entropy

IMPACT FACTOR 2.1

Indexed in: PubMed

CITESCORE 4.9

Article

Flexible Threshold Quantum Homomorphic Encryption on Quantum Networks

Yongli Tang, Menghao Guo, Binyong Li, Kaixin Geng, Jinxia Yu and Baodong Qin

Special Issue

Nonlocality and Entanglement in Quantum Networks

Edited by

Prof. Dr. Mingxing Luo and Dr. Xue Yang

MDPI

https://doi.org/10.3390/e27010007

entropy

MDPI

Article

Flexible Threshold Quantum Homomorphic Encryption on Quantum Networks

Yongli Tang [1], Menghao Guo [3], Binyong Li [2,*], Kaixin Geng [3], Jinxia Yu [3] and Baodong Qin [4]

[1] School of Software, Henan Polytechnic University, Jiaozuo 454000, China
[2] Advanced Cryptography and System Security Key Laboratory of Sichuan, Chengdu 610225, China
[3] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China
[4] Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China
* Correspondence: lby@cuit.edu.cn

Abstract: Currently, most quantum homomorphic encryption (QHE) schemes only allow a single evaluator (server) to accomplish computation tasks on encrypted data shared by the data owner (user). In addition, the quantum computing capability of the evaluator and the scope of quantum computation it can perform are usually somewhat limited, which significantly reduces the flexibility of the scheme in quantum network environments. In this paper, we propose a novel $(t,\ n)$-threshold QHE (TQHE) network scheme based on the Shamir secret sharing protocol, which allows k $(t \leq k \leq n)$ evaluators to collaboratively perform evaluation computation operations on each qubit within the shared encrypted sequence. Moreover, each evaluator, while possessing the ability to perform all single-qubit unitary operations, is able to perform arbitrary single-qubit gate computation task assigned by the data owner. We give a specific (3, 5)-threshold example, illustrating the scheme's correctness and feasibility, and simulate it on IBM quantum computing cloud platform. Finally, it is shown that the scheme is secure by analyzing encryption/decryption private keys, ciphertext quantum state sequences during transmission, plaintext quantum state sequence, and the result after computations on the plaintext quantum state sequence.

Keywords: threshold quantum homomorphic encryption; Shamir secret sharing; quantum computation; quantum computing cloud platform

Academic Editors: Mingxing Luo and Xue Yang

Received: 3 December 2024
Revised: 21 December 2024
Accepted: 23 December 2024
Published: 26 December 2024

Citation: Tang, Y.; Guo, M.; Li, B.; Geng, K.; Yu, J.; Qin, B. Flexible Threshold Quantum Homomorphic Encryption on Quantum Networks. *Entropy* **2025**, *27*, 7. https://doi.org/10.3390/e27010007

1. Introduction

In the current era of information, safeguarding the security of private data is of paramount importance. Although traditional encryption techniques are excellent at securing data, the need to decrypt data arises when private data need to be computed and analyzed, which can potentially give rise to security risks. However, the privacy-preserving homomorphic encryption technique, as a revolutionary cryptographic tool, offers an innovative solution by allowing evaluation computations to be performed on encrypted data without the need for decryption. This means that the necessary computations and analyses can be performed while the private data remain encrypted, at the same time protecting the confidentiality and integrity of the private data. Since Rivest et al. [1] introduced the concept of classical fully homomorphic encryption (FHE) in 1978 and Gentry [2] proposed a classical FHE scheme in 2009, FHE has been widely used in various fields, including functional encryption [3], delegating computations [4], obfuscation [5] and plaintext encryption [6–9]. However, most classical FHE schemes cannot satisfy the security requirements of information theory.

With the rapid advancement in the field of cryptography, quantum cryptography has opened up a new avenue for the development of privacy-preserving homomorphic techniques. Quantum cryptography is based on the principles of quantum mechanics, and its security relies solely on the correctness of quantum mechanics, rather than computational assumptions. Therefore, it is capable of achieving information-theoretic security. In this context, quantum homomorphic encryption which integrates the concepts of homomorphic encryption and delegated quantum computation, as an important research branch in the field of quantum cryptography, provides a more secure data processing and computation mechanism, simultaneously offering a higher level of protection for sensitive information privacy.

In recent years, there has been a significant growth trend in research on quantum homomorphic encryption. This is attributed to the fact that quantum homomorphic encryption technology not only achieves the security requirements of information theory, but also provides an efficient approach to computing encrypted data. In 2012, Rohde et al. [10] proposed a restricted QHE scheme using the quantum wandering walk model. In 2013, Liang et al. [11] gave mathematical definitions of symmetric and asymmetric quantum homomorphic encryption schemes, and proposed a framework of QHE schemes with reference to classical homomorphic encryption schemes, which has been used until now. In their paper [11], four symmetric quantum homomorphic encryption schemes and one asymmetric quantum fully homomorphic encryption (QFHE) scheme were constructed based on the quantum one-time pad (QOTP). In 2015, Liang [12] constructed a QFHE scheme based on the quantum universal circuit. This scheme [12] uses the universal set of quantum gates $\{X, Y, Z, H, S, T, CNOT\}$ to achieve arbitrary quantum computation. However, the decryption of a T gates in this scheme requires interaction steps and is only applicable to delegated quantum computation between two parties. In the same year, Broadbent and Jeffery [13] proposed two QHE schemes with a constant finite number of non-Clifford gates (such as T gates) in the quantum circuit, where evaluator can perform at most a constant number of non-Clifford gates on the encrypted data. They also give a formal definition of QFHE. In 2016, Dulek et al. [14] extended the research from the scheme [13] and introduced a novel QHE scheme, which can effectively compute quantum circuits of arbitrary polynomial size and correct errors that arise during the evaluation computation stage when computing the T gates on the ciphertext quantum state sequence. In 2018, Ouyang and Tan [15] proposed a QHE scheme with a constant number of non-Clifford gates based on quantum codes. In addition, numerous other QHE schemes based on various approaches have been introduced [16–23].

However, it should be highlighted that currently, most QHE schemes are primarily designed for scenarios involving only a single evaluator [10–15,17–23]. When a data owner requires an evaluator to perform a large number of evaluation computation tasks on his or her encrypted private data, the burden on the single evaluator becomes quite heavy and may not be able to respond in a timely manner to perform computation tasks assigned by other users [24–27]. The data owner sometimes may be unwilling to place complete trust in a single evaluator, instead, anticipate multiple evaluators collaborating to accomplish some significant evaluation computation tasks [28–30]. In addition, the flexibility of most QHE schemes is significantly diminished due to the constraints on the quantum computing capacity of the evaluator and the scope of quantum computation it can perform [12,13,15,23,31,32].

In 2019, Chen et al. [31] proposed a (t, n)-threshold QHE scheme with a flexible number of evaluators based on Cao and Ma's [33] (t, n)-threshold quantum state sharing (QSTS) scheme. In this scheme [31], the data owner selects any d $(t \leq d \leq n)$ evaluators from the n evaluators to share encrypted data, and these d evaluators can collaboratively perform

all evaluation computation tasks (single-qubit gate unitary operations) on the encrypted data. Then the data owner can obtain the expected result after computations on her private plaintext data by decrypting the final ciphertext data. However, the scheme proposed by Chen et al. [31] imposes restrictions on the types of single-qubit gate unitary operations that evaluators can perform on encrypted data. The first $d-1$ evaluators are limited to performing only single-qubit gate unitary operations from the set of single-qubit gates $\{X, Y, Z, H\}$, while the last evaluator is allowed to perform single-qubit gate unitary operations from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$. That is, the first $d-1$ evaluators are not able to perform the single-qubit gates S and T, and not every evaluator can perform arbitrary single-qubit gate unitary operation in the set of single-qubit gates $\{X, Y, Z, H, S, T\}$ on the encrypted data. We know that in the scheme [31], the d evaluators selected randomly from the n evaluators are a stochastic process. Since any evaluator could potentially be selected to serve as the d-th evaluator, each evaluator has the ability to perform single-qubit unitary operations $\{X, Y, Z, H, S, T, U(\theta)\}$. However, it is important to note that the scope of quantum computation that evaluators can perform is subject to highly limitations.

In 2022, Liu et al. [32] proposed a (t, n)-threshold QHE scheme based on the Chen et al. [31] scheme. By exploring the relationship between the sets of single-qubit gates $\{X, Y, Z\}$ and $\{H, S, T\}$, they transferred the operations of $\{H, S, T\}$ which originally needed to be performed by the first $d-1$ evaluators to the last evaluator. It makes all k evaluators can perform any single-qubit gate unitary operation from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$, but the first $d-1$ evaluators only have ability to perform the single-qubit unitary operations from $\{X, Y, Z, U(\theta)\}$, while the last evaluator have ability to perform the single-qubit unitary operations from $\{X, Y, Z, S, T, U(\theta)\}$. It also has a certain impact on the flexibility and efficiency of the scheme. For this we propose a novel TQHE scheme based on the Shamir (t, n)-threshold secret sharing protocol [34]. Our proposed scheme not only supports a flexible number of evaluators but also ensures that all evaluators have the ability to perform all single-qubit unitary operations from $\{X, Y, Z, H, S, T, U(\theta)\}$ and are allowed to perform any computation task assigned by the data owner on the encrypted data. The proposed TQHE scheme exhibits excellent flexibility and can be easily implemented through simple operations. In the future, we anticipate that this scheme will play a crucial role in quantum network communication applications, providing more solutions to address security and privacy concerns in the computation of private data.

Our Contributions

First, we propose a novel (t, n)-threshold QHE scheme based on Shamir secret sharing, supporting any k $(t \leq k \leq n)$ evaluators of the n evaluators to cooperatively perform homomorphic evaluation computations on the ciphertext quantum state sequence shared by the data owner Alice. Alice is able to obtain the expected result after computations on the plaintext quantum state sequence by decrypting the final computed ciphertext quantum state sequence. Second, in our TQHE scheme, all evaluators, while having the capability to perform all single-qubit unitary operations $\{X, Y, Z, H, S, T, U(\theta)\}$, can perform any single-qubit gate unitary operation from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$ on the ciphertext quantum state sequence. Third, we provide a (3, 5)-threshold QHE example to clarify our scheme and validate its correctness and feasibility through simulations on the IBM quantum cloud platform. Finally, the security of the scheme is shown by analyzing the encryption/decryption private keys, ciphertext quantum state sequences during transmission, the plaintext quantum state sequence, and the result after computations on the plaintext quantum state sequence.

The remaining sections of this paper are organized as follows: Section 2 covers some preliminaries. Section 3 provides a detailed introduction to our novel TQHE scheme.

Section 4 provides a concrete example and simulates it on the IBM quantum computing cloud platform. Section 5 analyzes the security. Section 6 provides some comparisons. Section 7 concludes the entire paper.

2. Preliminaries

In this section, we introduce some background knowledge that is crucial for understanding our scheme, including the classical Shamir (t, n)-threshold secret sharing protocol [34] and a brief overview of the TQHE scheme framework.

2.1. Shamir (t, n)-Threshold Secret Sharing Protocol

The principle of Shamir (t, n)-threshold secret sharing protocol consists of two core algorithms as follow.

(1) **The share generation algorithm:** Given a finite field $\mathrm{GF}(d)$, where d is a large prime number, the trusted data owner Alice randomly selects a $t-1$ degree polynomial:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \bmod d, \tag{1}$$

where $(a_0, a_1, a_2, \ldots, a_{t-1}) \in \mathrm{GF}(d)$, and a_0 is a secret integer. Alice selects n nonzero and distinct elements $\{x_i \in \mathrm{GF}(d) | i = (1, 2, \ldots, n)\}$ to compute n secret shares $\{f(x_i) \in \mathrm{GF}(d) | i = (1, 2, \ldots, n)\}$, and then securely transmits them to n participants $R = \{Bob_{g_1}, Bob_{g_2}, \cdots, Bob_{g_n}\}, g_i \in \{1, 2, \ldots, n\} (0 < i \leq n)$ through secure classical channels, ensuring that each participant holds a private share $f(x_i) \in \mathrm{GF}(d) (i = 1, 2, \ldots, n)$.

(2) **The secret reconstruction algorithm**: If any t participants $\{Bob_1, Bob_2, \cdots, Bob_t\}$ out of n participants want to reconstruct the secret a_0, each participant Bob_i in $\{Bob_1, Bob_2, \cdots, Bob_t\}$ first takes out their private share $f(x_i)$, and then uses the following Lagrange interpolation equation:

$$f(x) = \left(\sum_{r=1}^{t} f(x_r) \prod_{1 \leq j \leq t, j \neq r} \frac{x - x_j}{x_r - x_j} \right) \bmod d, \tag{2}$$

to cooperate in reconstructing and calculating the secret information. In Equation (2), if $x = 0$, then

$$a_0 = f(0) = \left(\sum_{r=1}^{t} f(x_r) \prod_{1 \leq j \leq t, j \neq r} \frac{x_j}{x_j - x_r} \right) \bmod d, \tag{3}$$

is the original secret integer.

In Equation (3), the secret information a_0 can be reconstructed by t participants using their private shares after cooperative computation. k $(t \leq k \leq n)$ participants can still accurately reconstruct the secret information a_0 through cooperative computation using their private shares.

$$\begin{aligned} a_0 &= \left(\sum_{r=1}^{t} f(x_r) \prod_{1 \leq j \leq t, j \neq r} \frac{x_j}{x_j - x_r} \right) \bmod d \\ &= \left(\sum_{r=1}^{t+1} f(x_r) \prod_{1 \leq j \leq t+1, j \neq r} \frac{x_j}{x_j - x_r} \right) \bmod d \\ &= \cdots \\ &= \left(\sum_{r=1}^{k} f(x_r) \prod_{1 \leq j \leq k, j \neq r} \frac{x_j}{x_j - x_r} \right) \bmod d \\ &= \cdots \\ &= \left(\sum_{r=1}^{n} f(x_r) \prod_{1 \leq j \leq n, j \neq r} \frac{x_j}{x_j - x_r} \right) \bmod d. \end{aligned} \tag{4}$$

2.2. Definition of TQHE

Based on the existing TQHE schemes [31,32], we provide a specific definition for TQHE as follows.

Definition 1 (TQHE framework [31,32]). *A TQHE scheme consists of four main steps:*

(1) **Key Generation.** *TQHE.KeyGen* $(1^\kappa) \rightarrow (sk_e, \rho_{evk_i}, sk_d)$. *It is used to generate a series of keys, which include the encryption private key* sk_e *and the decryption private key* sk_d *for the data owner Alice, and evaluation keys* ρ_{evk_i} $(i = 1, 2, \ldots, k)$ *for the* k *evaluators* $\{Bob_1, Bob_2, \cdots, Bob_k\}$;

(2) **Encryption.** $TQHE.Enc_{sk_e}(|m_a\rangle) \rightarrow (|c_a\rangle)$. *Alice runs the* $TQHE.Enc_{sk_e}(|m_a\rangle)$ *to encrypt her original plaintext quantum state sequence* $|m_a\rangle$ *using her encryption key* sk_e *to obtain the original ciphertext quantum state sequence* $|c_a\rangle$;

(3) **Homomorphic Evaluation.** $TQHE.Eval\ (\chi_i, \rho_{evk_i}, |c_a\rangle) \rightarrow (|c_b\rangle)$. *It is used to process the quantum ciphertext state sequence without decryption. After Alice shares the encrypted ciphertext quantum state sequence* $|c_a\rangle$ *to* k $(t \leq k \leq n)$ *evaluators* $\{Bob_1, Bob_2, \cdots, Bob_k\}$, *they run the* $TQHE.Eval\ (\chi_i, \rho_{evk_i}, |c_a\rangle)$ *sequentially to obtain the final homomorphic ciphertext quantum state sequence* $|c_f\rangle$;

(4) **Decryption.** $TQHE.Dec_{sk_d}(|c_f\rangle) \rightarrow (|m_f\rangle)$. *Alice runs the* $TQHE.Dec_{sk_d}(|c_f\rangle)$ *to decrypt the final ciphertext quantum state sequence* $|c_f\rangle$ *using her decryption key* sk_d *to obtain the expected plaintext quantum state sequence* $|m_f\rangle = \prod_{i=1}^{k} \chi_i |c_a\rangle$.

Here, $|m_a\rangle, |m_f\rangle \in$ *the quantum message space* M, *and* $c_a, c_f \in$ *the quantum ciphertext space* C. $\chi_i \in$ *the set of quantum gates* $\mathcal{F}_\Delta$, *i.e.,* $\chi_i \in \mathcal{F}_\Delta$. *As for the security of a TQHE scheme, it should satisfy indistinguishability under chosen-plaintext attacks (q-IND-CPA) in quantum polynomial time (QPT). Hence, a TQHE scheme is said to be q-IND-CPA secure if for any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *there exists a negligible function satisfying* $\Pr[PubK^{cpa}_{\mathcal{A},TQHE}(\kappa) = 1] \leq \frac{1}{2} + negl(\kappa)$, *where* $PubK^{cpa}_{\mathcal{A},TQHE}$ *is a model of quantum indistinguishability under CPA [14].*

Definition 2 (Quantum indistinguishability under CPA). *The game model of quantum indistinguishability under chosen-plaintext attack (IND-CPA)* $PubK^{cpa}_{\mathcal{A},TQHE}(\kappa)$ *for a TQHE scheme and a QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *is defined as*

(1) The challenger runs $TQHE.KeyGen\ (1^\kappa) \rightarrow (sk_e, \rho_{evk_j}, sk_d), j \in [1, k] \cap \mathbb{Z}$.

(2) The challenger sends $\rho_{evk_i}, i \in [2, k] \cap \mathbb{Z}$ *to* $\mathcal{A}_1$. *Then,* $\mathcal{A}_1$ *outputs a quantum state in* $\mathcal{M} \otimes \mathcal{E}$, *where* $\mathcal{M}$ *is the message space and* $\mathcal{E}$ *is an arbitrary state related to the environment.*

(3) For $r \in \{0, 1\}$, *let* $\Xi^{cpa,r}_{TQHE} : D(\mathcal{M}) \rightarrow D(\mathcal{C})$ *be* $\Xi^{cpa,0}_{TQHE}(\rho) = TQHE.Enc_{sk_e}(|0\rangle\langle 0|)$ *and* $\Xi^{cpa,1}_{TQHE}(\rho) = TQHE.Enc_{sk_e}(\rho)$. *A random bit* $r \in \{0, 1\}$ *is chosen and* $\Xi^{cpa,0}_{TQHE}(\rho)$ *is applied to the state in* $\mathcal{M}$.

(4) $\mathcal{A}_2$ *obtains the state in* $\mathcal{C} \otimes \mathcal{E}$ *and outputs a bit* r'.

(5) The output of the game is defined as 1 if $r' = r$ *and 0 otherwise. If* $r = r'$, $\mathcal{A}_2$ *wins the game.*

3. Our Scheme

In this section, we have proposed a novel TQHE scheme based on the Shamir (t, n)-threshold secret sharing protocol. It has a flexible number of evaluators, supporting k $(t \leq k \leq n)$ evaluators to perform arbitrary single-qubit gate unitary operation (any evaluation computation task) from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$ on the ciphertext quantum state sequence. Figure 1 shows the main flow chart of the proposed scheme.

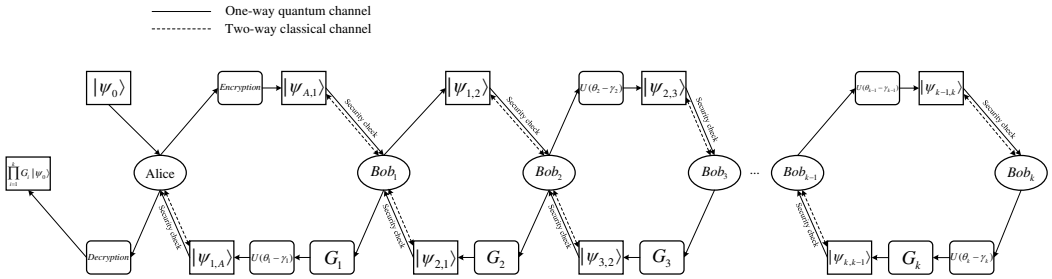


Figure 1. The main process of the (t, n)-threshold QHE scheme we proposed, where *Encryption*/*Decryption* refers to the encryption unitary operation $U(\theta_A)$ and decryption unitary operation $U(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)$, respectively.

The Proposed TQHE Scheme

In our TQHE scheme, initially, Alice encrypts her original plaintext quantum state sequence into the ciphertext quantum state sequence. Subsequently, k $(t \leq k \leq n)$ evaluators sequentially perform the evaluation computation task on the ciphertext quantum state sequence encrypted by Alice. Then, the last evaluator sends the final ciphertext quantum state sequence to Alice. Finally, Alice decrypts it to obtain the result computed on the original plaintext quantum state sequence. The detailed procedure of our proposed TQHE scheme is given as follows.

Step 1: Key generation stage

In this stage, Alice executes the key generation algorithm [35] to generate a series of initial keys.

(1) **The shadow key generation sub-algorithm**

Initially, Alice runs the shadow key generation sub-algorithm to randomly create a polynomial of degree $t-1$, denoted as $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \mod d$, where $\mathrm{GF}(d)$ is a finite field, d is a randomly selected large prime number, and $(a_0, a_1, a_2, \ldots, a_{t-1}) \in \mathrm{GF}(d)$.

Afterwards, Alice randomly selects n different elements $\{x_{g_i} \in \mathrm{GF}(d) | g_i \in \{1, 2, \ldots, n\}\}$ $(0 < i \leq n)$ as inputs for the polynomial $f(x)$ to sequentially obtain the shadow keys $\{wk_{g_i} = f(x_i) \in \mathrm{GF}(d) | i = (1, 2, \ldots, n)\}$ for each of the n participants $\{Bob_{g_1}, Bob_{g_2}, \ldots, Bob_{g_n}\}$. Subsequently, Alice selects any k $(t \leq k \leq n)$ participants $\{Bob_1, Bob_2, \ldots, Bob_k\}$ from the group of the n participants $\{Bob_{g_1}, Bob_{g_2}, \ldots, Bob_{g_n}\}$ as k evaluators, with Bob_1 being the first randomly chosen evaluator. Finally, Alice computes the private keys $\theta_i (i = 1, 2, \ldots, k)$ of k evaluators according to Equations (5) and (6), where

$$\theta_i = 2\pi \cdot \frac{L_i wk_i}{d} \bmod d, \tag{5}$$

$$L_i = \prod_{1 \leq r \leq k, r \neq i} \frac{x_r}{x_r - x_i} \bmod d. \tag{6}$$

The private key θ_1 of Bob_1 is kept secretly by Alice, while the remaining $k-1$ private keys $\{\theta_2, \theta_3, \ldots, \theta_k\}$ of $\{Bob_2, Bob_3 \ldots, Bob_k\}$ are secretly distributed by Alice over secure classical channels.

(2) **The rotation key generation sub-algorithm**

Alice runs the rotation key generation sub-algorithm to generate k randomly different rotation keys $\{\gamma_i \in [0, 2\pi] | i = 1, 2, \ldots, k\}$ and then distributes the $k-1$ rotation keys $\{\gamma_2, \gamma_3, \ldots, \gamma_k\}$ to the $k-1$ evaluators $\{Bob_2, Bob_3, \ldots, Bob_k\}$ through secure classical channels. Similarly, the rotation key γ_1 of Bob_1 is kept secretly by Alice.

Step 2: Encryption sharing stage

(1) The data owner Alice has an original $|\psi_0\rangle$ plaintext quantum state sequence of length m:

$$|\psi_0\rangle = \{|\varphi_{0,u}\rangle = \alpha_{0,u}|0\rangle + \beta_{0,u}|1\rangle | u = 1, 2, \ldots, m\}, \tag{7}$$

where $|\alpha_{0,u}|^2 + |\beta_{0,u}|^2 = 1$. She defines a 2×2 unitary matrix $U(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$, and an angle parameter "θ" is introduced to describe a specific transformation or operation of the matrix. The behavior of this matrix will depend on the specific value of "θ". $U(\theta)$ can be used to achieve various types of qubit operations, such as phase shift transformation of the quantum state. Therefore, by adjusting "θ", we can perform different phase shift unitary operations on the quantum state, which in turn enables the protection of private data. Performing the phase shift unitary operation $U(\theta)$ on a qubit state $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ satisfies Equation (8):

$$U(\theta_1)U(\theta_2)|\varphi\rangle = U(\theta_1 + \theta_2)|\varphi\rangle = \begin{pmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{pmatrix} |\varphi\rangle. \tag{8}$$

(2) Alice randomly selects $c \in \mathbb{Z}$ and uses $\theta_A = 2\pi - 2\pi \cdot \frac{a_0}{d} - \gamma_A$ as her encryption private key where $\gamma_A = 2c\pi - \theta_1 - \gamma_2 - \gamma_3 - \cdots - \gamma_k$, $\theta_1 = 2\pi \cdot \frac{L_1 w k_1}{d} \bmod d$ and $L_1 = \prod_{1 \le r \le k, r \ne 1} \frac{x_r}{x_r - x_1} \bmod d$. Next, she performs the encryption unitary operation $U(\theta_A)$ on the plaintext quantum state sequence $|\psi_0\rangle$, resulting in the original encrypted ciphertext quantum state sequence

$$\begin{aligned} |\psi_{A,1}\rangle &= U(\theta_A)|\psi_0\rangle \\ &= U(\theta_A)\{|\varphi_{0,1}\rangle, |\varphi_{0,2}\rangle, \ldots, |\varphi_{0,m}\rangle\} \\ &= \{|\varphi_{A,1,1}\rangle, |\varphi_{A,1,2}\rangle, \ldots, |\varphi_{A,1,3}\rangle\}. \end{aligned} \tag{9}$$

(3) Alice randomly prepares some decoy particles from states $\{|0\rangle, |1\rangle, |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}\}$, randomly inserts these decoy particles into the ciphertext quantum state sequence $|\psi_{A,1}\rangle$ to obtain a new quantum state sequence $|\psi_{A,1}\rangle'$, and meticulously records the insertion positions and initial states of each decoy particle. These decoy particles are primarily used for the security check during the transmission of quantum ciphertext sequences [36]. Subsequently, Alice transmits $|\psi_{A,1}\rangle'$ to the first evaluator Bob_1 over a quantum channel. After confirming that Bob_1 has received $|\psi_{A,1}\rangle'$, Alice informs Bob_1 of the positions where each decoy particle was inserted and the corresponding measurement basis (X-basis or Z-basis). Afterwards, Bob_1 measures the states of these decoy particles using the corresponding basis (X-basis or Z-basis) and sends the measurement results to Alice. Alice compares whether the results from Bob_1 match the initially states of the decoy particles. If the error rate is below a certain low threshold value, Alice instructs Bob_1 to proceed to the next step; otherwise, Alice will ask Bob_1 to abort the process and restart a new one. It is important to note that, during each transmission of the quantum state sequence, both the sender and the receiver should conduct similar security check procedures and this process will not be described repeatedly in the following text.

(4) Following the successful security check, Bob_1 removes the decoy particles from the sequence $|\psi_{A,1}\rangle'$ to obtain the ciphertext quantum state sequence $|\psi_{A,1}\rangle$. During this stage, Bob_1 temporarily refrains from performing the evaluation computation operation and the phase shift unitary operation on the sequence $|\psi_{A,1}\rangle$. Instead, he randomly re-inserts decoy particles into the sequence $|\psi_{A,1}\rangle$ to obtain a new quantum state sequence $|\psi_{1,2}\rangle'$ and then transmits it to the next evaluator Bob_2 through a quantum channel.

(5) After receiving $|\psi_{1,2}\rangle'$ on security, Bob_2 removes the decoy particles to obtain the quantum state sequence $|\psi_{1,2}\rangle = |\psi_{A,1}\rangle$. Then, Bob_2 applies the phase shift unitary operation $U(\theta_2 - \gamma_2)$ on each individual qubit in the quantum state sequence $|\psi_{1,2}\rangle$, resulting in the quantum state sequence

$$|\psi_{2,3}\rangle = U(\theta_2 - \gamma_2)|\psi_{1,2}\rangle. \tag{10}$$

Next, Bob_2 sends $|\psi_{2,3}\rangle'$ formed by inserted decoy particles into $|\psi_{2,3}\rangle$ to the next evaluator Bob_3. Similarly, $Bob_3, Bob_4, \ldots, Bob_{k-1}$ perform the security check and the phase shift operation $U(\theta_3 - \gamma_3), U(\theta_4 - \gamma_4), \ldots, U(\theta_{k-1} - \gamma_{k-1})$ on their respective received quantum state sequences $|\psi_{2,3}\rangle, |\psi_{3,4}\rangle, \ldots, |\psi_{k-2,k-1}\rangle$. Finally, Bob_{k-1} sends the quantum state sequence $|\psi_{k-1,k}\rangle'$ with inserted decoy particles to the k-th evaluator Bob_k.

Step 3: Evaluation computation stage

In this stage, Alice sends the order in which k ($t \leq k \leq n$) evaluators perform calculations to them separately through a secure classical channel. Moreover, Alice divides the evaluation computation tasks $G = G_1 G_2 \cdots G_k$ that she wishes to perform on the original plaintext quantum state sequence into k segments, and distributes the segmented task $G_i = \prod_{v=1}^{p} V_v$, ($V_v \in \{X, Y, Z, H, S, T\}$, $p \in \mathbb{Z}_+$, and $i = 1, 2, \ldots, k$) to the i-th evaluator Bob_i over secure classical channels for the subsequent evaluation computation. In our scheme, the single-qubit gate unitary operation assigned to the evaluators comes from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$, where $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ and $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{pmatrix}$. The set $\{X, Y, Z, H, S, T\}$ is the universal set of single-qubit gates. These single-qubit gates can be used to approximate any single-qubit unitary operation with arbitrary precision [37]. A quantum state of a single qubit can be represented as $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying $|\alpha\rangle^2 + |\beta\rangle^2 = 1$. These universal single-qubit gates are used to manipulate individual qubits to implement an arbitrary single-qubit transformation in the evaluation computation stage. Evaluators can utilize these gates to rotate and transform the state of qubits, enabling the execution of various quantum evaluation computation tasks. Our scheme involves performing the evaluation computation tasks G on a sequence of quantum states. This is achieved by performing single-qubit gate unitary operations from the set of single-qubit gates $\{X, Y, Z, H, S, T\}$ on each qubit of the quantum state sequence. The k evaluators sequentially perform the evaluation computation task G_i ($0 < i \leq k$) (the composite single-qubit gate unitary operation) assigned to them by Alice, and Bob_1 finally sends the final ciphertext quantum state sequence after completing the evaluation computation tasks for Alice.

(1) After the quantum state sequence $|\psi_{k-1,k}\rangle'$ successfully passes the security check, Bob_k removes the decoy particles to obtain the quantum state sequence $|\psi_{k-1,k}\rangle$. Next, he sequentially applies the phase shift unitary operation $U(\theta_k - \gamma_k)$ and performs the evaluation computation operation G_k on the sequence $|\psi_{k-1,k}\rangle$ to obtain the quantum state sequence

$$|\psi_{k,k-1}\rangle = G_k U(\theta_k - \gamma_k)|\psi_{k-1,k}\rangle. \tag{11}$$

Bob_k then sends $|\psi_{k,k-1}\rangle'$ obtained after inserting the decoy particles in $|\psi_{k,k-1}\rangle$ to the next evaluator Bob_{k-1}.

(2) Following the successful security check, Bob_{k-1} performs the evaluation computation operation G_{k-1} on the received sequence $|\psi_{k,k-1}\rangle$ and obtains the quantum state sequence

$$|\psi_{k-1,k-2}\rangle = G_{k-1}|\psi_{k,k-1}\rangle. \tag{12}$$

Bob_{k-1} then sends $|\psi_{k-1,k-2}\rangle^{'}$ formed by inserted decoy particles into $|\psi_{k-1,k-2}\rangle$ to the next evaluator Bob_{k-2}.

(3) $Bob_{k-2}, Bob_{k-3}, \ldots, Bob_2$ do something similar to Bob_{k-1}. After successfully passing the security check, they perform the evaluation computation operation $G_{k-2}, G_{k-3}, \cdots, G_2$ on the received sequence $|\psi_{k-1,k-2}\rangle, |\psi_{k-2,k-3}\rangle, \ldots, |\psi_{3,2}\rangle$, respectively. In the end, Bob_2 sends $|\psi_{2,1}\rangle^{'}$ obtained after inserting the decoy particles in $|\psi_{2,1}\rangle$ to the next evaluator Bob_1.

(4) After passing the successful security check, Bob_1 sequentially performs the evaluation computation operation G_1 and the phase shift unitary operation $U(\sigma)$ ($\sigma = \theta_1 - \gamma_1$ is secretly calculated by Alice and then secretly shared with Bob_1, who has no knowledge of the private key θ_1 and the rotation key γ_1) on the sequence $|\psi_{2,1}\rangle$, resulting in the quantum state sequence

$$|\psi_{1,A}\rangle = U(\sigma)G_1|\psi_{2,1}\rangle. \tag{13}$$

Finally, Bob_1 sends $|\psi_{1,A}\rangle^{'}$ obtained after inserting the decoy particles in $|\psi_{1,A}\rangle$ to the data owner Alice.

Step 4: Decryption stage

After securely receiving $|\psi_{1,A}\rangle$ resulting from the collaborative computations of k ($t \leq k \leq n$) evaluators, sent by Bob_1 who performed the last evaluation computation operation, Alice uses her rotation key γ_A and the random rotation keys $\{\gamma_i | i = 1, 2, \ldots, k\}$ of the k evaluators to perform the decryption unitary operation $D = U(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)$ on the sequence $|\psi_{1,A}\rangle$. Taking the u-th quantum bit $|\varphi_{1,A,u}\rangle$ of the decryption sequence $|\psi_{1,A}\rangle$ as an example, the decryption process is illustrated below:

$$\begin{aligned}
&D|\varphi_{1,A,u}\rangle = U(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)|\varphi_{1,A,u}\rangle \\
&= U(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)U(\theta_1 - \gamma_1)G_1G_2 \cdots G_kU(\theta_k - \gamma_k) \cdots U(\theta_2 - \gamma_2)|\varphi_{1,A,u}\rangle \\
&= G_1G_2 \cdots G_kU(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)U(\theta_1 - \gamma_1)U(\theta_k - \gamma_k) \cdots U(\theta_2 - \gamma_2)U(\theta_A)|\varphi_{0,u}\rangle \\
&= G_1G_2 \cdots G_kU(\gamma_A + \theta_A + \frac{2\pi}{d} \cdot a_0)|\varphi_{0,u}\rangle \\
&= \prod_{i=1}^{k} G_iU(2\pi)|\varphi_{0,u}\rangle \\
&= \prod_{i=1}^{k} G_i|\varphi_{0,u}\rangle.
\end{aligned} \tag{14}$$

In the above Equation (14), it should be noted that $U(\gamma_A + \gamma_1 + \gamma_2 + \cdots + \gamma_k)U(\theta_1 - \gamma_1) = U(\gamma_A + \theta_1 + \gamma_2 + \cdots + \gamma_k) = I$. Apparently, Alice performs the decryption unitary operation on the final ciphertext quantum state sequence $|\psi_{1,A}\rangle$ and obtains the correct result, which is the same as that obtained by performing evaluation computation directly on the plaintext quantum state sequence $|\psi_0\rangle$:

$$D|\psi_{1,A}\rangle = \prod_{i=1}^{k} G_i|\psi_0\rangle. \tag{15}$$

Specifically, any k ($t \leq k \leq n$) evaluators selected from n evaluators sequentially perform the evaluation computation operation on the ciphertext quantum state sequence encrypted by data owner Alice. The last evaluator, after completing the collaborative computations, sends the final ciphertext quantum state sequence to Alice. Alice decrypts

it to obtain the expected result of computing on the plaintext quantum state sequence. In other words, the result of Alice's decryption operation on the final ciphertext quantum state sequence is the same as the result of Alice performing the same evaluation computation operations directly on the plaintext quantum state sequence.

4. Example and Simulation

In Section 3, we proposed a novel TQHE scheme and analyzed its correctness. To better illustrate the scheme, in this section, we first present a specific example of (3, 5)-threshold QHE to demonstrate the correctness and feasibility of the TQHE scheme (Figure 2). Subsequently, the correctness of the example is verified by running simulation experiments on the IBM quantum computing cloud platform.

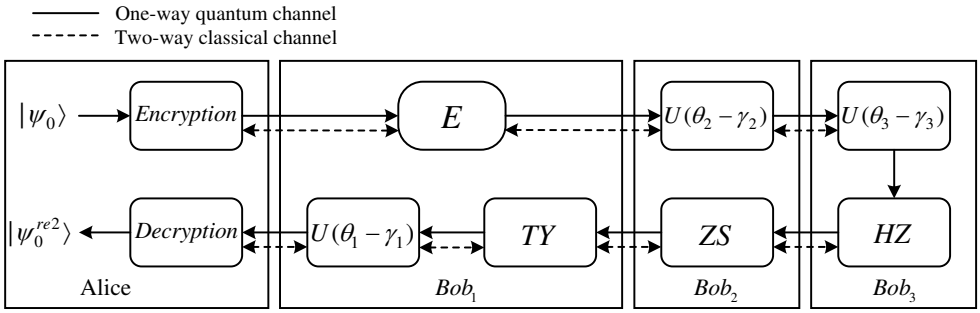


Figure 2. The example process of (3, 5)-threshold QHE for the proposed scheme, where $Encryption = U(\theta_A)$, $Decryption = U(\gamma_A + \gamma_1 + \gamma_2 + \gamma_3)$, and E indicates that no evaluation computation operation and unitary operation are being performed.

4.1. Example

In this example, suppose Alice wants at least three evaluators to collaborate on the evaluation computation tasks G. First, Alice selects three participants $\{Bob_1, Bob_2, Bob_3\}$ out of the five participants $\{Bob_{g_1}, Bob_{g_2}, Bob_{g_3}, Bob_{g_4}, Bob_{g_5}\}$ as evaluators to perform the evaluation computation operations on Alice's encrypted quantum state sequence $|\psi_{A,1}\rangle$. After completing the collaborative computations, the last evaluator Bob_1 sends the final ciphertext quantum state sequence $|\psi_{1,A}\rangle$ to Alice. Finally, Alice can obtain the expectant result by decrypting $|\psi_{1,A}\rangle$. In the following example, the security check steps are excluded.

First, Alice generates a series of keys. By inputting parameters $n = 5, t = 3, d = 7$ into the shadow key generation sub-algorithm, she obtains a random quadratic polynomial $f(x) = (2 + 3x + x^2) \bmod 7$ with coefficients $\{a_0 = 2, a_1 = 3, a_2 = 1\}$, where a_0 is the secret information. Next, by inputting five parameters $\{x_{g_1} = 1, x_{g_2} = 3, x_{g_3} = 5, x_{g_4} = 2, x_{g_5} = 4\}$ into the polynomial $f(x)$, she obtains five shadow keys $\{wk_{g_1} = f(x_{g_1}) = f(1) = 6, wk_{g_2} = f(3) = 6, wk_{g_3} = f(5) = 0, wk_{g_4} = f(2) = 5, wk_{g_5} = f(4) = 2\}$.

Suppose Alice selects three evaluators $\{Bob_1, Bob_2, Bob_3\}$ to perform the evaluation computation operation tasks $G = TYZSHZ$ on the original plaintext quantum state sequence $|\psi_0\rangle = \frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$, which consists of only a single quantum bit.

First, Alice computes the private keys $\{\theta_1 = 2\pi \cdot \frac{L_1 wk_1}{d} = 2\pi \cdot \frac{1}{d}((f(x_1)\frac{x_2 \cdot x_3}{(x_2-x_1)(x_3-x_1)}) \bmod d) = \frac{8}{7}\pi, \theta_2 = \frac{12}{7}\pi, \theta_3 = \frac{12}{7}\pi\}(x_1 = x_{g_1} = 1, x_2 = x_{g_4} = 2, x_3 = x_{g_2} = 3; wk_2 = wk_{g_4} = f(x_{g_4}) = 5, wk_3 = wk_{g_2} = f(x_{g_2}) = 6)$ and generates the rotation keys $\{\gamma_1 = \frac{17}{21}\pi, \gamma_2 = \frac{55}{28}\pi, \gamma_3 = \frac{65}{42}\pi\}$ by running the rotation key generation sub-algorithm. Later, Alice communicates with $\{Bob_1, Bob_2, Bob_3\}$ to instruct them to perform the evaluation computation operation $\{G_1 = TY, G_2 = ZS, G_3 = HZ\}$, and secretly distributes their respective shadow keys $\{\theta_2, \theta_3\}$ and rotation keys $\{\gamma_2, \gamma_3\}$ to the evaluators $\{Bob_2, Bob_3\}$ through secure classical channels. Bob_1's private key θ_1 and rotation key γ_1 are kept secretly by Alice.

Next, Alice utilizes her encryption private key $\theta_A = 2\pi - 2\pi \cdot \frac{a_0}{d} - \gamma_A = \frac{343}{84}\pi$ ($\gamma_A = 2\pi - \theta_1 - \gamma_2 - \gamma_3 = -\frac{223}{84}\pi$) to perform the encryption unitary operation $U(\theta_A)$ on the plaintext quantum state sequence $|\psi_0\rangle$ and obtains the encrypted quantum state sequence

$$|\psi_{A,1}\rangle = U(\theta_A)|\psi_0\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle, \tag{16}$$

and then transmits it to Bob_1.

After receiving $|\psi_{A,1}\rangle$, instead of performing the evaluation computation operation and the phase shift unitary operation on it, Bob_1 directly transmits $|\psi_{A,1}\rangle$ (renamed to $|\psi_{1,2}\rangle$) to Bob_2.

After receiving $|\psi_{1,2}\rangle$, Bob_2 applies the phase shift unitary operation $U(\theta_2 - \gamma_2)$ on $|\psi_{1,2}\rangle$, yielding

$$|\psi_{2,3}\rangle = U(\theta_2 - \gamma_2)|\psi_{1,2}\rangle = \frac{\sqrt{2}+\sqrt{6}}{4}|0\rangle + \frac{\sqrt{6}-\sqrt{2}}{4}|1\rangle, \tag{17}$$

and then sends the quantum state sequence $|\psi_{2,3}\rangle$ to Bob_3.

After receiving $|\psi_{3,2}\rangle$, Bob_2 performs the evaluation computation operation G_2 on $|\psi_{3,2}\rangle$, yielding

$$|\psi_{2,1}\rangle = G_2|\psi_{3,2}\rangle = -i|1\rangle, \tag{18}$$

and then sends the quantum state sequence $|\psi_{2,1}\rangle$ to Bob_1.

After receiving $|\psi_{2,1}\rangle$, Bob_1 sequentially performs the evaluation computation operation G_1 and the phase shift unitary operation $U(\sigma)$ ($\sigma = \theta_1 - \gamma_1$ is secretly calculated by Alice and then secretly shared with Bob_1) on $|\psi_{2,1}\rangle$, yielding

$$|\psi_{1,A}\rangle = U(\sigma)G_1|\psi_{2,1}\rangle = -\frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle, \tag{19}$$

and then sends the quantum state sequence $|\psi_{1,A}\rangle$ to Alice.

Finally, Alice performs the decryption unitary operation $D = U(\gamma_A + \gamma_1 + \gamma_2 + \gamma_3)$ on the quantum state sequence $|\psi_{1,A}\rangle$ resulting from the cooperative evaluation computation operations by the three evaluators, and obtains the final result

$$|\psi_0^{re2}\rangle = D|\psi_{1,A}\rangle = U(\gamma_A + \gamma_1 + \gamma_2 + \gamma_3)|\psi_{1,A}\rangle = -|0\rangle. \tag{20}$$

Now, let us analyze the result of directly using the evaluation computation operator G on the original plaintext quantum state sequence $|\psi_0\rangle$. Obviously, the result is

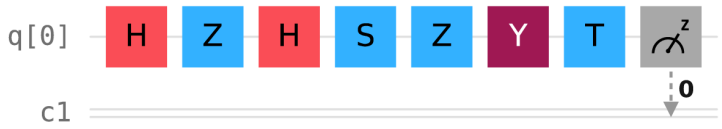
$$|\psi_0^{re1}\rangle = G|\psi_0\rangle = TYZSHZ|\psi_0\rangle = -|0\rangle. \tag{21}$$

From here, we see that Alice's decrypted result $|\psi_0^{re2}\rangle$ is the same as the result $|\psi_0^{re1}\rangle$ obtained by direct computation on the plaintext quantum state. Theoretically, if Alice measures $|\psi_0^{re2}\rangle$ in Z-basis, she can obtain a measurement result of 0 with the probability 100%.

4.2. Simulation Experiment

On the IBM quantum computing cloud platform, we experimentally verify the correctness and feasibility of the given (3, 5)-threshold QHE example. The quantum circuit diagram for this example is shown in Figure 3b. Considering that the IBM quantum computing cloud platform does not allow operations to be performed on an arbitrary quantum state (the initial quantum state of a single quantum circuit in the IBM quantum computing cloud platform is the $|0\rangle$ state) and has some limitations on the number and spatial

dimensions of quantum states [38–40], the experimental verification process of inserting and removing decoy particles is omitted in this section.



(**a**) Quantum circuit 1 on IBM quantum backend

(**b**) Quantum circuit 2 on IBM quantum backend

Figure 3. The quantum circuit diagram of the composite single-qubit gates $TYZSHZ$ directly acting on $|\psi_0\rangle$ in the IBM quantum computing cloud platform in the subplot (**a**). The quantum circuit diagram of the (3, 5)-threshold QHE example in the IBM quantum computing cloud platform in the subplot (**b**), where $RY(\theta)$ represents the phase shift unitary operation $U(\theta)$.

Five rounds of experiments are conducted on the (3, 5)-threshold example using two different backends, 'ibm_brisbane' and 'ibm_lagos', and three different measurement shots, 1024, 4096 and 8192, for the quantum circuit diagram presented in Figure 3b. The experimental measurement results are shown in Table 1.

On the one hand, on the quantum computing cloud platform, we measure the results of the composite evaluation computation operations $G_1G_2G_3$ directly acting on Alice's original plaintext quantum state $|\psi_0\rangle$. The quantum circuit diagram is shown in Figure 3a.

On the other hand, we simulate the entire process of the (3, 5)-threshold QHE example on the quantum computing cloud platform and measure the results of performing all the evaluation computation operations and the unitary operations. The quantum circuit diagram is shown in Figure 3b.

The results of the 1024 measurements performed on the backend 'ibm_brisbane' and 'ibm_logo' after applying the composite single quantum bit gate $G = G_1G_2G_3$ directly to the original quantum plaintext quantum state $|\psi_0\rangle = \frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$ have a high probability (99.2% and 100%) corresponding to the state $|0\rangle$, respectively.

Table 1. Comparison of experimental results using two different backends and different shots for the example.

Running Environment	Result (a)	Result (b)	Result (c)	Result (d)	Result (e)
Backend1: ibm_brisbane	$\|0\rangle$: 99.8%	$\|0\rangle$: 99.3%	$\|0\rangle$: 99.6%	$\|0\rangle$: 99.4%	$\|0\rangle$: 99.6%
Shots: 1024	$\|1\rangle$: 0.2%	$\|1\rangle$: 0.7%	$\|1\rangle$: 0.4%	$\|1\rangle$: 0.6%	$\|1\rangle$: 0.4%
Backend1: ibm_brisbane	$\|0\rangle$: 99.8%	$\|0\rangle$: 99.5%	$\|0\rangle$: 99.4%	$\|0\rangle$: 99.9%	$\|0\rangle$: 99.7%
Shots: 4096	$\|1\rangle$: 0.2%	$\|1\rangle$: 0.5%	$\|1\rangle$: 0.6%	$\|1\rangle$: 0.1%	$\|1\rangle$: 0.3%
Backend1: ibm_brisbane	$\|0\rangle$: 99.9%	$\|0\rangle$: 99.5%	$\|0\rangle$: 99.2%	$\|0\rangle$: 99.7%	$\|0\rangle$: 99.8%
Shots: 8192	$\|1\rangle$: 0.1%	$\|1\rangle$: 0.5%	$\|1\rangle$: 0.8%	$\|1\rangle$: 0.3%	$\|1\rangle$: 0.2%
Backend2: ibm_lagos	$\|0\rangle$: 99.6%	$\|0\rangle$: 99.8%	$\|0\rangle$: 99.3%	$\|0\rangle$: 99.5%	$\|0\rangle$: 99.4%
Shots: 1024	$\|1\rangle$: 0.4%	$\|1\rangle$: 0.2%	$\|1\rangle$: 0.7%	$\|1\rangle$: 0.5%	$\|1\rangle$: 0.6%
Backend2: ibm_lagos	$\|0\rangle$: 99.9%	$\|0\rangle$: 99.5%	$\|0\rangle$: 99.6%	$\|0\rangle$: 99.4%	$\|0\rangle$: 99.5%
Shots: 4096	$\|1\rangle$: 0.1%	$\|1\rangle$: 0.5%	$\|1\rangle$: 0.4%	$\|1\rangle$: 0.6%	$\|1\rangle$: 0.5%
Backend2: ibm_lagos	$\|0\rangle$: 99.8%	$\|0\rangle$: 99.7%	$\|0\rangle$: 99.9%	$\|0\rangle$: 99.4%	$\|0\rangle$: 99.7%
Shots: 8192	$\|1\rangle$: 0.2%	$\|1\rangle$: 0.3%	$\|1\rangle$: 0.1%	$\|1\rangle$: 0.6%	$\|1\rangle$: 0.3%

In all the measurement results in Table 1, the results of the 8192 measurements performed on the backend 'ibm_brisbane' have the smallest probability (99.2%) corresponding to the state $|0\rangle$, which shows that the fidelity should be the smallest in this situation. Assuming the theoretical density matrix is denoted by $\rho^T = 1.0|0\rangle\langle 0| + 0.0|1\rangle\langle 1|$, and the experimental density matrix is denoted by $\rho^E = 0.992|0\rangle\langle 0| + 0.008|1\rangle\langle 1|$, the smallest fidelity can be calculated as follows:

$$F(\rho^T, \rho^E) = Tr\sqrt{\sqrt{\rho^T} \cdot \rho^E \cdot \sqrt{\rho^T}} = 99.5\%, \tag{22}$$

the fidelity of 99.5% is very close to the theoretical value of 100%. In the five rounds of experiments, the 99.5% fidelity is the lowest, and the fidelities of the other twenty-nine measurement results are higher than the fidelity of this particular measurement. By calculation, the average fidelity of the thirty experiments is 99.8%, which implies that all the obtained fidelity values from the measurements are very close to the theoretical value of 100%. Thus, the experimental results further validate the correctness of the (3, 5)-threshold QHE example and the feasibility of the proposed TQHE scheme.

5. Security Analysis

In this section, we analyze the security of the proposed TQHE scheme in three aspects. The analysis includes the security of the encryption/decryption private keys, ciphertext quantum state sequences during transmission, the plaintext quantum state sequence, and the result after computations on the plaintext quantum state sequence.

(1) **The security of the encryption/decryption private keys**

Theorem 1. *The proposed TQHE scheme is perfect with respect to the probability distribution of the encryption private key θ_e over the private key space, that is,*

$$I(\theta_e;\Omega) = H(\theta_e) - H(\theta_e|\Omega) = 0, \tag{23}$$

where Ω denotes the set of key information distributed to the evaluators, $H(\theta_e)$ is the information entropy of the encryption private key θ_e and $I(\theta_e;\Omega)$ represents the mutual information of θ_e with Ω.

Proof. We know Alice's encryption private key $\theta_e = 2(c-1)\pi - 2\pi \cdot \frac{a_0}{d} + \theta_1 + \gamma_2 + \gamma_3 + \cdots + \gamma_k$. Essentially, the private share θ_1 in θ_e, and the rotation keys $\{\gamma_2, \gamma_3, \ldots, \gamma_k\}$ are each derived independently from a uniform distribution, and since θ_1 is kept secret by Alice, the secret a_0 cannot be recovered even if all evaluators conspire [34], which provides the maximum entropy for the encryption key. The conditional entropy of the encryption key θ_e is the same as its total entropy, $H(\theta_e|\Omega) = H(\theta_e)$, and hence the mutual information $I(\theta_e;\Omega) = H(\theta_e) - H(\theta_e|\Omega) = 0$, proving the security of the encryption key θ_e in the privacy key space [40–42], the encryption key is secure. Likewise, except for honest Bob_1, the decryption key $\theta_d = 2c\pi + \gamma_1 - \theta_1$ is random for the remaining $k-1$ evaluators and attackers, since they cannot obtain any information about the decryption key. The conditional entropy of the decryption key θ_d is the same as its total entropy, $H(\theta_d|\Omega) = H(\theta_d)$, and then the mutual information $I(\theta_d;\Omega) = H(\theta_d) - H(\theta_d|\Omega) = 0$. So, the decryption key θ_d is also secure. □

(2) **The security of the ciphertext quantum state sequences during transmission**

During the encryption and evaluation computation stages, the transmitted ciphertext quantum state sequences could be subject to an intercept–resend attack by an eavesdropper, Eve, over the quantum channel. If Eve intercepts and resends the quantum states, it may alter the state of the transmitted ciphertexts, potentially leading to incorrect quantum

computations by the evaluators on incorrect ciphertext quantum states and erroneous decryption by the data owner. To defend against this, we insert l decoy particles into the transmitted ciphertext quantum state sequence. These particles are randomly placed and can take one of four possible states $\{|1\rangle, |0\rangle, |+\rangle, |-\rangle\}$. If Eve intercepts and measures a decoy particle, the probability of correctly choosing the measurement basis is $1/2$, and the probability of matching the correct basis is also $1/2$.The probability that Eve successfully intercepts the decoy particles without altering their state is $\left(\frac{l}{4(m+l)}\right)^l$ where m is the length of the ciphertext sequence and l is the number of decoy particles inserted. The probability that Eve's attack is undetected is

$$Pr_{sd} = 1 - \left(\frac{l}{4(m+l)}\right)^l. \tag{24}$$

When the number l of inserted decoy particles in the transmitted quantum state sequence is sufficiently large, the probability Pr_{sd} of detecting the intercept–resend attack approaches 1, ensuring the security of the transmitted ciphertext quantum state sequence. Thus, by inserting decoy particles with different states, we can effectively prevent intercept–resend attacks. The ciphertext quantum state sequences is secure during transmission.

(3) **The security of the plaintext quantum state sequence**

Theorem 2. *The proposed TQHE scheme is q-IND-CPA secure. For any OPT adversary $\mathcal{A}$, he cannot distinguish between the original ciphertext quantum state sequences encrypted from different original plaintext quantum state sequences.*

Proof. We prove Theorem 2 through the following indistinguishability game $Game_{\mathcal{A},\xi}$. For any adversary $\mathcal{A}$ and a security parameter κ, the proposed TQHE scheme holds that

$$\Pr[Game_{\mathcal{A},\xi}(\kappa) = 1] \leq \frac{1}{2} + negl(\kappa), \tag{25}$$

where $\Pr\left[Game_{\mathcal{A},\xi}(\kappa) = 1\right]$ is the probability that $\mathcal{A}$ wins the indistinguishability game $Game_{\mathcal{A},\xi}$ and $negl(\kappa)$ is a negligible function. In the game $Game_{\mathcal{A},\xi}$, an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is constructed. $\mathcal{A}_1$ firstly selects an input $|m_0\rangle$ according to the evaluation key ρ_{evk_i}. The challenger samples a random bit $r \in \{0, 1\}$. If $r = 1$, the input $|m_0\rangle$ is encrypted to $|c_a\rangle$ and send to $\mathcal{A}_2$; otherwise, $r = 0$, $|m_0\rangle$ is swapped out and replaced by a dummy input $|0\rangle\langle 0|$ to obtain $|c_a\rangle$. Then, the challenger sends the challenge ciphertext $|c_a\rangle$ to $\mathcal{A}_2$. In our proposed TQHE scheme, $\mathrm{Adv}_E^{\text{q-IND-CPA}}(\mathcal{A})$ is the advantage of winning the q-IND-CPA game when the adversary $\mathcal{A}$ faces the encryption algorithm, where $\mathrm{Adv}_E^{\text{q-IND-CPA}}(\mathcal{A}) \leq negl(\kappa)$. The probability that $\mathcal{A}_2$ guesses $r^{'} = r$ (wins the $Game_{\mathcal{A},\xi}$) is

$$\Pr\left[Game_{\mathcal{A},\xi}(\kappa) = 1\right] = \Pr[r^{'} = r] = \frac{1}{2} + \mathrm{Adv}_E^{\text{q-IND-CPA}}(\mathcal{A}) \leq \frac{1}{2} + negl(\kappa), \tag{26}$$

we can conclude that the adversary $\mathcal{A}$ wins the indistinguishability game with a probability no higher than $1/2$. Therefore, the proposed TQHE scheme satisfies q-IND-CPA. In other words, the plaintext quantum state sequence is secure. □

(4) **The security of the result after computations on the plaintext quantum state sequence**

Theorem 3. *The proposed TQHE scheme for an arbitrary OPT adversary $\mathcal{A}$, he cannot distinguish the final ciphertext quantum state sequences evaluated computations on different original plaintext quantum state sequences.*

Proof. We prove Theorem 3 through the following indistinguishability game $Game_{\mathcal{A},\eta}$. In the game $Game_{\mathcal{A},\eta}$, an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2$ is constructed. $\mathcal{A}_1$ firstly selects an input $|m_0\rangle$ according to the evaluation key ρ_{evk_i}. The challenger samples a random bit $r \in \{0,1\}$. If $r = 1$, the input $|m_0\rangle$ is encrypted to $|c_a\rangle$ and sent to $\mathcal{A}_2$; otherwise, $r = 0$, $|m_0\rangle$ is swapped out and replaced by a dummy input $|0\rangle\langle 0|$ to obtain $|c_a\rangle$. Then, the challenger computes the homomorphic evaluation TQHE.Eval $(\chi_i, \rho_{evk_i}, |c_a\rangle)$ to obtain the result $|c_f\rangle$. These two types of computations are represented by $\eta.real$ and $\eta.ideal$, respectively. Then, the challenger decrypts the $|c_f\rangle$ to obtain $|m_f\rangle$ which is sent to $\mathcal{A}_2$. In our proposed TQHE scheme, $\mathcal{A}_2$ guesses $r^{'} = r$ with a probability significantly lower than $1/2$. The probability that the adversary $\mathcal{A}$ wins is

$$\begin{aligned}\Pr\left[Game_{\mathcal{A},\eta}(\kappa) = 1\right] &= \frac{1}{2}\Pr[\eta.real] + \frac{1}{2}\Pr[\eta.ideal] \\ &= \Pr[r = 0]\Pr[\mathcal{A}_3 \text{ guesses } 0 | r = 0] + \Pr[r = 1]\Pr[\mathcal{A}_3 \text{ guesses } 1 | r = 1] \\ &\leq \frac{1}{2} + negl(\kappa).\end{aligned} \tag{27}$$

□

We can conclude that $\mathcal{A}$ wins the indistinguishability game with a probability no higher than $1/2$. That means the adversary $\mathcal{A}$ cannot distinguish the final ciphertext quantum state sequences by evaluating computations on different original plaintext quantum state sequences. Furthermore, the result after computations on the plaintext quantum state sequence is secure.

6. Comparisons

In this section, the approximate efficiency of the scheme will be analysed in terms of the approximate counts of the total computational complexity and time complexity of one successful execution of the TQHE scheme by the data owner and the evaluators. Comparison with the existing TQHE schemes [31,32] in terms of flexibility and efficiency shows that our proposed TQHE scheme has better flexibility and higher efficiency.

Computational complexity: In the key generation stage, the data owner computes and distributes the key information for k evaluators with complexity $O(t + n + k)$; in the encryption sharing stage, the data owner performs m encryption unitary operations and the evaluators perform km phase shift unitary operations with complexity $O(m)$; in the evaluation computation stage, k evaluators perform the combined execution of the homomorphic evaluation computation task with complexity $O(Cm)$ (where C is the total number of evaluation computation gates to be performed, a non-negligible constant); in the decryption stage, the data owner performs m decryption unitary operations with complexity $O(m)$. Hence, the computational complexity of the proposed scheme is $O(t + n + k), O(m), O(Cm), O(m)$.

Time complexity: In the single execution scheme, in the key generation phase, the data owner executes the key generation and distribution algorithm with complexity $O(t + n + k)$; in the encryption sharing stage, the data owner applies m encryption unitary operations, while the evaluators execute km phase shift unitary operations, with an overall complexity of $O(m)$; in the evaluation computation stage, k evaluators cooperate to perform all homomorphic evaluation computation tasks with complexity $O(tk)$ (t is the average computation time for each evaluator to perform the assigned evaluation task); in the decryption stage, the data owner performs m decryption unitary operations with complexity $O(m)$. Therefore, the computational complexity of the proposed scheme is $O(t + n + k)$, $O(m)$, $O(tk)$, $O(m)$.

As shown in Table 2, the total computational complexity and time complexity of the TQHE schemes of Chen et al. [31] and Liu et al. [32] are the same in the four stages of the TQHE scheme, which are $O(n^k \cdot k^3)$, $O(m)$, $O(Cm)$, $O(m)$ and $O(n^k \cdot k^3)$, $O(m)$, $O(tk)$, $O(m)$, respectively. Moreover, in our scheme, any evaluators have the ability to perform any single-qubit unitary operations from $\{X,\ Y,\ Z,\ H,\ S,\ T, U(\theta)\}$ and can execute any single-qubit gate unitary operations from $\{X,\ Y,\ Z,\ H,\ S,\ T\}$, which shows that our proposed scheme has more excellent flexibility compared to scheme [31] and scheme [32].

Table 2. Comparison with the schemes proposed by Chen et al. [31] and by Liu et al. [32].

Property	Chen et al.'s Scheme [31]	Liu et al.'s Scheme [32]	The Proposed Scheme
The scope of quantum evaluation computation of k evaluators	$\{X,Y,Z,H\}^1$ $*\{X,Y,Z,H\}^2$ $*\cdots$ $*\{X,Y,Z,H\}^{k-1}$ $*\{X,Y,Z,H,S,T\}^k$	$\{X,Y,Z,H,S,T\}^1$ $*\{X,Y,Z,H,S,T\}^2$ $*\cdots$ $*\{X,Y,Z,H,S,T\}^{k-1}$ $*\{X,Y,Z,H,S,T\}^k$	$\{X,Y,Z,H,S,T\}^1$ $*\{X,Y,Z,H,S,T\}^2$ $*\cdots$ $*\{X,Y,Z,H,S,T\}^{k-1}$ $*\{X,Y,Z,H,S,T\}^k$
The quantum computing capability possessed by evaluators	All k evaluators: $\{X,Y,Z,H,S,T,U(\theta)\}$	The first $k-1$ evaluators: $\{X,Y,Z,U(\theta)\}$ The kth evaluators: $\{X,Y,Z,H,S,T,U(\theta)\}$	All k evaluators: $\{X,Y,Z,H,S,T,U(\theta)\}$
Computational complexity	$O(n^k \cdot k^3)$,$O(m)$,$O(Cm)$, $O(m)$	$O(n^k \cdot k^3)$,$O(m)$,$O(Cm)$, $O(m)$	$O(t+n+k)$, $O(m)$, $O(Cm)$, $O(m)$
Time complexity	$O(n^k \cdot k^3)$,$O(m)$,$O(tk)$, $O(m)$	$O(n^k \cdot k^3)$,$O(m)$,$O(tk)$, $O(m)$	$O(t+n+k)$, $O(m)$, $O(tk)$, $O(m)$

"$*$" represents the connection of the quantum evaluation computation scopes of each evaluator.

7. Conclusions

In summary, we first propose a novel TQHE network scheme with a flexible number of evaluators in this paper, in which any evaluators have the ability to perform single-qubit unitary operations $\{X,Y,Z,H,S,T,U(\theta)\}$ and can execute any single-qubit gate unitary operations from the set of single-qubit gates $\{X,Y,Z,H,S,T\}$ assigned by the data owner on the ciphertext quantum state sequence. Subsequently, a specific (3, 5)-threshold QHE example is given to further show the correctness and feasibility of our scheme. In addition, the example is then simulated on the IBM quantum computing cloud platform, and the results of the experiment also verify the correctness and feasibility of the scheme. Finally, a comprehensive analysis of the security of the encryption/decryption private keys, ciphertext quantum state sequences during transmission, the plaintext quantum state sequence, and the result after computations on the plaintext quantum state sequence indicates that our proposed scheme is secure.

Author Contributions: Conceptualization, Y.T. and M.G.; methodology, Y.T. and M.G.; validation, Y.T., M.G., B.L. and K.G.; formal analysis, M.G.; writing—original draft preparation, M.G.; writing—review and editing, Y.T. and B.L.; supervision, Y.T., J.Y. and B.Q.; funding acquisition, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62472144), the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China (Grant No. ICNS202006), the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No. SKLACSS-202401) and the Sichuan Science and Technology Program (Grant No. 2024NS FSC0515).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

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

References

1. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur Comput.* **1978**, *4*, 169–180.
2. Gentry, C. Fully homomorphic encryption using ideal lattices. In Proceedings of the 41th Annual ACM symposium on Theory of Computing, New York, NY, USA, 31 May–2 June 2009; pp. 169–178. [CrossRef]
3. Goldwasser, S.; Kalai, Y.; Popa, R.A.; Vaikuntanathan, V.; Zeldovich, N. Reusable garbled circuits and succinct functional encryption. In Proceedings of the 45th Annual ACM Symposium on Theory of Computing, New York, NY, USA, 1–4 June 2013; pp. 555–564. [CrossRef]
4. Chung, K.M.; Kalai, Y.; Vadhan, S. Improved delegation of computation using fully homomorphic encryption. In Proceedings of the 30th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2010; pp. 483–501. [CrossRef]
5. Garg, S.; Gentry, C.; Halevi, S.; Raykova, M.; Sahai, A.; Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.* **2016**, *45*, 882–929. [CrossRef]
6. Aung, K.M.M.; Lee, H.T.; Tan, B.H.M.; Wang, H.X. Fully homomorphic encryption over the integers for non-binary plaintexts without the sparse subset sum problem. *Theor. Comput. Sci.* **2019**, *28*, 49–70. [CrossRef]
7. Cheon, J.H.; Coron, J.S.; Kim, J.; Lee, M.S.; Lepoint, T.; Tibouchi, M.; Yun, A. Batch fully homomorphic encryption over the integers. In Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, 26–30 May 2013; pp. 315–335. [CrossRef]
8. Nuida, K.; Kurosawa, K. (Batch) fully homomorphic encryption over integers for non-binary message spaces. In Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015; pp. 537–555. [CrossRef]
9. Benarroch, D.; Brakerski, Z.; Lepoint, T. FHE over the integers: Decomposed and batched in the post-quantum regime. In Proceedings of the 20th IACR International Workshop on Public Key Cryptography, Amsterdam, The Netherlands, 28–31 March 2017; pp. 271–301. [CrossRef]
10. Rohde, P.P.; Fitzsimons, J.F.; Gilchrist, A. Quantum walks with encrypted data. *Phys. Rev. Lett.* **2012**, *109*, 150501. [CrossRef] [PubMed]
11. Liang, M. Symmetric quantum fully homomorphic encryption with perfect security. *Quant. Inf. Proc.* **2013**, *12*, 3675–3687. [CrossRef]
12. Liang, M. Quantum fully homomorphic encryption scheme based on universal quantum circuit. *Quant. Inf. Proc.* **2015**, *14*, 2749–2759. [CrossRef]
13. Broadbent, A.; Jeffery, S. Quantum homomorphic encryption for circuits of low T-gate complexity. In Proceedings of the 35th Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; pp. 609–629. [CrossRef]
14. Dulek, Y.; Schaffner, C.; Speelman, F. Quantum homomorphic encryption for polynomial-sized circuits. In Proceedings of the 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2016; pp. 3–32. [CrossRef]
15. Ouyang, Y.; Tan, S.H.; Fitzsimons, J.F. Quantum homomorphic encryption from quantum codes. *Phys. Rev. A* **2018**, *98*, 042334. [CrossRef]
16. Tan, S.H.; Kettlewell, J.A.; Ouyang, Y.; Chen, L.; Fitzsimons, J.F. A quantum approach to homomorphic encryption. *Sci. Rep.* **2016**, *6*, 33467. [CrossRef]
17. Fisher, K.A.G.; Broadbent, A.; Shalm, L.K.; Yan, Z.; Lavoie, J.; Prevedel, R.; Jennewein, T.; Resch, K.J. Quantum computing on encrypted data. *Nat. Commun.* **2014**, *5*, 3074. [CrossRef]
18. Mahadev, U. Classical homomorphic encryption for quantum circuits. In Proceedings of the IEEE 59th Annual Symposium on Foundations of Computer Science, Paris, France, 7–9 October 2018; pp. 189–215. [CrossRef]
19. Alagic, G.; Dulek, Y.; Schaffner, C.; Speelman, F. Quantum fully homomorphic encryption with verification. In Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017; pp. 438–467. [CrossRef]
20. Tham, W.K.; Ferretti, H.; Bonsma-Fisher, K.; Brodutch, A.; Sanders, B.C.; Steinberg, A.M.; Jeffery, S. Experimental demonstration of quantum fully homomorphic encryption with application in a two-party secure protocol. *Phys. Rev. X* **2020**, *10*, 011038. [CrossRef]
21. Marshall, K.; Jacobsen, C.S.; Schäfermeier, C.; Gehring, T.; Weedbrook, C.; Andersen, U.L. Continuous-variable quantum computing on encrypted data. *Nat. Commun.* **2016**, *7*, 13795. [CrossRef]
22. Zeuner, J.; Pitsios, I.; Tan, S.H.; Sharma, A.N.; Fitzsimons, J.F.; Osellame, R.; Walther, P. Experimental quantum homomorphic encryption. *NPJ Quantum. Inform.* **2021**, *7*, 25. [CrossRef]

23. Zhou, Q.; Lu, S.; Cui, Y. Quantum search on encrypted data based on quantum homomorphic encryption. *Sci. Rep.* **2020**, *10*, 5135. [CrossRef]
24. Kimble, H.J. The quantum internet. *Nature* **2008**, *453*, 1023–1030. [CrossRef]
25. Kong, X.Q.; Li, Q.; Wu, C.H.; Yu, F.; He, J.J.; Sun, Z.Y. Multiple-server flexible blind quantum computation in networks. *Int. J. Theor. Phys.* **2016**, *55*, 3001–3007. [CrossRef]
26. Cicconetti, C.; Conti, M.; Passarella, A. Resource allocation in quantum networks for distributed quantum computing. In Proceedings of the IEEE International Conference on Smart Computing, Helsinki, Finland, 20–24 June 2022; pp. 124–132. [CrossRef]
27. Kozlowski, W.; Wehner, S. Towards large-scale quantum networks. In Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication, Dublin, Ireland, 25–27 September 2019; pp. 1–7. [CrossRef]
28. Moreno, M.G.M.; Brito, S.; Nery, R.V.; Chaves, R. Device-independent secret sharing and a stronger form of Bell nonlocality. *Phys. Rev. A* **2020**, *101*, 052339. [CrossRef]
29. De Oliveira, M.; Nape, I.; Pinnell, J.; TabeBordbar, N.; Forbes, A. Experimental high-dimensional quantum secret sharing with spin-orbit-structured photons. *Phys. Rev. A* **2020**, *101*, 042303. [CrossRef]
30. Senthoor, K.; Sarvepalli, P.K. Communication efficient quantum secret sharing. *Phys. Rev. A* **2019**, *100*, 052313. [CrossRef]
31. Chen, X.B.; Sun, Y.R.; Xu, G.; Yang, Y.X. Quantum homomorphic encryption scheme with flexible number of evaluator based on (k, n)-threshold quantum state sharing. *Inf. Sci.* **2019**, *501*, 172–181. [CrossRef]
32. Liu, J.; Li, Q.; Quan, J.Y.; Wang, C.; Shi, J.; Situ, H. Efficient quantum homomorphic encryption scheme with flexible evaluators and its simulation. *Des. Codes Cryptogr.* **2022**, *90*, 577–591. [CrossRef]
33. Cao, H.; Ma, W.P. (t, n) threshold quantum state sharing scheme based on linear equations and unitary operation. *IEEE Photonics J.* **2017**, *9*, 1–7. [CrossRef]
34. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
35. Song, X.L.; Liu, Y.B.; Xiao, M.; Deng, H.Y. A verifiable (t, n) threshold quantum state sharing scheme on IBM quantum cloud platform. *Quantum Inf. Process.* **2020**, *19*, 337. [CrossRef]
36. Fitzsimons, J.F.; Kashefi, E. Unconditionally verifiable blind quantum computation. *Phys. Rev. A* **2017**, *96*, 012303. [CrossRef]
37. Nielsen, M.A.; Chuang, I.L. Quantum computation and quantum information. *Phys. Today* **2001**, *54*, 60. [CrossRef]
38. Huang, H.L.; Zhao, Y.W.; Li, T.; Li, T.; Li, F.G.; Du, Y.T.; Fu, X.Q.; Zhang, S.; Wang, X.; Bao, W.S. Homomorphic encryption experiments on IBM's cloud quantum computing platform. *Front. Phys.* **2017**, *12*, 120305. [CrossRef]
39. Soeparno, H.; Perbangsa, A.S. Cloud quantum computing concept and development: A systematic literature review. *Procedia Comput. Sci.* **2021**, *179*, 944–954. [CrossRef]
40. Ravi, G.S.; Smith, K.N.; Gokhale, P.; Chong, F.T. Quantum Computing in the Cloud: Analyzing job and machine characteristics. In Proceedings of the IEEE International Symposium on Workload Characterization, Virtual, 7–9 November 2021; pp. 39–50. [CrossRef]
41. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
42. Corniaux, C.L.F.; Hossein, G. An entropy-based demonstration of the security of Shamir's secret sharing scheme. In Proceedings of the 2014 International Conference on Information Science, Hokkaido, Japan, 26–28 April 2014; pp. 46–48. [CrossRef]