for decryption. Yang et al. [16] proposed a QPKE scheme based on conjugate coding and studied related authentication issues. Their work also provided an improved security analysis and argued that the protocol can achieve information-theoretic security under the considered attacks. Wang [17] proposed a practical model for quantum public-key encryption, giving explicit requirements for key generation, key distribution, authentication, and usage of secret keys. Liu et al. [18] proposed a quantum public-key encryption scheme in which the public key is represented using four quantum states. The scheme uses quantum-state properties to support secure message encryption and decryption. Kitagawa et al. [19] proposed a quantum public-key encryption scheme with tamper-resilient public keys

based on one-way functions. Their construction protects quantum public keys against malicious modifications and provides stronger security in practical communication environments. Barooti et al. [20] introduced a public-key encryption framework in which the public key is a quantum state. Their work established formal security definitions and constructions for encryption using quantum public keys. Malavolta and Walter [21] proposed a robust quantum public-key encryption scheme that remains secure even when the quantum public key is modified or corrupted. They also demonstrated its application to the design of secure quantum key distribution protocols. Grilo et al. [22] studied encryption schemes that use quantum states as public keys. They presented security models and constructions that highlight the advantages and challenges of quantum public-key encryption. Wu et al. [23] designed a secure quantum sealed-bid auction protocol based on quantum public-key encryption. Their protocol aims to preserve the privacy of bidders while ensuring fairness and security during the auction process. Recently, advanced encryption protocols have also been developed for specialized security requirements. Coladangelo et al. [24] proposed deniable encryption for the quantum setting, while Rao et al. [25] designed a quantum-secure attribute-based encryption scheme for fine-grained access control.

Despite these advances, the security of concrete quantum public-key encryption constructions requires careful analysis. In particular, we examine the four-state quantum public-key encryption scheme proposed by Liu et al. [18] and identify a vulnerability that allows an adversary to recover plaintexts by measuring the transmitted quantum state. Motivated by the growing interest in encryption with quantum public keys, this paper develops a new quantum public-key encryption scheme, called sQPKE. The sQPKE also address this limitations of [18]. We formally establish the correctness and security properties of the construction and provide a complete implementation and experimental evaluation to demonstrate its feasibility.

1.1 Our Contribution

The main contributions of this paper are summarized as follows.

- We provide a detailed security analysis of the four-state quantum public-key encryption scheme proposed in [18]. We show that the ciphertext produced by encryption algorithm in [18] contains a classical measurement bit that is directly useful to an adversary. By measuring the quantum ciphertext component in the computational basis and combining the result with this classical bit, an adversary can recover the plaintext whenever the plaintext is a computational-basis state. This attack breaks the confidentiality claim of the scheme. The attack does not require the private key and works independently of which of the four public-key states is used.
- To address the identified leakage, we propose a new quantum public-key encryption scheme namely sQPKE. sQPKE preserves the simple four-state quantum encryption structure while incorporating a parity-based quantum register that encodes the correction information required for decryption. In the sQPKE construction, the correction bit is not sent as an exposed classical ciphertext component. Instead, it is encoded into a quantum state whose recovery requires knowledge of the secret key.

- The sQPKE uses only elementary classical and quantum operations. The classical part requires only XOR and parity computations. The quantum part uses basic gates such as the Hadamard gate, Pauli-Y gate, CNOT gate, and computational-basis measurement. The construction avoids complicated algebraic operations, and expensive quantum subroutines. This makes the scheme simple, easy to implement, and suitable for near-term quantum circuit simulation.
- We proved the correctness of the sQPKE and provide a detailed security analysis in Section 6. In particular, we provide a detailed analysis on the security of the parity register, the security of the quantum ciphertext component, confidentiality of the complete ciphertext, resistance to ciphertext-leakage attacks, security of the secret key, and resistance to eavesdropping attacks.
- We also implement the ciphertext-leakage attack against the scheme of Liu et al. [18] using Qiskit. The implementation confirms that for computational-basis plaintexts, the adversary can recover the plaintext with probability one by measuring the ciphertext qubit and combining the result with the exposed measurement bit. We also implement the sQPKE construction and verify its correctness through quantum circuit simulation. Refer Section 7.
- We analyze the computational and quantum resources required by the sQPKE. The resource estimation includes the number of qubits, the number of elementary gates, the number of measurements, and the classical operations required for encryption and decryption. Since the sQPKE design relies only on simple gates and parity computations, its circuit depth and gate complexity remain low. This makes sQPKE efficient compared with schemes that require more complex quantum transformations or computationally expensive classical operations.

1.2 Paper Organization

Section 2 introduces the preliminaries that are useful for describing and analyzing the paper. The protocol of [18] is summarized in Section 3. The proposed attack strategy for cryptanalysis of [18] is provided in Section 4. The proposed Quantum Public-Key Encryption scheme, namely sQPKE is presented in Section 5. In Section 6 we discussed the security analysis of our paper. Section 7 describes the implementation and experimental validation.The efficiency analysis and resource estimation is covered in Section 8. Manuscript is concluded in Section 9

2 Preliminaries

In this section, we recall some basic notions of quantum computation that are used in the sQPKE scheme. The fundamental unit of information in a classical computer is a bit, which can take one of two possible values, either 0 or 1. In quantum computation, the corresponding unit of information is called a qubit. We use Dirac notation to denote quantum states. A single-qubit state is represented as $|\rho\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$ Here, $\mathbb{C}$ denotes the field of complex numbers. The states $|0\rangle$ and $|1\rangle$ form the computational basis and are represented by the column vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Thus, the state $|\rho\rangle = \alpha|0\rangle + \beta|1\rangle$ can also be written

as $|\rho\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$. More generally, an n-dimensional complex vector can be written as $|a\rangle = \begin{pmatrix}a_1 & a_2 & \cdots & a_n\end{pmatrix}^T \in \mathbb{C}^n$. For multi-qubit systems, the joint state is described using the tensor product. For example, the two-qubit computational basis states are $|00\rangle = |0\rangle \otimes |0\rangle$, $|01\rangle = |0\rangle \otimes |1\rangle$, $|10\rangle = |1\rangle \otimes |0\rangle$, $|11\rangle = |1\rangle \otimes |1\rangle$. The Hadamard-basis states are defined as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

2.1 Pauli Operators

The Pauli operators are fundamental single-qubit unitary operators. They are denoted by I, X, Y, and Z, and are represented by the matrices

$$I = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \qquad X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \qquad Y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}, \qquad Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}.$$

The operator X acts as a bit-flip operation: $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$. The operator Z acts as a phase-flip operation: $Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle$. The operator Y flips the computational-basis state up to a phase: $Y|0\rangle = i|1\rangle, Y|1\rangle = -i|0\rangle$. *Notation.* Throughout the remainder of the paper, we write the Pauli operators X, Y, Z interchangeably as σ_x, σ_y, σ_z (so $\sigma_x \equiv X$, $\sigma_y \equiv Y$, $\sigma_z \equiv Z$, as defined above), and correspondingly write σ_x^j, σ_y^j, σ_z^j for $X^{j_1} \otimes \cdots \otimes X^{j_n}$, $Y^{j_1} \otimes \cdots \otimes Y^{j_n}$, and $Z^{j_1} \otimes \cdots \otimes Z^{j_n}$, respectively, acting on an n-qubit register indexed by a bit string $j = (j_1, \ldots, j_n) \in \{0,1\}^n$. The identity operator is always written as I.

2.2 Quantum Gates

Quantum operations are represented by unitary matrices. A quantum gate acting on n qubits is represented by a $2^n \times 2^n$ unitary matrix. The Hadamard gate is a single-qubit gate given by $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. It maps the computational basis to the Hadamard basis: $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$. The phase gate is defined as $P = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$. We also use two-qubit gates. The controlled-NOT gate, denoted by CNOT, acts on two qubits. The first qubit is the control qubit and the second qubit is the target qubit. If the control qubit is $|1\rangle$, the target qubit is flipped; if the control qubit is $|0\rangle$, the target qubit remains unchanged. Its action on the computational basis is

$$\text{CNOT}|00\rangle = |00\rangle, \qquad \text{CNOT}|01\rangle = |01\rangle,$$

$$\text{CNOT}|10\rangle = |11\rangle, \qquad \text{CNOT}|11\rangle = |10\rangle.$$

The matrix representation of the CNOT gate is

$$\text{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}.$$

Table 1 Correction operators used during decryption

Private-key index	$\mu = 0$	$\mu = 1$
$d = 0$	$P^\dagger$	$XP^\dagger$
$d = 1$	P	XP
$d = 2$	I	X
$d = 3$	Z	XZ

2.3 Measurement

A measurement of a qubit in the computational basis gives outcome 0 with probability $|\alpha|^2$ and outcome 1 with probability $|\beta|^2$, where $|\rho\rangle = \alpha|0\rangle + \beta|1\rangle$. After the measurement, the state collapses to the basis state corresponding to the observed outcome. This measurement rule is used in the encryption and decryption procedures of the sQPKE.

2.4 Correction Operators

Let $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, P^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$ denote the phase gate and its inverse, respectively. For an index $d \in \{0, 1, 2, 3\}$ and $\mu \in \{0, 1\}$, define the single-qubit correction operator $R_{d,\mu}$ as

$$R_{d,\mu} = \begin{cases} P^\dagger, & d = 0,\ \mu = 0, \\ XP^\dagger, & d = 0,\ \mu = 1, \\ P, & d = 1,\ \mu = 0, \\ XP, & d = 1,\ \mu = 1, \\ I, & d = 2,\ \mu = 0, \\ X, & d = 2,\ \mu = 1, \\ Z, & d = 3,\ \mu = 0, \\ XZ, & d = 3,\ \mu = 1. \end{cases}$$

The operators are composed according to the standard right-to-left convention. For example, $XP^\dagger$ means that $P^\dagger$ is applied first, followed by X. The correction operators are summarized in Table 1.

3 Description of Liu et al.'s Quantum Public-Key Encryption Scheme

We describe the QPKE scheme of Liu et al. [18] in a compact form. The scheme consists of three algorithms: key generation, encryption, and decryption. The private key is classical, whereas the public key is a string of quantum states. The plaintext is an n-qubit product state and encryption is performed qubit-wise using the corresponding public-key qubit.

Let $K_p = (K_{p,0}, K_{p,1}, \ldots, K_{p,n-1})$ be the private key, where each component is chosen

independently and uniformly from $\{k_0, k_1, k_2, k_3\}$. The corresponding public key is an n-qubit state

$$|K_v\rangle = \bigotimes_{i=0}^{n-1} |K_{v,i}\rangle ,$$

where each public-key qubit is determined by the corresponding private-key symbol according to the following rule:

Private key $K_{p,i}$	Preparation	Public-key state $\|K_{v,i}\rangle$
k_0	$PH\|0\rangle$	$\|P_+\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + i\|1\rangle)$
k_1	$XPH\|0\rangle$	$\|P_-\rangle = \frac{1}{\sqrt{2}}(i\|0\rangle + \|1\rangle)$
k_2	$H\|0\rangle$	$\|+\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
k_3	$HX\|0\rangle$	$\|-\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$

Here X is the Pauli-X gate, H is the Hadamard gate, and P denotes the phase gate $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$. The plaintext is an n-qubit product state

$$|M\rangle = \bigotimes_{i=0}^{n-1} |M_i\rangle , \qquad |M_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle ,$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1$. The ciphertext consists of a quantum part

$$|M_{\text{en}}\rangle = \bigotimes_{i=0}^{n-1} |M_{\text{en},i}\rangle$$

and a classical string $m = (m_0, m_1, \ldots, m_{n-1}) \in \{0,1\}^n$.

Key Generation. The receiver chooses the private key $K_p \leftarrow \{k_0, k_1, k_2, k_3\}^n$ uniformly at random. For each i, the receiver prepares the corresponding public-key qubit $|K_{v,i}\rangle$ using the rule above. Thus,

$$|K_v\rangle = \bigotimes_{i=0}^{n-1} |K_{v,i}\rangle , \qquad |K_{v,i}\rangle \in \{|P_+\rangle , |P_-\rangle , |+\rangle , |-\rangle\}.$$

The quantum public key is then distributed to the sender through a secure authenticated channel. The same private key may be used to prepare multiple copies of the public key.

Encryption. To encrypt the plaintext qubit $|M_i\rangle$, the sender uses the corresponding public-key qubit $|K_{v,i}\rangle$. The sender applies a CNOT gate with $|K_{v,i}\rangle$ as the control qubit and $|M_i\rangle$ as the target qubit. The target qubit is then measured in the computational basis. The measurement outcome is denoted by $m_i \in \{0, 1\}$. The remaining control qubit becomes the encrypted quantum state $|M_{\mathrm{en},i}\rangle$. Therefore, the encryption of the ith plaintext qubit is

$$|K_{v,i}\rangle \otimes |M_i\rangle \xrightarrow{\text{CNOT}} \text{entangled state} \xrightarrow{\text{measure target}} |M_{\mathrm{en},i}\rangle, m_i.$$

The full ciphertext is $(|M_{\mathrm{en}}\rangle, m)$, where

$$|M_{\mathrm{en}}\rangle = \bigotimes_{i=0}^{n-1} |M_{\mathrm{en},i}\rangle, \qquad m = (m_0, m_1, \ldots, m_{n-1}).$$

For a plaintext qubit $|M_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle$, the possible encrypted states are as follows:

$K_{p,i}$	$m_i = 0$	$m_i = 1$
k_0	$\alpha_i \|0\rangle + i\beta_i \|1\rangle$	$\beta_i \|0\rangle + i\alpha_i \|1\rangle$
k_1	$i\alpha_i \|0\rangle + \beta_i \|1\rangle$	$i\beta_i \|0\rangle + \alpha_i \|1\rangle$
k_2	$\alpha_i \|0\rangle + \beta_i \|1\rangle$	$\beta_i \|0\rangle + \alpha_i \|1\rangle$
k_3	$\alpha_i \|0\rangle - \beta_i \|1\rangle$	$\beta_i \|0\rangle - \alpha_i \|1\rangle$

Decryption. The receiver decrypts each encrypted qubit using the private-key symbol $K_{p,i}$ and the classical measurement bit m_i. The required correction operations are given below. The gates are written in the order in which they are applied.

$K_{p,i}$	m_i	Correction applied to $\|M_{\mathrm{en},i}\rangle$
k_0	0	$P,\ Z$
k_0	1	$P,\ Z,\ X$
k_1	0	$X,\ P,\ Z,\ X$
k_1	1	$X,\ P,\ Z$
k_2	0	I
k_2	1	X
k_3	0	Z
k_3	1	$X,\ Z$

Equivalently, for every i, the receiver applies the correction determined by $(K_{p,i}, m_i)$ and obtains $|M_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle$. Thus, applying the decryption procedure independently to all ciphertext qubits gives

$$|M\rangle = \bigotimes_{i=0}^{n-1} |M_i\rangle.$$

4 The Proposed Attack

In this section, we show that the Liu et al.'s [18] design leaks information about the plaintext. Consider one message qubit. Let the public-key qubit be $|K_v\rangle = a|0\rangle + b|1\rangle$, where $|a|^2 + |b|^2 = 1$. In the proposed scheme, the public-key qubit is chosen from one of the four states

$$\frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \qquad \frac{i|0\rangle + |1\rangle}{\sqrt{2}}, \qquad \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Thus every public-key qubit has the general form $a|0\rangle + b|1\rangle$. During encryption, the sender applies a CNOT gate, where the public-key qubit is the control qubit and the message qubit is the target qubit. After the CNOT operation, the target qubit is measured in the computational basis. Let the measurement outcome be $m \in \{0, 1\}$. The remaining first qubit is denoted by $|M_{\text{en}}\rangle$. Therefore, the ciphertext corresponding to one message qubit is $C = (|M_{\text{en}}\rangle, m)$. Here $|M_{\text{en}}\rangle$ is the quantum part of the ciphertext and m is the classical part of the ciphertext. The bit m is necessary for decryption, and hence it is available to the receiver.

4.1 Attack Algorithm

We now show that the ciphertext reveals whether the plaintext is $|0\rangle$ or $|1\rangle$. First, suppose that the plaintext is

$$|M\rangle = |0\rangle.$$

The initial two-qubit state is

$$|K_v\rangle|M\rangle = (a|0\rangle + b|1\rangle)|0\rangle = a|00\rangle + b|10\rangle.$$

After applying the CNOT gate, we obtain

$$a|00\rangle + b|10\rangle \longmapsto a|00\rangle + b|11\rangle.$$

The second qubit is then measured. If the measurement result is $m = 0$, the remaining first qubit becomes $|0\rangle$. If the measurement result is $m = 1$, the remaining first qubit becomes $|1\rangle$. Hence, for plaintext $|0\rangle$, the possible ciphertexts are

$$(|0\rangle, 0) \qquad \text{and} \qquad (|1\rangle, 1).$$

Thus, if an adversary measures the quantum ciphertext component $|M_{\text{en}}\rangle$ in the computational basis and obtains the outcome t, then

$$t = m.$$

Now suppose that the plaintext is

$$|M\rangle = |1\rangle.$$

The initial two-qubit state is

$$|K_v\rangle|M\rangle = (a|0\rangle + b|1\rangle)|1\rangle = a|01\rangle + b|11\rangle.$$

After applying the CNOT gate, we obtain

$$a|01\rangle + b|11\rangle \longmapsto a|01\rangle + b|10\rangle.$$

Again, the second qubit is measured. If the measurement result is $m = 0$, the remaining first qubit becomes $|1\rangle$. If the measurement result is $m = 1$, the remaining first qubit becomes $|0\rangle$. Hence, for plaintext $|1\rangle$, the possible ciphertexts are

$$(|1\rangle, 0) \qquad \text{and} \qquad (|0\rangle, 1).$$

Therefore, if the adversary measures $|M_{\text{en}}\rangle$ in the computational basis and obtains the outcome t, then

$$t \neq m.$$

Consequently, the adversary can recover the plaintext bit by computing

$$x = t \oplus m.$$

If $x = 0$, the plaintext was $|0\rangle$. If $x = 1$, the plaintext was $|1\rangle$. The ciphertext-leakage attack can be formalized as Algorithm 1. The adversary uses only the intercepted ciphertext and does not require the private key.

Algorithm 1 Ciphertext-leakage attack on one encrypted qubit

Require: Ciphertext $C = (|M_{\text{en}}\rangle, m)$, where $|M_{\text{en}}\rangle$ is the quantum ciphertext component and $m \in \{0, 1\}$ is the classical measurement bit.

Ensure: Recovered computational-basis plaintext $|x\rangle$.

1: Measure $|M_{\text{en}}\rangle$ in the computational basis.

2: Let the measurement outcome be $t \in \{0, 1\}$.

3: Compute

$$x \leftarrow t \oplus m.$$

4: **return** $|x\rangle$.

For computational-basis plaintexts, the output of Algorithm 1 is always correct. Indeed, if the plaintext is $|0\rangle$, then the measurement outcome t of the encrypted qubit satisfies $t = m$, and hence $x = t \oplus m = 0$. On the other hand, if the plaintext is $|1\rangle$, then $t \neq m$, and hence $x = t \oplus m = 1$. Therefore, the adversary recovers the plaintext with probability one.

4.2 Toy Example

We illustrate the attack using the private-key value k_2. In this case, the corresponding public-key qubit is

$$|K_v\rangle = |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

If the plaintext is $|0\rangle$, then the initial state is

$$|+\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right).$$

After applying CNOT, the state becomes

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right).$$

After measuring the second qubit, the possible ciphertexts are

$$(|0\rangle, 0) \qquad \text{and} \qquad (|1\rangle, 1).$$

In both cases, the measurement outcome of the encrypted qubit is equal to the classical ciphertext bit m. If the plaintext is $|1\rangle$, then the initial state is

$$|+\rangle|1\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |11\rangle\right).$$

After applying CNOT, the state becomes

$$\frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right).$$

After measuring the second qubit, the possible ciphertexts are

$$(|1\rangle, 0) \qquad \text{and} \qquad (|0\rangle, 1).$$

In both cases, the measurement outcome of the encrypted qubit is different from the classical ciphertext bit m. Therefore, by measuring the encrypted qubit and comparing the result with m, an adversary can determine whether the plaintext was $|0\rangle$ or $|1\rangle$.

4.3 Indistinguishability Attack

The above observation directly gives an indistinguishability attack. The adversary chooses two plaintexts $M_0 = |0\rangle, \qquad M_1 = |1\rangle$. The challenger chooses a random bit $\beta \in \{0, 1\}$ and returns an encryption of M_β. Let the challenge ciphertext be $C^* = (|M^*_{\text{en}}\rangle, m^*)$. The adversary measures $|M^*_{\text{en}}\rangle$ in the computational basis and

obtains t^*. It then computes $\beta' = t^* \oplus m^*$. From the analysis above, if $\beta = 0$, then $t^* = m^*$ and hence $\beta' = 0$. If $\beta = 1$, then $t^* \neq m^*$ and hence $\beta' = 1$. Therefore,

$$\Pr[\beta' = \beta] = 1.$$

Thus, the adversary distinguishes encryptions of $|0\rangle$ and $|1\rangle$ with probability one. Hence, the scheme does not satisfy indistinguishability for quantum plaintexts, since $|0\rangle$ and $|1\rangle$ are valid plaintext states.

4.4 Extension to Multi-Qubit Messages

The same attack applies independently to each qubit of an n-qubit computational-basis message. Let $|M\rangle = |x_1 x_2 \cdots x_n\rangle$, where $x_i \in \{0, 1\}$ for each i. For the i-th ciphertext component, write $C_i = \big(|M_{\mathrm{en}(i)}\rangle, m_i\big)$. The adversary measures $|M_{\mathrm{en}(i)}\rangle$ in the computational basis and obtains t_i. Then the i-th plaintext bit is recovered as $x_i = t_i \oplus m_i$. Therefore, the entire classical basis message is recovered as $x_1 x_2 \cdots x_n = (t_1 \oplus m_1)(t_2 \oplus m_2) \cdots (t_n \oplus m_n)$.

5 Proposed Quantum Public-Key Encryption Scheme sQPKE

Technical Overview. The proposed scheme consists of three phases: key generation, encryption, and decryption. During key generation, the receiver prepares a *four-state quantum public-key register* together with an auxiliary quantum register. The auxiliary register is generated from randomly selected even-parity binary strings, which are transformed using tensor-product Hadamard operations determined by a secret function. The resulting quantum states, together with the associated public labels, constitute the public key, while the four-state indices and the secret function form the private key. For each plaintext qubit, the sender applies a CNOT gate using the corresponding four-state public-key qubit as the control and the plaintext qubit as the target. The target qubit is subsequently measured in the computational basis, producing a one-bit measurement outcome that is later required during decryption. Instead of transmitting this syndrome explicitly, it is encoded into the auxiliary quantum register by selecting a random binary string with the required parity and applying tensor-product Pauli-Y operations. The idea of quantum parity register was introduced in [10]. During decryption, the receiver reconstructs the tensor-product Hadamard transformation using the secret function, applies it to the auxiliary register, and performs computational-basis measurements. The parity of the measured string recovers the correction syndrome, which is then combined with the private four-state index to determine the appropriate recovery operator for the encrypted data qubit. The entire construction relies only on CNOT gates, Hadamard gates, Pauli-Y operators, computational-basis measurements, XOR operations, and parity computations, resulting in a simple quantum public-key encryption scheme.
We now describe the proposed design sQPKE in detail. Let ℓ denote the number of message qubits. Let n denote the length of the auxiliary parity block used to encode one classical bit.

Key Generation. Alice generates the secret key and the public key as follows.

1. First, Alice generates the private key for the quantum encryption component. For each $1 \leq r \leq \ell$, she chooses
$$d_r \in \{0, 1, 2, 3\}$$
uniformly at random. The value d_r determines the corresponding quantum public-key state. More precisely, define
$$\Phi_0 = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \qquad \Phi_1 = \frac{i|0\rangle + |1\rangle}{\sqrt{2}},$$
$$\Phi_2 = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad \Phi_3 = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$
For each r, Alice prepares $|V_r\rangle = \Phi_{d_r}$. These states form the first quantum component of the public key.
2. Next, Alice chooses a keyed Pseudo Random Function F $F : \{0,1\}^n \to \{0,1\}^n$. For each $1 \leq r \leq \ell$, Alice chooses a random string $s_r \in \{0,1\}^n$ and computes $k^{(r)} = F(s_r)$. Write $k^{(r)} = (k_1^{(r)}, \ldots, k_n^{(r)})$.
3. Alice also chooses a random even-parity string
$$i^{(r)} = (i_1^{(r)}, \ldots, i_n^{(r)}) \in \{0,1\}^n$$
such that $i_1^{(r)} \oplus i_2^{(r)} \oplus \cdots \oplus i_n^{(r)} = 0$. She encodes this string in the computational basis as $|i^{(r)}\rangle = |i_1^{(r)}\rangle \otimes \cdots \otimes |i_n^{(r)}\rangle$.
4. Then she applies $H^{k^{(r)}} = H^{k_1^{(r)}} \otimes \cdots \otimes H^{k_n^{(r)}}$ and obtains the auxiliary public-key state
$$|P_r\rangle = H^{k^{(r)}}|i^{(r)}\rangle.$$

Thus, Alice's secret key is $\mathsf{SK} = (d_1, \ldots, d_\ell, F)$, and the public key consists of the quantum states and public labels $\mathsf{PK} = \{(|V_r\rangle, s_r, |P_r\rangle)\}_{r=1}^{\ell}$.

Encryption. Let Bob want to encrypt an ℓ-qubit plaintext $|M\rangle = |M_1\rangle \otimes \cdots \otimes |M_\ell\rangle$, where
$$|M_r\rangle = \alpha_r|0\rangle + \beta_r|1\rangle.$$
For each $1 \leq r \leq \ell$, Bob performs the following steps.

1. Bob takes the public-key qubit $|V_r\rangle$ and the message qubit $|M_r\rangle$.
2. Applies a CNOT gate with $|V_r\rangle$ as the control qubit and $|M_r\rangle$ as the target qubit.
3. Measures the target qubit in the computational basis. Let the measurement outcome be $\mu_r \in \{0,1\}$. The remaining first qubit is denoted by $|C_r\rangle$.[1]

[1] Note that this qubit is the quantum ciphertext component corresponding to $|M_r\rangle$. The bit μ_r is necessary for decryption, but it is not transmitted directly. Instead, Bob encodes it into the auxiliary public-key component.

4. Chooses a random string $j^{(r)} = (j_1^{(r)}, \ldots, j_n^{(r)}) \in \{0,1\}^n$ such that

$$j_1^{(r)} \oplus j_2^{(r)} \oplus \cdots \oplus j_n^{(r)} = \mu_r.$$

5. Applies $\sigma_y^{j^{(r)}} = \sigma_y^{j_1^{(r)}} \otimes \cdots \otimes \sigma_y^{j_n^{(r)}}$ to the auxiliary public-key state $|P_r\rangle$. This results into

$$|D_r\rangle = \sigma_y^{j^{(r)}} |P_r\rangle = \sigma_y^{j^{(r)}} H^{k^{(r)}} |i^{(r)}\rangle.$$

The ciphertext component corresponding to the r-th plaintext qubit is $\mathsf{CT_r} = (|C_r\rangle, s_r, |D_r\rangle)$. The full ciphertext is $\mathsf{CT} = (CT_1, \ldots, CT_\ell)$.

Decryption. Alice receives the ciphertext $\mathsf{CT} = (\mathsf{CT}_1, \ldots, \mathsf{CT}_\ell)$, where the ciphertext block corresponding to the r-th message qubit is $\mathsf{CT}_r = (|C_r\rangle, s_r, |D_r\rangle)$. For each $1 \leq r \leq \ell$, Alice performs the following operations.

1. Evaluates her secret function on the public string s_r and obtains $k^{(r)} = F(s_r)$.
2. Applies $H^{k^{(r)}} = H^{k_1^{(r)}} \otimes \cdots \otimes H^{k_n^{(r)}}$ to the parity ciphertext register $|D_r\rangle$.
3. Measures all qubits of the resulting register in the computational basis. Let $z^{(r)} = (z_1^{(r)}, \ldots, z_n^{(r)})$ be the obtained measurement string.
4. Computes its parity as $\widehat{\mu}_r = z_1^{(r)} \oplus z_2^{(r)} \oplus \cdots \oplus z_n^{(r)}$.
5. Uses the private-key index d_r and the recovered correction bit $\widehat{\mu}_r$ to select the operator $R_{d_r,\widehat{\mu}_r}$ from Table 1.
6. Applies this operator to the encrypted data qubit $|C_r\rangle$ and obtains

$$|\widehat{M}_r\rangle = R_{d_r,\widehat{\mu}_r} |C_r\rangle.$$

After processing all ciphertext blocks, Alice outputs

$$|\widehat{M}\rangle = |\widehat{M}_1\rangle \otimes \cdots \otimes |\widehat{M}_\ell\rangle.$$

5.1 Correctness

We prove that an honestly generated ciphertext is decrypted to the original plaintext quantum state.

Theorem 5.1. *Let $|M\rangle = |M_1\rangle \otimes \cdots \otimes |M_\ell\rangle$ be an ℓ-qubit plaintext, where $|M_r\rangle = \alpha_r|0\rangle + \beta_r|1\rangle$. For every key pair generated by the key-generation algorithm, the proposed scheme satisfies Decryption(Encryption($|M\rangle$)) $\equiv |M\rangle$, where $\equiv$ denotes equality up to a global phase.*

Proof Fix an index $r \in \{1, \ldots, \ell\}$. We first show that Alice recovers the correction bit μ_r correctly, and then show that the selected correction operator recovers the message qubit $|M_r\rangle$. The parity ciphertext register generated during encryption is $|D_r\rangle = \sigma_y^{j^{(r)}} H^{k^{(r)}} |i^{(r)}\rangle$,

For every position $1 \le a \le n$, we have

$$H^{k_a^{(r)}} \sigma_y^{j_a^{(r)}} H^{k_a^{(r)}} = (-1)^{k_a^{(r)} j_a^{(r)}} \sigma_y^{j_a^{(r)}}.$$

Taking the tensor product over all positions gives

$$H^{k^{(r)}} \sigma_y^{j^{(r)}} H^{k^{(r)}} = (-1)^{j^{(r)} \cdot k^{(r)}} \sigma_y^{j^{(r)}}.$$

Therefore, after Alice applies $H^{k^{(r)}}$, the parity register becomes

$$H^{k^{(r)}} |D_r\rangle = (-1)^{j^{(r)} \cdot k^{(r)}} \sigma_y^{j^{(r)}} |i^{(r)}\rangle.$$

The Pauli-Y operator flips a computational-basis state up to a phase. Hence, there exists a complex number γ_r satisfying $|\gamma_r| = 1$ such that

$$\sigma_y^{j^{(r)}} |i^{(r)}\rangle = \gamma_r |i^{(r)} \oplus j^{(r)}\rangle.$$

The factors $(-1)^{j^{(r)} \cdot k^{(r)}}$ and γ_r are global phases and do not affect the measurement outcome. Consequently, Alice obtains $z^{(r)} = i^{(r)} \oplus j^{(r)}$ with probability one.
The parity computed by Alice is therefore

$$\begin{aligned}
\widehat{\mu}_r &= \text{parity}\left(z^{(r)}\right) \\
&= \text{parity}\left(i^{(r)} \oplus j^{(r)}\right) \\
&= \text{parity}\left(i^{(r)}\right) \oplus \text{parity}\left(j^{(r)}\right) \\
&= 0 \oplus \mu_r \\
&= \mu_r.
\end{aligned}$$

Thus, Alice recovers the correction bit generated during encryption.

Case I $d_r = 0$. For $d_r = 0$, the public-key qubit is $\frac{|0\rangle + i|1\rangle}{\sqrt{2}}$. The corresponding encrypted data qubit is

$$|C_r\rangle = \begin{cases} \alpha_r|0\rangle + i\beta_r|1\rangle, & \mu_r = 0, \\ \beta_r|0\rangle + i\alpha_r|1\rangle, & \mu_r = 1. \end{cases}$$

Therefore, $P^\dagger|C_r\rangle = |M_r\rangle$ when $\mu_r = 0$, and $XP^\dagger|C_r\rangle = |M_r\rangle$ when $\mu_r = 1$.

Case II, $d_r = 1$. For $d_r = 1$, the public-key qubit is $\frac{i|0\rangle + |1\rangle}{\sqrt{2}}$, and the encrypted data qubit is

$$|C_r\rangle = \begin{cases} i\alpha_r|0\rangle + \beta_r|1\rangle, & \mu_r = 0, \\ i\beta_r|0\rangle + \alpha_r|1\rangle, & \mu_r = 1. \end{cases}$$

In this case, $P|C_r\rangle = i|M_r\rangle$ when $\mu_r = 0$, and $XP|C_r\rangle = i|M_r\rangle$ when $\mu_r = 1$. The factor i is a global phase.

Case III, $d_r = 2$ For $d_r = 2$, the public-key qubit is $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$, and

$$|C_r\rangle = \begin{cases} \alpha_r|0\rangle + \beta_r|1\rangle, & \mu_r = 0, \\ \beta_r|0\rangle + \alpha_r|1\rangle, & \mu_r = 1. \end{cases}$$

Thus, $I|C_r\rangle = |M_r\rangle$ when $\mu_r = 0$, while $X|C_r\rangle = |M_r\rangle$ when $\mu_r = 1$.

Case IV, $d_r = 3$. For $d_r = 3$, the public-key qubit is $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, and

$$|C_r\rangle = \begin{cases} \alpha_r|0\rangle - \beta_r|1\rangle, & \mu_r = 0, \\ \beta_r|0\rangle - \alpha_r|1\rangle, & \mu_r = 1. \end{cases}$$

Hence, $Z|C_r\rangle = |M_r\rangle$ when $\mu_r = 0$, and $XZ|C_r\rangle = |M_r\rangle$ when $\mu_r = 1$.

Since $\widehat{\mu}_r = \mu_r$, Alice selects the correct operator R_{d_r,μ_r}. Therefore, for every $d_r \in \{0,1,2,3\}$ and $\mu_r \in \{0,1\}$,

$$R_{d_r,\widehat{\mu}_r}|C_r\rangle = e^{i\theta_r}|M_r\rangle$$

for some global phase $e^{i\theta_r}$. Applying the same argument independently to every message position gives

$$|\widehat{M}\rangle = e^{i\Theta}|M_1\rangle \otimes \cdots \otimes |M_\ell\rangle$$

for an overall global phase $e^{i\Theta}$. Therefore,

$$|\widehat{M}\rangle \equiv |M\rangle,$$

which proves the correctness of the proposed scheme. □

6 Security Analysis

In this section, we analyze the security of the proposed sQPKE in the one-time quantum public-key setting. Each quantum public-key block is generated for one encryption and is consumed during the encryption procedure. For every message position r, the private index d_r is chosen uniformly from $\{0,1,2,3\}$. The string s_r is public and $k^{(r)} = F(s_r)$ is unknown to the adversary. In the information-theoretic analysis below, F is modeled as a uniformly random function and the values s_r are taken to be distinct; consequently, the strings $k^{(r)}$ are independent and uniformly distributed. The strings $i^{(r)}$ and $j^{(r)}$ are generated independently subject to their prescribed parity conditions. For two density operators ρ_0 and ρ_1 occurring with equal prior probability, the optimal probability of distinguishing them is

$$P_{\text{dist}}(\rho_0,\rho_1) = \frac{1}{2} + \frac{1}{4}\|\rho_0 - \rho_1\|_1,$$

where $\|\cdot\|_1$ denotes the trace norm. Hence, the distinguishing advantage over random guessing is

$$\text{Adv}(\rho_0,\rho_1) = \frac{1}{4}\|\rho_0 - \rho_1\|_1.$$

Accordingly, two ciphertext ensembles are statistically indistinguishable when their trace distance is negligible in the security parameter n. Throughout this section, we analyze one ciphertext block first and omit the index r whenever no confusion arises. For a basis string $k = (k_1,\ldots,k_n)$, define

$$Q_k = Q_{k_1} \otimes \cdots \otimes Q_{k_n}, \qquad Q_{k_a} = \begin{cases} Z, & k_a = 0, \\ X, & k_a = 1. \end{cases}$$

The notation Q_k is used only for the security analysis and should not be confused with the public-key register $|P_r\rangle$.

Hiding of the Correction Syndrome . Let $\mu \in \{0,1\}$ be the correction syndrome produced by the measurement in the data-encryption step. Bob chooses $j \in \{0,1\}^n$ uniformly subject to $\text{parity}(j) = \mu$ and prepares the parity ciphertext register $|D\rangle = \sigma_y^j H^k |i\rangle$, where i is uniformly chosen from the even-parity strings.

Theorem 6.1 (Syndrome indistinguishability)**.** *Let ρ_0 and ρ_1 denote the density operators of the parity ciphertext register corresponding to $\mu = 0$ and $\mu = 1$, respectively, as viewed by an adversary that does not know k. Then*

$$\|\rho_0 - \rho_1\|_1 = 2^{1-n/2}.$$

Consequently,

$$\mathrm{Adv}_\mu = \frac{1}{4}\|\rho_0 - \rho_1\|_1 = 2^{-1-n/2},$$

which is negligible in n.

Proof For fixed k, applying H^k to $|D\rangle$ gives

$$H^k|D\rangle = H^k\sigma_y^j H^k|i\rangle = (-1)^{j\cdot k}\sigma_y^j|i\rangle.$$

Since σ_y flips a computational-basis bit up to a phase, the resulting computational string is $i \oplus j$. Because i has even parity and j has parity μ, the string $i \oplus j$ is uniformly distributed over the strings of parity μ. Therefore, for fixed k,

$$\rho_\mu^{(k)} = \frac{1}{2^n}\left(I + (-1)^\mu Q_k\right).$$

Averaging over the uniformly distributed basis string k yields

$$\rho_\mu = \frac{I}{2^n} + (-1)^\mu \frac{1}{2^n}\left(\frac{X+Z}{2}\right)^{\otimes n}.$$

Hence,

$$\rho_0 - \rho_1 = \frac{2}{2^n}\left(\frac{X+Z}{2}\right)^{\otimes n}.$$

The two eigenvalues of $(X+Z)/2$ are $\pm 1/\sqrt{2}$, and therefore

$$\left\|\left(\frac{X+Z}{2}\right)^{\otimes n}\right\|_1 = 2^{n/2}.$$

It follows that

$$\|\rho_0 - \rho_1\|_1 = 2^{1-n/2}.$$

The claimed distinguishing advantage follows from the Helstrom bound. □

Security of the Data Ciphertext Register. Let $|M\rangle = \alpha|0\rangle + \beta|1\rangle$ be a message qubit. Up to a global phase, the four data public-key states can be written as

$$|V_\omega\rangle = \frac{|0\rangle + \omega|1\rangle}{\sqrt{2}}, \qquad \omega \in \{1, -1, i, -i\}.$$

Conditioned on the measurement outcome μ, the remaining data ciphertext qubit is $|C_{0,\omega}\rangle = \alpha|0\rangle + \omega\beta|1\rangle$ for $\mu = 0$, and $|C_{1,\omega}\rangle = \beta|0\rangle + \omega\alpha|1\rangle$ for $\mu = 1$.

Theorem 6.2 (Perfect hiding of the data-register marginal)**.** *When the private index d and the measurement outcome μ are averaged over their honest distributions, the reduced density operator of the data ciphertext register is*

$$\rho_C(M) = \frac{I}{2}$$

for every plaintext qubit $|M\rangle$.

Proof For fixed $\mu = 0$, averaging over the four values of ω removes all off-diagonal terms and gives

$$A_{M,0} = \frac{1}{4}\sum_{\omega} |C_{0,\omega}\rangle\langle C_{0,\omega}| = \begin{pmatrix} |\alpha|^2 & 0 \\ 0 & |\beta|^2 \end{pmatrix}.$$

Similarly,

$$A_{M,1} = \frac{1}{4}\sum_{\omega} |C_{1,\omega}\rangle\langle C_{1,\omega}| = \begin{pmatrix} |\beta|^2 & 0 \\ 0 & |\alpha|^2 \end{pmatrix}.$$

The two measurement outcomes occur with probability $1/2$. Therefore,

$$\rho_C(M) = \frac{1}{2}A_{M,0} + \frac{1}{2}A_{M,1} = \frac{I}{2}.$$

Thus the data ciphertext register, considered by itself, is independent of the plaintext. □

Ciphertext Indistinguishability. The previous theorem concerns only the reduced state of the data ciphertext. Since the data register and the parity ciphertext register are correlated through μ, confidentiality must be proved for their joint state rather than for the two marginals separately. For a plaintext $|M\rangle = \alpha|0\rangle + \beta|1\rangle$, define $\delta_M = |\alpha|^2 - |\beta|^2$. Then $A_{M,0} = \frac{I+\delta_M Z}{2}$, $A_{M,1} = \frac{I-\delta_M Z}{2}$.

Theorem 6.3 (One-block ciphertext indistinguishability)**.** *Let $|M_0\rangle$ and $|M_1\rangle$ be any two one-qubit plaintext states, and let τ_{M_0} and τ_{M_1} be the corresponding density operators of the ciphertext block $\mathsf{CT} = (|C\rangle, s, |D\rangle)$. Under the one-time public-key model described above, $\|\tau_{M_0} - \tau_{M_1}\|_1 \le 2^{1-n/2}$. Therefore,*

$$\mathrm{Adv}_{\mathrm{ct}} \le 2^{-1-n/2},$$

which is negligible in n.

Proof The public classical label s is independent of the plaintext and may be conditioned upon without changing the trace-distance calculation. Let ρ_0 and ρ_1 be the parity-register density operators from theorem 6.1. The joint ciphertext state for a plaintext $|M\rangle$ is

$$\tau_M = \frac{1}{2}\left(A_{M,0} \otimes \rho_0 + A_{M,1} \otimes \rho_1\right).$$

Substituting

$$A_{M,0} = \frac{I + \delta_M Z}{2}, \qquad A_{M,1} = \frac{I - \delta_M Z}{2},$$

gives

$$\tau_M = \frac{1}{4}\left[I \otimes (\rho_0 + \rho_1) + \delta_M Z \otimes (\rho_0 - \rho_1)\right].$$

Hence, for two plaintexts M_0 and M_1,

$$\tau_{M_0} - \tau_{M_1} = \frac{\delta_{M_0} - \delta_{M_1}}{4} Z \otimes (\rho_0 - \rho_1).$$

Using multiplicativity of the trace norm under tensor products, $\|Z\|_1 = 2$, and $|\delta_{M_0} - \delta_{M_1}| \leq 2$, we obtain

$$\begin{aligned}\|\tau_{M_0} - \tau_{M_1}\|_1 &= \frac{|\delta_{M_0} - \delta_{M_1}|}{4}\|Z\|_1\|\rho_0 - \rho_1\|_1 \\ &\leq \|\rho_0 - \rho_1\|_1 \\ &= 2^{1-n/2}.\end{aligned}$$

The distinguishing-advantage bound follows immediately. □

Corollary 6.1 (Resistance to the ciphertext-leakage attack)**.** *For computational-basis plaintexts, the direct recovery rule $x = t \oplus \mu$ cannot be applied from the ciphertext with more than negligible advantage, because μ is not available as a public classical ciphertext component. More generally, Theorem 6.3 bounds every quantum measurement on the complete ciphertext block, and not only the particular measurement used in the ciphertext-leakage attack.*

Corollary 6.2 (Passive eavesdropping)**.** *A passive adversary that obtains an honestly generated ciphertext block but does not possess the secret key has distinguishing advantage at most $2^{-1-n/2}$ between any two one-qubit plaintexts.*

Recovery of the Hidden Basis String. The security of the parity register should not be inferred merely from the fact that its average state is close to the maximally mixed state. Instead, we directly bound the optimal probability of recovering the hidden basis string k from a single public-key parity register.

Theorem 6.4 (Basis-string recovery bound)**.** *Let k be uniformly distributed over $\{0,1\}^n$, and let the adversary receive one public-key parity register*

$$|P\rangle = H^k|i\rangle,$$

where i is uniformly distributed over the even-parity strings. Then the optimal probability of recovering the entire string k satisfies

$$P_{\text{guess}}(k \mid P) \leq 2^{1-n}.$$

Proof For fixed k, averaging over the even-parity string i gives

$$\rho_P^{(k)} = \frac{1}{2^n}(I + Q_k).$$

Since Q_k has eigenvalues ± 1, the largest eigenvalue of $\rho_P^{(k)}$ is

$$\|\rho_P^{(k)}\|_\infty = 2^{1-n}.$$

Let $\{M_{\widehat{k}}\}_{\widehat{k}\in\{0,1\}^n}$ be any POVM used to guess k. Then

$$\begin{aligned} P_{\text{guess}}(k \mid P) &= \frac{1}{2^n}\sum_k \operatorname{Tr}\left(M_k \rho_P^{(k)}\right) \\ &\le \frac{1}{2^n}\sum_k \|\rho_P^{(k)}\|_\infty \operatorname{Tr}(M_k) \\ &= \frac{2^{1-n}}{2^n}\operatorname{Tr}\left(\sum_k M_k\right) \\ &= \frac{2^{1-n}}{2^n}\operatorname{Tr}(I) \\ &= 2^{1-n}. \end{aligned}$$

Thus a single parity public-key register does not permit recovery of the entire basis string with non-negligible probability. □

Joint Exposure of the Public-Key and Ciphertext Parity Registers . We now consider a stronger experiment in which an adversary is additionally given one retained copy of the public-key parity register together with the corresponding parity ciphertext register.

Theorem 6.5 (Joint parity-register exposure)**.** *Let ρ_0^{PD} and ρ_1^{PD} denote the joint density operators corresponding to $\mu = 0$ and $\mu = 1$, respectively, when the adversary is given both $|P\rangle$ and $|D\rangle$. Then*

$$\left\|\rho_0^{PD} - \rho_1^{PD}\right\|_1 \le 2^{1-n/2} + 2^{1-n}.$$

Consequently,

$$\text{Adv}_{\text{joint}} \le 2^{-1-n/2} + 2^{-1-n},$$

which is negligible in n.

Proof For fixed k, the public parity register has density operator

$$\rho_P^{(k)} = \frac{1}{2^n}(I + Q_k),$$

whereas the parity ciphertext register conditioned on μ has density operator

$$\rho_{D,\mu}^{(k)} = \frac{1}{2^n}\left(I + (-1)^\mu Q_k\right).$$

Let $z = i \oplus j$. The mapping

$$(i, j) \longmapsto (i, z = i \oplus j)$$

is a bijection from pairs satisfying

$$\text{parity}(i) = 0, \qquad \text{parity}(j) = \mu$$

to pairs satisfying

$$\text{parity}(i) = 0, \qquad \text{parity}(z) = \mu.$$

Since i and j are chosen uniformly under these constraints, i and z are independent conditioned on k. Hence,

$$\rho_{PD,\mu}^{(k)} = \rho_P^{(k)} \otimes \rho_{D,\mu}^{(k)}.$$

Averaging over k gives

$$\rho_\mu^{PD} = \frac{1}{2^n} \sum_k \rho_P^{(k)} \otimes \rho_{D,\mu}^{(k)}.$$

Therefore,

$$\begin{aligned}\rho_0^{PD} - \rho_1^{PD} &= 2^{1-3n} \sum_k (I \otimes Q_k + Q_k \otimes Q_k) \\ &= 2^{1-3n} \Big[I \otimes (X+Z)^{\otimes n} + (X \otimes X + Z \otimes Z)^{\otimes n} \Big].\end{aligned}$$

By the triangle inequality,

$$\begin{aligned}\left\| \rho_0^{PD} - \rho_1^{PD} \right\|_1 \le 2^{1-3n} \Big(&\left\| I \otimes (X+Z)^{\otimes n} \right\|_1 \\ &+ \left\| (X \otimes X + Z \otimes Z)^{\otimes n} \right\|_1 \Big).\end{aligned}$$

Since

$$\|X + Z\|_1 = 2\sqrt{2},$$

we have

$$\left\| I \otimes (X+Z)^{\otimes n} \right\|_1 = 2^{5n/2}.$$

Furthermore, $X \otimes X + Z \otimes Z$ has Bell-basis eigenvalues $2, 0, 0, -2$, and therefore

$$\|X \otimes X + Z \otimes Z\|_1 = 4.$$

Hence,

$$\left\| (X \otimes X + Z \otimes Z)^{\otimes n} \right\|_1 = 2^{2n}.$$

Substitution yields

$$\left\| \rho_0^{PD} - \rho_1^{PD} \right\|_1 \le 2^{1-n/2} + 2^{1-n}.$$

The corresponding distinguishing-advantage bound follows from the Helstrom bound. □

Corollary 6.3 (Ciphertext security under retained parity-key exposure)**.** *Suppose the adversary receives the data ciphertext register together with one retained public parity register and the corresponding parity ciphertext register. For any two one-qubit plaintexts $|M_0\rangle$ and $|M_1\rangle$, the distinguishing advantage is at most*

$$2^{-1-n/2} + 2^{-1-n}.$$

Proof Let $\Omega_\mu = \rho_\mu^{PD}$. The joint state for plaintext M is

$$\Gamma_M = \frac{1}{2} \left(A_{M,0} \otimes \Omega_0 + A_{M,1} \otimes \Omega_1 \right).$$

Exactly as in the proof of Theorem 6.3,

$$\|\Gamma_{M_0} - \Gamma_{M_1}\|_1 \le \|\Omega_0 - \Omega_1\|_1.$$

Applying Theorem 6.5 and dividing by four gives the claimed distinguishing advantage. □

Multi-Qubit Product Plaintexts . The protocol description encrypts plaintexts of the product form $|M\rangle = |M_1\rangle \otimes \cdots \otimes |M_\ell\rangle$. For such plaintexts, the one-block security bound extends by a standard hybrid argument.

Theorem 6.6 (Multi-qubit product-message security)**.** *Let*

$$|M^{(0)}\rangle = \bigotimes_{r=1}^{\ell} |M_r^{(0)}\rangle, \qquad |M^{(1)}\rangle = \bigotimes_{r=1}^{\ell} |M_r^{(1)}\rangle$$

be two product-form ℓ-qubit plaintexts encrypted using independent one-time public-key blocks. Then

$$\mathrm{Adv}_\ell \le \ell\, 2^{-1-n/2}.$$

If the stronger retained-parity-key exposure of Theorem 6.5 is allowed independently for each block, then

$$\mathrm{Adv}_\ell^{\mathrm{joint}} \le \ell \left(2^{-1-n/2} + 2^{-1-n}\right).$$

For polynomially bounded ℓ, both quantities remain negligible in n.

Proof Introduce hybrid ciphertext states in which the first r blocks encrypt the corresponding components of $M^{(1)}$ and the remaining blocks encrypt those of $M^{(0)}$. Consecutive hybrids differ in exactly one ciphertext block. Theorem 6.3 bounds the distinguishing advantage between consecutive hybrids by $2^{-1-n/2}$. Summing over the ℓ hybrids gives

$$\mathrm{Adv}_\ell \le \ell\, 2^{-1-n/2}.$$

The same argument using Corollary 6.3 gives the second bound. □

7 Implementation and Experimental Validation

In this section, we validate both the ciphertext-leakage attack of Section 4 and the proposed sQPKE construction of Section 5 using Qiskit. All circuits were built with Qiskit 2.5 and executed on the Qiskit Aer simulator (`AerSimulator`). Every measurement-conditioned operation described in the protocol – the receiver's correction of the data qubit in Liu et al.'s scheme, the single-bit parity-encoding step in sQPKE, and the receiver's final correction of the sQPKE data qubit – is implemented as a genuine dynamic circuit, using mid-circuit measurement together with classically controlled gates (`if_test`) applied directly to the classical bit produced by the relevant measurement, rather than being applied afterward by classical post-processing. The secret function F used in sQPKE key generation is instantiated concretely as a keyed pseudorandom function, $F(s) = \mathrm{trunc}_n(\text{SHA-256}(\mathsf{sk} \,\|\, s))$ for a fixed secret key sk, rather than as an abstract random function on $\{0,1\}^n$; every experiment reported below that depends on F uses this instantiation, with a freshly sampled public string s per trial.

7.1 Validation of the Ciphertext-Leakage Attack and of Legitimate Decryption

For every private-key symbol $K_{p,i} \in \{k_0, k_1, k_2, k_3\}$ and every computational-basis plaintext $|0\rangle$ and $|1\rangle$, we built two circuits from the same encryption stage: (i) the adversary's ciphertext-leakage attack circuit, which measures the ciphertext qubit directly and never applies a decryption gate or uses the private key, exactly as in Algorithm 1; and (ii) the legitimate receiver's own decryption circuit, which measures the message qubit to obtain m and then applies the correction operator $R_{K_{p,i},m}$ to the ciphertext qubit conditioned on the measured bit, using a classically controlled gate. Figure 1 shows circuit (i), illustrated for private-key symbol k_2 and plaintext $|1\rangle$; Figure 2 shows circuit (ii) for the same configuration.

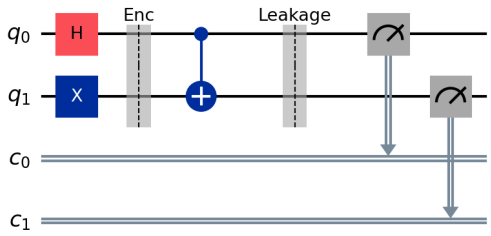


Fig. 1 Qiskit circuit for Liu et al.'s encryption together with the ciphertext-leakage attack ($K_{p,i} = k_2$, plaintext $|1\rangle$). The register c_1 holds the exposed classical bit m produced during encryption; the register c_0 holds the adversary's direct measurement of the ciphertext qubit. No decryption gate and no private-key information is used.

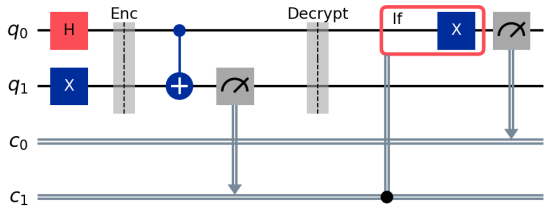


Fig. 2 Qiskit circuit for the legitimate receiver's decryption of the same ciphertext as Figure 1, as a genuine dynamic circuit: the message qubit is measured to obtain m (register c_1), and the correction operator $R_{K_{p,i},m}$ is applied to the data qubit conditioned on the measured bit before it is measured (register c_0).

For every one of the 8 configurations, we ran 4000 shots[2] of each circuit. Table 2 reports the resulting legitimate-decryption accuracy and attack success rate side by side.

Table 2 Qiskit validation of Liu et al.'s legitimate decryption and of the ciphertext-leakage attack (4096 shots per configuration)

Private key	Plaintext	Legit. decryption accuracy	Attack success rate ($x = t \oplus m$)
k_0	$\|0\rangle$	1.0000	1.0000
k_0	$\|1\rangle$	1.0000	1.0000
k_1	$\|0\rangle$	1.0000	1.0000
k_1	$\|1\rangle$	1.0000	1.0000
k_2	$\|0\rangle$	1.0000	1.0000
k_2	$\|1\rangle$	1.0000	1.0000
k_3	$\|0\rangle$	1.0000	1.0000
k_3	$\|1\rangle$	1.0000	1.0000

In every one of the $8 \times 4096 = 32768$ shots of the legitimate-decryption circuit, the receiver recovered the correct plaintext bit using only $K_{p,i}$ and the measured bit m. In every one of the 32768 shots of the attack circuit, the adversary's rule $x = t \oplus m$ recovered the same plaintext bit with 100% accuracy, independently of the private-key symbol, and without ever applying a decryption gate or using $K_{p,i}$. The full breakdown of observed (t, m) outcome patterns and their frequencies for every configuration is reported in the accompanying results table; for plaintext $|1\rangle$ under $K_{p,i} = k_2$, only the patterns with $t \oplus m = 1$ occur, exactly as predicted in Section 4.

We also validated two claims beyond the basis-state case, using exact statevector simulation (no sampling noise, via Qiskit's `Statevector` and `StatePreparation`) rather than shot-based sampling. First, we verified that the receiver's decryption is correct for *arbitrary* plaintext qubits, not only computational-basis states: for 400 Haar-random single-qubit plaintexts per private-key symbol (generated via `qiskit.quantum_info.random_statevector`), we deterministically extracted both measurement branches $m = 0$ and $m = 1$ of the post-CNOT statevector, applied the corresponding correction operator to each branch, and computed the fidelity with the original plaintext using `qiskit.quantum_info.state_fidelity`. This yields 800 fidelity values per private-key symbol (400 trials $\times$ 2 branches); the minimum and mean fidelity over all 3200 values, across all four symbols, were both 1.0000000000 up to floating-point round-off (minimum observed value 0.9999999999999993). Second, we confirmed that the attack is specific to computational-basis plaintexts, as claimed in Section 4: for the genuine superposition plaintext $|+\rangle$ under $K_{p,i} = k_2$, the adversary's recovered bit $x = t \oplus m$ satisfied $\Pr[x = 0] = 0.4873$ and $\Pr[x = 1] = 0.5127$ over 4096 shots, statistically indistinguishable from a fair coin, and thus carrying no information about the plaintext.

[2] 4096 shots per configuration in the implementation, the nearest power of two to the 4000 shots quoted in the text; all reported rates are computed over the full shot count returned by the simulator.

7.2 Validation of sQPKE Correctness

We validated sQPKE correctness in two complementary ways: an isolated sweep over the parity-register mechanism alone (to characterize correctness as a function of n over many independent random trials), and a full end-to-end dynamic circuit that combines the data-qubit component and the parity-register component in a single circuit with no classical post-processing between measurements.

Parity-register correctness sweep.

For each parity-register length $n \in \{2, 4, 8, 16\}$, we ran 300 independent trials. In each trial, a fresh public string s is drawn and the correction-syndrome basis is computed as $k = F(s)$ using the SHA-256 PRF instantiation described above; an even-parity string i and a string j with parity equal to a randomly drawn target bit μ are drawn as in Key Generation and Encryption; and the resulting circuit $H^k \to \sigma_y^j \to H^k \to$ measure (Figure 3) is executed with 128 shots. In every one of the $4 \times 300 = 1200$ trials, the recovered syndrome satisfied $\widehat{\mu} = \mu$ and the corresponding computational-basis plaintext bit was recovered correctly, i.e. $\Pr[\widehat{\mu} = \mu] = \Pr[\widehat{x} = x] = 1.0000$ at every tested value of n (Table 3).

Table 3 Correctness of the sQPKE parity mechanism, F instantiated as a SHA-256 PRF (300 trials per value of n, 128 shots per trial)

n	$\widehat{\mu} = \mu$ **rate**	$\widehat{x} = x$ **rate**
2	1.0000	1.0000
4	1.0000	1.0000
8	1.0000	1.0000
16	1.0000	1.0000

Full end-to-end decryption.

The parity-register sweep above validates the syndrome-recovery mechanism in isolation, but does not by itself exercise the data-qubit correction step or rule out unintended interaction between the data and parity components when they share a single circuit. We therefore also built the complete one-message-qubit sQPKE circuit as a single dynamic circuit with no classical post-processing between any two measurements (Figure 4): the data qubit is encrypted via CNOT and the message qubit is measured to obtain μ; the parity register is prepared as $H^k |i\rangle$; Bob's correction string j is applied via σ_y^j, with the parity-determining final bit of j applied conditionally on the measured value of μ (a single-bit `if_test`, since the remaining $n - 1$ bits of j are fixed independently of μ); H^k is reapplied to decode the parity register; the n parity qubits are then fanned into a single fresh ancilla via CNOTs and only the ancilla is measured, recovering $\widehat{\mu}$ as a single bit through a purely quantum computation rather than classical post-processing of n separate measurement outcomes (valid because, by

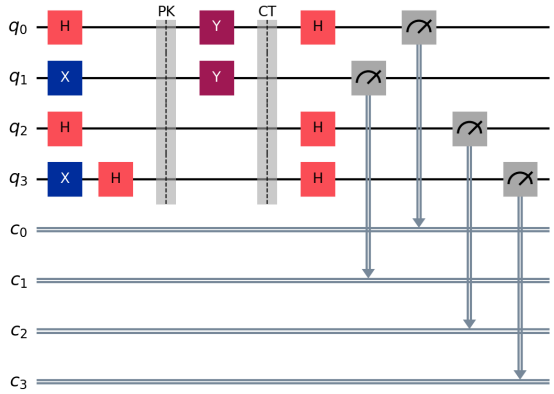


Fig. 3 Qiskit circuit for the isolated $n = 4$ parity-register mechanism: preparation of the public-key state $H^k |i\rangle$ (barrier "PK"), parity encoding via σ_y^j with parity$(j) = 0$ (barrier "CT"), and decoding via H^k followed by measurement of all four parity qubits.

the correctness proof of Section 5.1, the decoded parity register is guaranteed to be in a computational basis state, not a superposition, so the CNOT fan-in reproduces the classical parity exactly); and finally the correction operator $R_{d,\widehat{\mu}}$ is applied to the data qubit, conditioned on the measured ancilla bit, before the data qubit itself is measured.

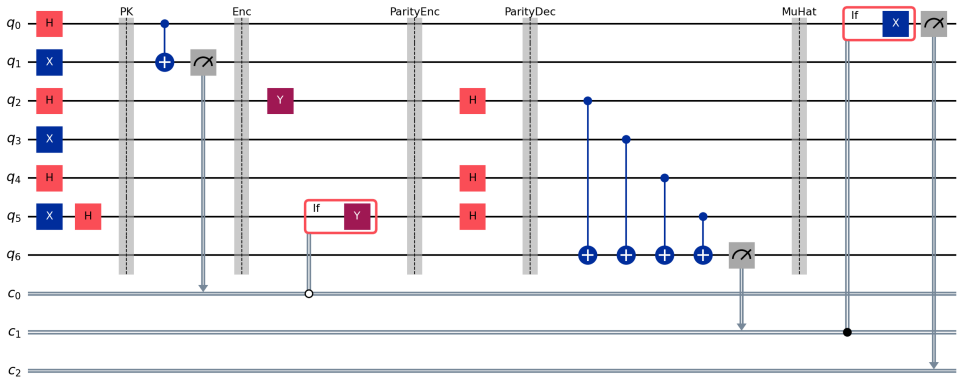


Fig. 4 Qiskit circuit for the complete one-message-qubit sQPKE decryption, $n = 4$ ($K_{p,i} = k_2$, plaintext $|1\rangle$): data-qubit encryption ("Enc"), parity encoding of μ conditioned on the measured message-qubit outcome ("ParityEnc"), decoding via H^k ("ParityDec"), quantum parity computation via a CNOT fan-in into a fresh ancilla and measurement of that ancilla ("MuHat"), and the conditional correction operator applied to the data qubit before its final measurement.

Because the four-state public-key qubit is itself in superposition for every $d \in \{0, 1, 2, 3\}$, running many shots of a single fixed circuit configuration naturally samples both the $\mu = 0$ and $\mu = 1$ branches, so a single circuit run already exercises both

correction branches statistically. We ran this circuit for every private-key symbol $d \in \{0, 1, 2, 3\}$ and every computational-basis plaintext, at parity-register lengths $n \in \{2, 4, 8\}$, with 2000 shots at $n \in \{2, 4\}$ and 1000 shots at $n = 8$.[3] In every one of the resulting $3 \times 4 \times 2 = 24$ configurations, the recovered syndrome agreed with the true measurement outcome ($\widehat{\mu} = \mu$) in 100% of shots, and the decrypted data qubit matched the original plaintext bit in 100% of shots (Table 4), confirming that the data and parity components decrypt correctly when combined in a single circuit rather than validated separately.

Table 4 Full end-to-end sQPKE decryption: agreement rate between recovered and true correction bit, and plaintext recovery rate, aggregated over all private-key symbols $d \in \{0, 1, 2, 3\}$ and both computational-basis plaintexts at each n

n	**Shots**	$\widehat{\mu} = \mu$ **rate**	**Plaintext recovery rate**
2	2000	1.0000	1.0000
4	2000	1.0000	1.0000
8	1000	1.0000	1.0000

Finally, Table 5 reports gate counts and circuit depth obtained directly from Qiskit (`QuantumCircuit.count_ops()` and `QuantumCircuit.depth()`) for representative circuits from both schemes, including – unlike an earlier version of this validation – the full end-to-end sQPKE decryption circuit, so that the correction operator applied to the data qubit is reflected in the reported gate counts rather than omitted.

The full end-to-end decryption circuit (last row) uses $n+3$ qubits for parity-register length n (data, message, n parity qubits, and one fan-in ancilla) and 3 classical bits (μ, $\widehat{\mu}$, and the decrypted data bit), consistent with the resource estimates of Section 8. Its two `if_else` blocks implement the single-bit conditional parity-encoding step and the two-branch ($\widehat{\mu} = 0$ and $\widehat{\mu} = 1$) conditional data-qubit correction, respectively; for $K_{p,i} = k_2$ the $\widehat{\mu} = 0$ correction branch is the identity and contributes no gates, which is why only one conditional X gate is visible in Figure 4.

8 Efficiency Analysis and Resource Estimation

In this section, we analyze the computational and quantum resources required by the sQPKE. The analysis is given for an ℓ-qubit plaintext, where each message qubit is encrypted independently. For each message qubit, the scheme uses one data qubit and one parity register of length n. The parameter n is the security parameter associated with the parity-encoded correction register.

[3] Larger n (e.g. $n = 16$) were not included in this particular end-to-end dynamic-circuit sweep: `AerSimulator`'s handling of classically controlled operations was observed not to scale as smoothly with qubit count as static circuits, with per-circuit runtime increasing sharply beyond $n = 8$ in our environment. Correctness at larger n is already established both by the parity-only sweep of Table 3, which is validated up to $n = 16$ without requiring dynamic-circuit execution, and by the exact-statevector argument of Section 5.1, which holds for all n.

Table 5 Qiskit logical circuit metrics before hardware transpilation ($n = 4$ where applicable).

Circuit	Qubits	Cbits	Depth	Size	Operation counts
Original attack	2	2	3	5	H: 1, X: 1, CX: 1, meas.: 2
Liu et al. decryption	2	2	5	7	H: 1, X: 1, CX: 1, meas.: 2, conditional[a]: 2
sQPKE parity circuit	4	4	5	13	H: 6, X: 2, Y: 1, meas.: 4
sQPKE parity key generation	4	0	2	5	H: 3, X: 2
sQPKE parity decoding	4	4	2	7	H: 3, meas.: 4
sQPKE end-to-end decryption	7	3	13	22	H: 7, X: 3, Y: 1, CX: 5, meas.: 3, conditional[a]: 3

[a]A *conditional* block is a classically controlled operation (implemented in Qiskit as an `if_test/if_else` block): a gate that is applied only when a classical bit produced by an earlier mid-circuit measurement takes a specified value. CX denotes the CNOT gate and meas. denotes a computational-basis measurement.

Parameter Description. Let ℓ denote the number of message qubits and let n denote the length of the parity register used for encoding the correction bit associated with one message qubit. For each $1 \le r \le \ell$, the public key contains a data public-key qubit and an n-qubit parity public-key register. Thus, each message qubit is associated with a public-key block of size $n + 1$ qubits. The notation used in the resource estimation is summarized below.

- ℓ: number of plaintext qubits.
- n: length of the parity register for one plaintext qubit.
- $w(k^{(r)})$: Hamming weight of the string $k^{(r)} = F(s_r)$.
- $w(j^{(r)})$: Hamming weight of the parity string $j^{(r)}$ chosen during encryption.
- $T_F(n)$: classical cost of evaluating the function F on an n-bit input.
- $g(d_r, \widehat{\mu}_r)$: number of single-qubit gates required to implement the correction operator $R_{d_r,\widehat{\mu}_r}$ applied to the data ciphertext qubit $|C_r\rangle$ during decryption (Table 1).

For a uniformly random n-bit string, the expected Hamming weight is $n/2$. Therefore, whenever $k^{(r)}$ or $j^{(r)}$ is uniformly distributed, the expected number of position-wise Hadamard or Pauli-Y operations is $n/2$, while the worst-case number is n. We also record the cost of the correction operator explicitly, since it is the final step of decryption and is easy to overlook in an asymptotic count. From Table 1, the operator $R_{d_r,\widehat{\mu}_r}$ takes one of the eight values $I, X, Z, P, P^\dagger, XP, XP^\dagger, XZ$, depending on the private index $d_r \in \{0, 1, 2, 3\}$ and the recovered correction bit $\widehat{\mu}_r \in \{0, 1\}$. Counting single-qubit

gates,

$$g(d_r, \widehat{\mu}_r) = \begin{cases} 0, & (d_r, \widehat{\mu}_r) = (2,0) \quad (R = I), \\ 1, & (d_r, \widehat{\mu}_r) \in \{(0,0),(1,0),(2,1),(3,0)\} \quad (R \in \{P, P^\dagger, X, Z\}), \\ 2, & (d_r, \widehat{\mu}_r) \in \{(0,1),(1,1),(3,1)\} \quad (R \in \{XP^\dagger, XP, XZ\}). \end{cases}$$

Hence $g(d_r, \widehat{\mu}_r) \in \{0,1,2\}$ for every r; in particular $g(d_r, \widehat{\mu}_r) \leq 2$ always. If d_r and $\widehat{\mu}_r$ are each uniformly distributed and independent, the expected value is

$$\mathbb{E}\big[g(d_r, \widehat{\mu}_r)\big] = \frac{0+1+1+2+1+2+1+2}{8} = \frac{10}{8} = 1.25.$$

This quantity is used below to give an exact, rather than purely asymptotic, account of the decryption cost.

Key Generation Cost. For each plaintext position r, Alice prepares two quantum components: a single-qubit data public key and an n-qubit parity public-key register. First, Alice chooses a private index $d_r \in \{0,1,2,3\}$, which determines one of the four data public-key states. The preparation of this single-qubit state requires only a constant number of elementary single-qubit gates. Hence, the cost of preparing the data public-key qubit is $O(1)$. Next, Alice chooses a public string $s_r \in \{0,1\}^n$ and computes $k^{(r)} = F(s_r)$. She also chooses an even-parity string $i^{(r)} \in \{0,1\}^n$ and prepares the computational-basis state $|i^{(r)}\rangle$. The preparation of $|i^{(r)}\rangle$ requires at most n Pauli-X operations if the register is initialized to $|0^n\rangle$. Alice then applies $H^{k^{(r)}} = H^{k_1^{(r)}} \otimes \cdots \otimes H^{k_n^{(r)}}$. This requires exactly $w(k^{(r)})$ Hadamard gates. Therefore, for one plaintext position, the key-generation cost is $O(1) + T_F(n) + O(n)$. For ℓ plaintext qubits, the total key-generation cost is $O(\ell n) + \ell T_F(n)$. The quantum public key contains $(n+1)$ qubits per plaintext position. Hence, the total public-key size for encrypting ℓ plaintext qubits is $\ell(n+1)$ qubits, together with the classical strings $s_1, \ldots, s_\ell$, which require ℓn classical bits.

Encryption Cost. For each plaintext qubit $|M_r\rangle$, Bob performs two operations. First, he encrypts the data qubit using the corresponding data public-key qubit. Second, he encodes the correction bit into the parity register. For the data-qubit encryption, Bob applies one CNOT gate, where the public-key qubit is the control qubit and the plaintext qubit is the target qubit. He then measures the target qubit in the computational basis and obtains a correction bit $\mu_r \in \{0,1\}$. Thus, the data-qubit part of the encryption requires 1 CNOT gate + 1 measurement To encode the correction bit, Bob chooses a random string $j^{(r)} = (j_1^{(r)}, \ldots, j_n^{(r)})$ such that $j_1^{(r)} \oplus \cdots \oplus j_n^{(r)} = \mu_r$. He then applies $\sigma_y^{j^{(r)}} = \sigma_y^{j_1^{(r)}} \otimes \cdots \otimes \sigma_y^{j_n^{(r)}}$ to the parity register. This requires exactly $w(j^{(r)})$ Pauli-Y gates. In the worst case, this is n gates, while in expectation it is $n/2$ gates. Therefore, the encryption cost for one plaintext qubit is

$$1 \text{ CNOT} + 1 \text{ measurement} + w(j^{(r)}) \text{ Pauli-}Y \text{ gates}.$$

In the worst case, this is $O(n)$ quantum operations per plaintext qubit. For an ℓ-qubit plaintext, the total encryption cost is

$$\ell \text{ CNOT gates} + \ell \text{ measurements} + \sum_{r=1}^{\ell} w(j^{(r)}) \text{ Pauli-}Y \text{ gates.}$$

Hence, the worst-case encryption complexity is $O(\ell n)$ and the expected number of Pauli-Y gates is $\frac{\ell n}{2}$.

Decryption Cost. For each ciphertext component, Alice performs two stages: she first recovers the correction bit from the parity register, and then applies the corresponding correction operator to the data ciphertext qubit. We account for both stages explicitly. *Stage 1: recovery of the correction bit.* Given the public string s_r, Alice computes $k^{(r)} = F(s_r)$. This requires classical cost $T_F(n)$. She then applies $H^{k^{(r)}}$ to the parity register. This requires exactly $w(k^{(r)})$ Hadamard gates. After that, Alice measures all n qubits of the parity register in the computational basis and obtains a classical string $z^{(r)} = (z_1^{(r)}, \ldots, z_n^{(r)})$. She computes its parity $z_1^{(r)} \oplus z_2^{(r)} \oplus \cdots \oplus z_n^{(r)}$, which gives the recovered correction bit $\widehat{\mu}_r$. The parity computation requires $n-1$ XOR operations. *Stage 2: correction of the data qubit.* Alice selects the correction operator $R_{d_r,\widehat{\mu}_r}$ from Table 1 using her private index d_r and the recovered bit $\widehat{\mu}_r$, and applies it to $|C_r\rangle$. As shown above, this requires $g(d_r, \widehat{\mu}_r) \in \{0, 1, 2\}$ single-qubit gates, drawn from $\{I, X, Z, P, P^\dagger\}$ (composite operators XP, $XP^\dagger$, XZ being applied as two consecutive single-qubit gates). This stage therefore costs $O(1)$ gates in the worst case (2 gates), and 1.25 gates on average over uniformly random $(d_r, \widehat{\mu}_r)$, per message qubit. Although this term does not change the asymptotic complexity of decryption, we retain it explicitly rather than absorbing it silently into a generic $O(1)$ term, since it is the step that ultimately outputs the recovered plaintext qubit. Combining both stages, the decryption cost for one plaintext qubit is

$$\begin{aligned} &T_F(n) + w(k^{(r)}) \text{ Hadamard gates} + n \text{ measurements} \\ &\quad + (n-1) \text{ XOR operations} + g(d_r, \widehat{\mu}_r) \text{ correction gates.} \end{aligned}$$

For ℓ plaintext qubits, the total decryption cost is

$$\begin{aligned} \text{DecCost} = \ell T_F(n) + \sum_{r=1}^{\ell} w(k^{(r)}) \text{ Hadamard gates} + \ell n \text{ measurements} \\ + \ell(n-1) \text{ XOR operations} + \sum_{r=1}^{\ell} g(d_r, \widehat{\mu}_r) \text{ correction gates.} \end{aligned}$$

Since $g(d_r, \widehat{\mu}_r) \leq 2$ for every r, the correction-gate term contributes at most 2ℓ single-qubit gates in the worst case, and $1.25\,\ell$ gates in expectation; either way it is $O(\ell)$ and does not affect the dominant $O(\ell n)$ scaling. Hence, the worst-case decryption complexity remains $O(\ell n) + \ell T_F(n)$.

Ciphertext Size. For each plaintext qubit, the ciphertext consists of three parts: $C_r = (|C_r\rangle, s_r, |D_r\rangle)$. Here, $|C_r\rangle$ is one encrypted data qubit, s_r is an n-bit public classical string, and $|D_r\rangle$ is an n-qubit parity register. Therefore, the ciphertext size per plaintext qubit is $n+1$ quantum qubits and n classical bits. For an ℓ-qubit plaintext, the total ciphertext size is $\ell(n+1)$ qubits and ℓn classical bits. Thus, the ciphertext expansion is linear in both the message length ℓ and the security parameter n. The resource requirements of the sQPKE are summarized in Table 6. The estimates are given for an ℓ-qubit plaintext and parity-register length n.

Table 6 Resource estimation of the sQPKE

Operation	Resource requirement
Public-key size	$\ell(n+1)$ qubits and ℓn classical bits
Secret-key size	ℓ private indices and the secret function F
Ciphertext size	$\ell(n+1)$ qubits and ℓn classical bits
Key generation	$O(\ell n) + \ell T_F(n)$
Encryption	$O(\ell n)$ quantum operations
Decryption	$O(\ell n) + \ell T_F(n)$
CNOT gates in encryption	ℓ
Measurements in encryption	ℓ
Pauli-Y gates in encryption	Worst case ℓn, expected $\ell n/2$
Hadamard gates in decryption	Worst case ℓn, expected $\ell n/2$
Measurements in decryption	ℓn
Classical XOR operations	$\ell(n-1)$
Correction gates in decryption	Worst case 2ℓ, expected $1.25\,\ell$

The table shows that the sQPKE has linear complexity in the plaintext length and in the parity-register length. No part of the construction requires large matrix operations, modular exponentiation, lattice reduction, decoding, or other computationally expensive classical subroutines. The quantum part uses only elementary gates and measurements. We emphasize that the correction-gate row, although $O(\ell)$ and therefore asymptotically dominated by the $O(\ell n)$ terms, is listed explicitly because it is the operation that directly produces the decrypted plaintext qubit $|\widehat{M}_r\rangle$ from $|C_r\rangle$; any implementation or empirical gate-count validation of the scheme (e.g., Table 5 in Section 7) should likewise report it alongside the CNOT, Hadamard, and Pauli-Y counts.

Comparison with the Four-State Base Construction Table 7 summarizes the security differences between Liu et al.'s scheme and the sQPKE. The table also highlights that the sQPKE improves resistance against passive eavesdropping and key-recovery attempts under the one-time public-key setting.

Circuit Depth. The circuit depth of the sQPKE is also low. In encryption, the CNOT operation on the data qubit is followed by a measurement of the target qubit. The Pauli-Y gates applied to the parity register act on distinct qubits and can be performed in parallel. Therefore, the depth of the parity encoding step is one, assuming parallel

Table 7 Security comparison

Security Property	**Liu et al. [18]**	sQPKE
Ciphertext-leakage resistance	×	✓
Plaintext confidentiality	×	✓
Key-recovery resistance	Partial	Improved under one-time use
Ciphertext indistinguishability	×	✓
Eavesdropping resistance	×	✓

gate execution. Similarly, during decryption, the Hadamard gates applied according to $k^{(r)}$ act independently on different qubits of the parity register and can also be performed in parallel. The measurement of the parity register can be performed in one measurement layer. The classical parity computation requires $n-1$ XOR operations, which can be done in logarithmic depth using a tree-based XOR circuit, or in linear time using a sequential computation. After the parity computation, the correction operator $R_{d_r,\widehat{\mu}_r}$ is applied to the data qubit; since this operator consists of at most two single-qubit gates applied sequentially to a single wire (Table 1), it contributes at most a constant additive depth of 2, independent of n. Thus, under parallel execution, the quantum depth per message qubit is constant, apart from the classical parity computation. The total sequential cost remains linear when the message qubits are processed one by one, but the construction also permits parallel encryption and decryption across different message positions if enough public-key registers are available.

The efficiency analysis shows that sQPKE is lightweight in terms of gate complexity and algorithmic structure. The scheme requires more qubits than the basic four-state encryption mechanism because each encrypted data qubit is accompanied by an n-qubit parity register. However, this overhead is linear and uses only simple quantum operations. The main advantages of the sQPKE are as follows. First, the encryption and decryption algorithms are easy to implement using standard quantum gates. Second, the classical processing is limited to random string generation, XOR operations, parity computation, and evaluation of F. Third, the circuit depth is low because most operations on the parity register are parallelizable, and the final data-qubit correction adds only a small constant depth on top of this.

9 Conclusion

In this work, the four-state QPKE scheme of Liu et al.[18] was analyzed. To address the security limitations of [18], the sQPKE scheme was proposed using only elementary quantum and classical operations. Its correctness and security against ciphertext leakage, eavesdropping, and distinguishing attacks were established and further supported through Qiskit-based experiments and resource estimation. In future work, the proposed framework will be extended toward advanced encryption functionalities, including

functional encryption, fully homomorphic encryption, searchable symmetric encryption, and attribute-based encryption.

Acknowledgments

Vikas Srivastava acknowledges the support received from the ANRF-PMECRG project with Ref. `ANRF/ECRG/2025/002808/PMS` and NIT Warangal Research Seed Grant. Rahul Kumar acknowledges the support received from University Grants Commission.

References

[1] Jajodia, S., Samarati, P., Yung, M.: Encyclopedia of Cryptography, Security and Privacy. Springer, (2025). https://doi.org/10.1007/978-3-030-71522-9

[2] Li, H., Huang, Q., Huang, J., Susilo, W.: Public-key authenticated encryption with keyword search supporting constant trapdoor generation and fast search. IEEE Transactions on Information Forensics and Security **18**, 396–410 (2022) https://doi.org/10.1109/TIFS.2022.3224308

[3] Elhabob, R., Eltayieb, N., Xiong, H., Khan, F., Bashir, A.K., Kumari, S., Alturki, R., Kumar, S.: Equality test public key encryption with cryptographic reverse firewalls for cloud-based e-commerce. IEEE Transactions on Consumer Electronics **70**(4), 6763–6775 (2024) https://doi.org/10.1109/TCE.2024.3405496

[4] Saranya, A., Naresh, R.: Efficient mobile security for e health care application in cloud for secure payment using key distribution. Neural Processing Letters **55**(1), 141–152 (2023) https://doi.org/10.1007/s11063-021-10482-1

[5] Srivastava, V., Roy, D., Mesnager, S., Kundu, N., Debnath, S.K., Mukhopadhyay, S.: A post-quantum secure end-to-end verifiable e-voting protocol based on multivariate polynomials. arXiv preprint arXiv:2512.17613 (2025) https://doi.org/10.48550/arXiv.2512.17613

[6] Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700 . Ieee

[7] Portmann, C., Renner, R.: Security in quantum cryptography. Reviews of Modern Physics **94**(2), 025008 (2022) https://doi.org/10.1103/RevModPhys.94.025008

[8] Okamoto, T., Tanaka, K., Uchiyama, S.: Quantum public-key cryptosystems. In: Annual International Cryptology Conference, pp. 147–165 (2000). https://doi.org/10.1007/3-540-44598-6_9 . Springer

[9] Gottesman, D.: Quantum Public-Key Cryptography with Information-Theoretic Security. Invited talk, Workshop on Classical and Quantum Information Security,

California Institute of Technology (Caltech), Pasadena, CA, USA. 15–18 December 2005 (2005)

[10] Yang, L., Yang, B., Pan, J.: Quantum public-key encryption protocols with information-theoretic security. In: Durt, T., Zadkov, V.N. (eds.) Quantum Optics II, vol. 8440, p. 84400. SPIE, (2012). https://doi.org/10.1117/12.922444 . International Society for Optics and Photonics

[11] Nikolopoulos, G.M.: Applications of single-qubit rotations in quantum public-key cryptography. Phys. Rev. A **77**, 032348 (2008) https://doi.org/10.1103/PhysRevA.77.032348

[12] Nikolopoulos, G.M., Ioannou, L.M.: Deterministic quantum-public-key encryption: forward search attack and randomization. Physical Review A—Atomic, Molecular, and Optical Physics **79**(4), 042327 (2009) https://doi.org/10.1103/PhysRevA.79.042327

[13] Liang, M., Yang, L.: Public-key encryption and authentication of quantum information. Science China Physics, Mechanics and Astronomy **55**(9), 1618–1629 (2012) https://doi.org/10.1007/s11433-011-4806-y

[14] Wu, C., Yang, L.: Bit-oriented quantum public-key encryption based on quantum perfect encryption. Quantum Information Processing **15**(8), 3285–3300 (2016) https://doi.org/10.1007/s11128-016-1339-3

[15] Wu, W., Cai, Q., Zhang, H., Liang, X.: Quantum public key cryptosystem based on bell states. International Journal of Theoretical Physics **56**(11), 3431–3440 (2017) https://doi.org/10.1007/s10773-017-3506-4

[16] Yang, L., Yang, B., Xiang, C.: Quantum public-key encryption schemes based on conjugate coding: L. yang et al. Quantum Information Processing **19**(11), 415 (2020) https://doi.org/10.1007/s11128-020-02912-1

[17] Wang, Y., She, K.: A practical quantum public-key encryption model. In: 2017 3rd International Conference on Information Management (ICIM), pp. 367–372 (2017). https://doi.org/10.1109/INFOMAN.2017.7950410 . IEEE

[18] Liu, Z., Xie, Q., Zha, Y., Dong, Y.: Quantum public key encryption scheme with four states key. Physica Scripta **97**(4), 045102 (2022) https://doi.org/10.1088/1402-4896/ac576c

[19] Kitagawa, F., Morimae, T., Nishimaki, R., Yamakawa, T.: Quantum public-key encryption with tamper-resilient public keys from one-way functions. In: Annual International Cryptology Conference, pp. 93–125 (2024). https://doi.org/10.1007/978-3-031-68394-7_4 . Springer

[20] Barooti, K., Grilo, A.B., Huguenin-Dumittan, L., Malavolta, G., Sattath, O.,

Vu, Q.-H., Walter, M.: Public-key encryption with quantum keys. In: Theory of Cryptography Conference, pp. 198–227 (2023). https://doi.org/10.1007/978-3-031-48624-1_8 . Springer

[21] Malavolta, G., Walter, M.: Robust quantum public-key encryption with applications to quantum key distribution. In: Annual International Cryptology Conference, pp. 126–151 (2024). https://doi.org/10.1007/978-3-031-68394-7_5 . Springer

[22] Grilo, A.B., Sattath, O., Vu, Q.-H.: Encryption with quantum public keys. arXiv preprint arXiv:2303.05368 (2023) https://doi.org/10.48550/arXiv.2303.05368

[23] Wu, M., Shi, R.-h., Gao, W., Li, K.: A secure quantum sealed-bid auction protocol based on quantum public key encryption. Quantum Information Processing **21**(2), 77 (2022) https://doi.org/10.1007/s11128-021-03404-6

[24] Coladangelo, A., Goldwasser, S., Vazirani, U.: Deniable encryption in a quantum world. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, pp. 1378–1391 (2022). https://doi.org/10.1145/3519935.3520019

[25] Rao, Y.S., Srivastava, V., Mohanty, T., Debnath, S.K.: Designing quantum-secure attribute-based encryption. Cluster Computing **27**(9), 13075–13091 (2024) https://doi.org/10.1007/s10586-024-04546-9

[26] Goldreich, O.: Foundations of Cryptography, Volume 2: Basic Applications. Cambridge University Press, Cambridge, UK (2004). https://doi.org/10.1017/CBO9780511721656

[27] Koshiba, T.: Security notions for quantum public-key cryptography. arXiv preprint quant-ph/0702183 (2007) https://doi.org/10.48550/arXiv.quant-ph/0702183