Quantum Searchable Encryption for Cloud Data Based on Delegating Quantum Computing

Yinsong Xu[1(✉)], Wenjie Liu[1,2], Junxiu Chen[1], and Lian Tong[3]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, People's Republic of China
mugongxys@foxmail.com, wenjiel@163.com, cjxccc981@163.com
[2] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, People's Republic of China
[3] School of Information Engineering, Jiangsu Maritime Institute, Nanjing 211100, People's Republic of China
dianxin040204nv@126.com

Abstract. Based on delegating quantum computing (DQC), a DQC model that adapts to multi-qubit and composite quantum circuits is given firstly. In this model, the single client with limited quantum ability can give her encrypted data to a powerful but untrusted quantum data server and let the data server computes over the encrypted data without decryption, where the computation is a quantum circuit composed of multiple quantum gates. Then, the client generates the decryption key to decrypt the computing result according to the circuit of computation. However, this model cannot meet the situation of multi-client accessing or computing encrypted cloud data in the cloud environment. To solve this problem, we let the client outsource key generation to a trusted key server, which composes the quantum cloud center with the data server. The clients only perform X and Z operation according to the encryption or decryption key. Then, combined with Grover algorithm, a quantum searchable encryption scheme for cloud data based on delegating quantum computing is proposed in this paper. The data server mainly uses Grover algorithm to perform search computation on the encrypted data. Moreover, a concrete example of our scheme is discussed next, where the data server searches for 2 target items from 8 items of the encrypted data. Finally, security of our proposed scheme is analysed, which can protect the security of the data.

This work was supported by the National Natural Science Foundation of China under Grant 61672290, Grant 71461005, and Grand 61802002, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171458, in part by the Natural Science Foundation of Jiangsu Higher Education Institutions under Grant 19KJB520028, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 396–409, 2020.
https://doi.org/10.1007/978-3-030-48513-9_32

Keywords: Quantum searchable encryption · Delegating quantum computing · Untrusted data server · Trusted key server · Grover algorithm

1 Introduction

In recent years, cloud computing has achieved great development both in academic and industry communities as it provides economic and convenient service, which can recommend to people what they want [1,2], provide energy saving solutions [3,4], and so on. And now more and more clients are planning to upload their data onto the public clouds. However, data stored in the cloud server may suffer from malicious use by cloud service providers. Considering data privacy and security, it is a recommended practice for data owners to encrypt data before uploading onto the cloud [5,6]. Therefore, an efficient search technique for encrypted data is extremely urgent.

A popular way to search over encrypted data is searchable encryption (SE). The first searchable encryption was proposed by Song *et al.* [7]. This scheme uses stream ciphers and pseudo-random functions to implement ciphertext retrieval, but it also has a series of problems, such as low search efficiency and data privacy. Therefore, Goh [8] built a index structure based on the Bloom filter to achieve fast retrieval of ciphertext data. However, the Bloom filter itself has a certain error rate, and the result returned by the cloud server to the data user may not be accurate. Besides, Curtmola *et al.* [9] and Boneh *et al.* [10] use the idea of "keyword-file" to construct a symmetric searchable encryption scheme and a public key search able encryption scheme, respectively. Both schemes have significant improvements in safety and efficiency. Nowadays, many researchers have tried to use kNN algorithm [11], user interest model [12], blockchain technology [13], and so on, to improve the search efficiency and data privacy.

On the other hand, in the field of quantum computation, to protect the privacy of client's data, many researchers have proposed a novel model of quantum computation: blind quantum computation (BQC), where the client with limited quantum resources can perform quantum computation by delegating the computation to an untrusted quantum server, and the privacy of the client can still be guaranteed. BQC can be generally divided into two categories: one is the measurement-based blind quantum computation (MBQC), and the other is the circuit-based blind quantum computation (CBQC). In MBQC, measurement is the main driving force of computation, which follows the principle of "entangle-measure-correct", and a certain number of quantum qubits are entangled to form a standard graph state [14,15]. Different from MBQC, CBQC is based on the quantum circuit that is composed of many kinds of quantum gates [16–19]. Among them, Fisher [18] and Broadbent [19] firstly proposed a representative CBQC model: delegating quantum computation (DQC). In their protocols, an untrusted server can perform arbitrary quantum computations on encrypted quantum bits (qubits) without learning any information about the inputs, where the quantum computations are implemented by a universal set of

quantum gates (X, Z, H, S, T, $CNOT$). And then the client can easily decrypt the results of the computation with the decryption key. However, since Fisher and Broadbent only considered two parties, Kashefi *et al.* [20] proposes a multi-party delegated quantum computing protocol later. But, this protocol is actually under the measurement-based quantum computing framework, which belongs to MBQC and is not DQC.

In order to implement multiclient DQC, i.e., different clients can store or search their data in the quantum cloud center, we propose a quantum searchable encryption scheme for cloud data based on delegating quantum computing. Our scheme has five components: encryption key generation, encryption, search, decryption key generation and decryption. Clients firstly use X and Z gates to encrypt their data with the encryption keys, where the encryption keys are generated by the key server, and then send the encrypted data to the data server. The data server performs search computation (i.e., Grover algorithm) on the encrypted data if other clients need, where the search computation are implemented by a universal set of quantum gates (X, Z, H, S, T, $CNOT$). During the search computation, the data server assists the key server to generate decryption keys. Finally, the clients who need the search result from the data server, also use X and Z gates to decrypt the encrypted search result with the decryption keys from the key server.

The rest of the paper is organized as follows. Section 2 provides some preliminary knowledge about quantum computation and how to perform quantum computing on encrypted qubit. Then, a quantum searchable encryption scheme for cloud data based on delegating quantum computing is proposed in Sect. 5. Moreover, we give a concrete example that use Grover algorithm to search on encrypted 2-qubit state in Sect. 4. And security analysis is discussed in Sect. 5. Finally, Sect. 6 gives conclusion of this paper.

2 Preliminaries

2.1 Quantum Computation

As we know, the bit is the fundamental concept of classical information, and has a state, either 0 or 1. Similar to the classical bit, the quantum bit (called qubit) [21] is the basic unit of quantum information and has two possible states $|0\rangle$ and $|1\rangle$, which is often referred to as quantum superposition state,

$$|\varphi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where α, β are complex numbers, and $|\alpha|^2 + |\beta|^2 = 1$. $|0\rangle$ and $|1\rangle$ can be represented by vectors,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2}$$

Then, $|\varphi\rangle$ can be expressed in vector form $|\varphi\rangle = \binom{\alpha}{\beta}$.

Analogous to the way that a classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information. Single-qubit gates, such as *Pauli-X*, *Pauli-Z*, *H* (*Hadamard*), *S* and *T* are the simplest form of quantum gates, and they can be described as 2×2 unitary matrices as below,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{3}$$

Multi-qubit gates are also the important units in a quantum circuit. The prototypical multi-qubit quantum logic gate is *controlled-NOT* (i.e., *CNOT*) gate (shown in Fig. 1), which has two input qubits, known as the control qubit and the target qubit, respectively. If the control qubit is set to 0, then the target qubit is left alone. If the control qubit is set to 1, then the target qubit is flipped.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

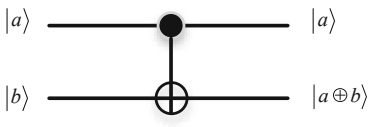

Fig. 1. Matrix representation and quantum circuit of *CNOT* gate.

2.2 Delegating Quantum Computing

This delegating quantum computing (DQC) scheme was firstly proposed by Fisher [18] and Broadbent [19]. It (see Fig. 2a) starts with a client who has quantum information that needs to be sent to a remote server for processing. The client first encrypts one input qubit $|\psi\rangle$ and sends it to a quantum server, who performs a computation U on the encrypted qubit. The server returns the state which the client decrypts to get $U|\psi\rangle$.

In the scheme, to encrypt a qubit $|\psi\rangle$, a client applies a combination of Pauli X and Z operations to get a encrypted qubit $X^a Z^b |\psi\rangle$, where $a, b \in \{0, 1\}$ (as well as $c, d \in \{0, 1\}$ for the *CNOT* gate in Fig. 2f). Then, the server perform quantum computing U, which is composed of unitary operations from the Clifford group $\{X, Z, H, S, CNOT\}$ and one additional non-Clifford gate, T gate. As shown in Fig. 2b-f, when $U \in \{X, Z, H, S, CNOT\}$, clifford gates do not require any additional resources, and decryption is straightforward. However, when $U = T$ (see Fig. 2g), the server requires the client to send an auxiliary qubit $Z^d P^y |+\rangle$, where $y, d \in \{0, 1\}$. to control a *CNOT* gate with the encrypted qubit. The server measures the encrypted qubit and outcome $c \in \{0, 1\}$ is returned to the client, which is used in decryption. The client sends a single classical bit, $x = a \oplus y$, to control a S gate on the auxiliary qubit, which is returned to the client as $X^{a''} Z^{b''} R |\psi\rangle$, where $a'' = a \oplus c$ and $b'' = a(c \oplus y \oplus 1) \oplus b \oplus d \oplus y$.

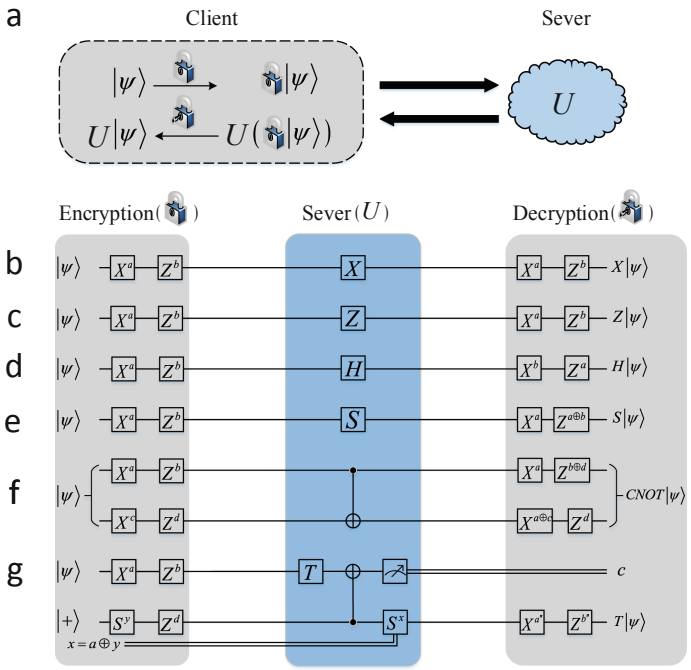

Fig. 2. Protocol for delegating quantum computing

3 Quantum Searchable Encryption for Cloud Data Based on Delegating Quantum Computing

In this section, we firstly give a simple multi-qubit DQC model, which only contains two parties: client and single server.

3.1 A Multi-qubit DQC Model

Suppose the client *Alice* wants the single server *Bob* to search over her encrypted data. The basic process of this model is given as below, and the frequently used variables and notations are listed in Table 1.

1. *Alice* should encrypt the data with Pauli operators $\{X, Z\}$ depending on a classical encryption key $ek = (x_0, z_0)$, and send it to *Bob*.
2. When *Bob* performs search computation on the encrypted data, *Alice* computes the decryption key for the encrypted data, where the search computation is generally composed of a set of unitary gates $\{X, Z, H, S, T, CNOT\}$ in the quantum circuit. For the sake of clarity, the decryption key generation rules for arbitrary unitary transforms $\{X, Z, H, S, T, CNOT\}$ in the circuit are combed in Algorithm 1.

3. *Alice* also only needs to decrypt the search result with X and Z gates depending on the decryption key.

Table 1. Explanations for frequently used variables and notations.

Variables and notations	Explanations
$\mathbb{N}$, $\mathbb{N}^+$	$\mathbb{N} = \{0, 1, 2, \cdots\}$ is a set of non-negative integers, and $\mathbb{N}^+ = \{1, 2, 3, \cdots\}$ is a set of positive integers
M, m, $n(m, n \in \mathbb{N}^+)$	$M = 2^m$ is the number of items to be searched, and each item $data(j)$ contains data of n bits, where $j \in \{0, 1, 2, \cdots, M-1\}$
ek, dk, sk	ek and dk are $2n$-bit encryption and $2n$-bit decryption keys, respectively, for Alice's data. dk is encrypted with sk
$x_r(k)$, $z_r(k)(r \in \mathbb{N}^+, k \in \mathbb{N})$	(x_r, z_r) is the $2n$-bit intermediate key of the r^{th} round in Algorithm 1; $ek = (x_0, z_0)$; $x_r(k)$ is the k^{th} bit of x_r and $z_r(k)$ is the k^{th} bit of z_r
ζ	$\zeta = \{I, X, Z, H, S, T, CNOT\}$
X_i, Z_i, H_i, S_i, T_i, $CNOT_{i,l}$	X_i, Z_i, H_i, S_i, or T_i denotes applying a X, Z, H, S or T gate on the i^{th} qubit of the input state and letting the other qubits unchanged; $CNOT_{i,l}$ denotes performing a $CNOT$ gate on the i^{th} and l^{th} qubits of the input, which act as the control and target qubits, respectively

Algorithm 1. (decryption key generation rules for arbitrary unitary transform in ζ). Suppose $|\psi\rangle$ is an n-qubit quantum state, U is an n-qubit unitary transform composed of gates from the universal gate set ζ, and G represents any one gate of ζ. Let $ek = (x_0, z_0)$ and $U_0 = \otimes_{k=1}^{n} I$, the encrypted quantum state $(\otimes_{k=1}^{n} X^{x_0(k)} Z^{z_0(k)}) |\psi\rangle$ is equivalent to $U_0(\otimes_{k=1}^{n} X^{x_0(k)} Z^{z_0(k)}) |\psi\rangle$; the updated decryption key $dk_{r+1} = (x_{r+1}, z_{r+1})$ for U_{r+1} and ek satisfying Eq. 4

$$G \otimes U_r(\otimes_{k=1}^{n} X^{x_r(k)} Z^{z_r(k)}) |\psi\rangle = (\otimes_{k=1}^{n} X^{x_{r+1}(k)} Z^{z_{r+1}(k)}) U_{r+1} |\psi\rangle, \quad (4)$$

where $U_{r+1} = G \otimes U_r$, is calculated as follows:

- If $G = I$, X_i, or Z_i, then
  $dk_{r+1} = dk_r$.
- If $G = H_i$, then
  $(x_{r+1}(i), z_{r+1}(i)) = (z_r(i), x_r(i))$,
  $(x_{r+1}(k), z_{r+1}(k)) = (x_r(k), z_r(k))(k \neq i)$.
- If $G = S_i$, then
  $(x_{r+1}(i), z_{r+1}(i)) = (x_r(i), x_r(i) \oplus z_r(i))$,
  $(x_{r+1}(k), z_{r+1}(k)) = (x_r(k), z_r(k))(k \neq i)$.
- If $G = CNOT_{i,l}$, then
  $(x_{r+1}(i), z_{r+1}(i)) = (x_r(i), z_r(i) \oplus z_r(l))$,
  $(x_{r+1}(l), z_{r+1}(l)) = (x_r(i) \oplus x_r(l), z_r(l))$,
  $(x_{r+}(k), z_{r+1}(k)) = (x_r(k), z_r(k))(k \neq i)$.

– If $G = T_i$ (suppose the secret bits *Alice* chooses for this T gate are y and d, and the related one-bit measurement result from *Bob* is c, which is shown in Fig. 2g), then
$(x_{r+1}(i), z_{r+1}(i)) = (x_r(i) \oplus c, x_r(i) \cdot (c \oplus y \oplus 1) \oplus z_r(i) \oplus d \oplus y)$,
$(x_{r+}(k), z_{r+1}(k)) = (x_r(k), z_r(k))(k \neq i)$.

3.2 Outsourcing Key Generation to a Trusted Key Server in the Cloud Environment

As mentioned above, we can see that this DQC model consumes a large amount of computing and communication resources on clients. Let us give a concrete example first. Suppose *Alice* sends the encrypted superposition state to *Bob* and *Bob* use Grover algorithm to search out result state which *Alice* needs. Since Grover's algorithm is composed of a series of unitary transforms, it can be applied directly on an encrypted superposition state and obtained the encrypted search result by the use of DQC. It is known that the Grover's search is made up of a sequence of repeated Grover iterations, and each iteration contains an oracle that has the ability to mark items satisfying a specific search condition. For NP problems, solutions can be recognized in polynomial time; this means each Grover iteration can be constructed with polynomial elementary gates. Suppose the search space has $M = 2^m$, then, there may be $O(\sqrt{M} \cdot ploy(m))$ T gates (when each Grover iteration contains polynomial T gates). So, *Alice* needs to interact with *Bob* frequently to update the decryption key. This will put a huge amount of computing and communication pressure on *Alice*.

Besides, clients only search over their own encrypted data in this model, which is not beneficial to data sharing. To solve these problems, outsourcing key generation to a trusted cloud key server is a good solution. There are rich computing and communication resources in the cloud environment. Moreover, it is also suitable for data sharing and key management. That is, we divide the client into two parties: a thin client (*Alice*) and a trusted cloud key server (*Charlie*). The requirements and constraints on *Charlie* are given in Constraint 1.

Constraint 1. (Requirements and constraints on the key server). The key server *Charlie* obeys the following two constraints:

1. *Charlie* has the ability of performing key update rules and prepares four different states of qubits:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, |+_y\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}, |-_y\rangle = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}}, \quad (5)$$

   which can serve as auxiliary qubits for T gates in the circuit of search as well as the encodings of keys by quantum key distribution (QKD).
2. *Charlie* honestly negotiates with clients about the encryption key, performs decryption key generation rules with *Bob*, then, sends the encrypted decryption key to clients who need it. The key transforming also relies on quantum key distribution.

Thus, our scheme runs among clients ($Alice_1$, $Alice_2$, $\cdots$, $Alice_n$), the key server ($Charlie$) and the data server (Bob) as illustrated in Fig. 3. These clients should firstly negotiate with the key center about the encryption key which is used to encrypt their data, and then send the encrypted data to Bob. Bob can perform search computation on the encrypted data as long as other clients need. Once Bob finishes search, $Charlie$ should generate the decryption key for the encrypted data synchronously. Finally, the clients can decrypt the search result to get the plain data with the decryption key from $Charlie$.

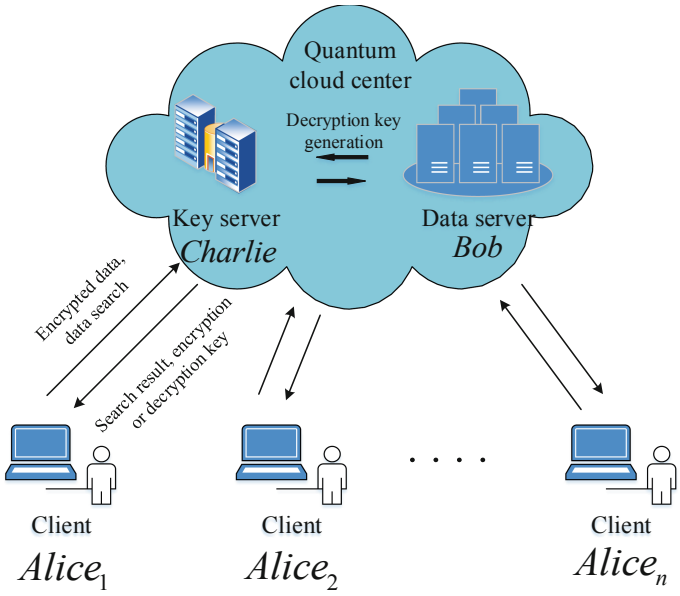

Fig. 3. The situation of quantum searchable encryption for cloud data

3.3 Quantum Searchable Encryption for Cloud Data Based on Delegating Quantum Computing

For the sake of simplicity, we take four parties (the data owner $Alice_1$, the data searcher $Alice_2$, the data server Bob and the key server $Charlie$) as an example to describe our scheme. The specific process of our scheme is as follows and shown in Fig. 5.

1. $Alice_1$ sends a number n (the length of her encrypted state) to $Charlie$.
2. $Charlie$ sends a string of $2n$ random binary bits back to $Alice_1$ by the BB84 protocol [22], where $|+\rangle$, $|+_y\rangle$ stands for 0, and $|-\rangle$, $|-_y\rangle$ stands for 1. The $2n$ bits of the binary string act as ek.

3. $Alice_1$ encrypts her superposition state $|\psi\rangle = \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |j, data(j)\rangle$ with $ek = (x_0, z_0)$ and sends encrypted state $E_{ek}|\psi\rangle$ (shown in Eq. 6) to Bob, where the item index j within $|\psi\rangle$ is not encrypted.

$$E_{ek}|\psi\rangle = \frac{1}{\sqrt{M}}(I^{\otimes m} \otimes (\otimes_{k=1}^{n} X^{x_0(k)} Z^{z_0(k)})) \sum_{j=0}^{M-1} |j, data(j)\rangle \tag{6}$$

4. $Alice_2$ wants Bob to search on $E_{ek}|\psi\rangle$, and $Charlie$ generates the decryption key synchronously. The search computation can be composed of Grover algorithm, which is illustrated in Fig. 4. For a search space of $N = 2^n$ elements and one solution, we need only apply the search oracle $O(\sqrt{N})$ times to obtain a solution. And the decryption key generation rules for arbitrary unitary transform in the circuit of search computation is listed in Algorithm 1 as below. During the search, once a T gate appears, Bob asks $Charlie$ to send an auxiliary qubit from $\{|+\rangle, |+_y\rangle, |-\rangle, |-_y\rangle\}$ along with a related key bit w (i.e. x in Fig. 2g) to him and gives $Charlie$ a measurement result (i.e. c in Fig. 2g).
5. When the search is completed, Bob sends the search result state $E_{dk}(Search(|\psi\rangle))$ to $Alice_2$.
6. $Charlie$ sends the encrypted decryption key $sk(dk)$ to $Alice_2$ by QKD, where $sk(dk)$ can only be decrypted by $Alice_2$.
7. $Alice_2$ decrypts $sk(dk)$ to get dk, and then uses dk to decrypt the state $E_{dk}(Search(|\psi\rangle))$ to get the search result $Search(|\psi\rangle)$ (i.e., $Alice_2$ performs X^{x_r} and Z^{z_r} gates on $Search(E_{ek}|\psi\rangle)$, where $dk = (x_r, z_r)$ and r represents the number of times that Algorithm 1 is executed.).

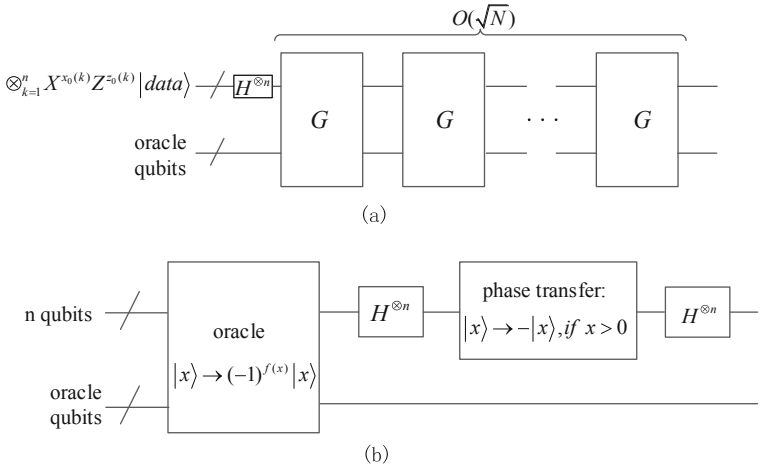

Fig. 4. Schematic circuit for Grover algorithm. (b) is the schematic circuit for G in (a).

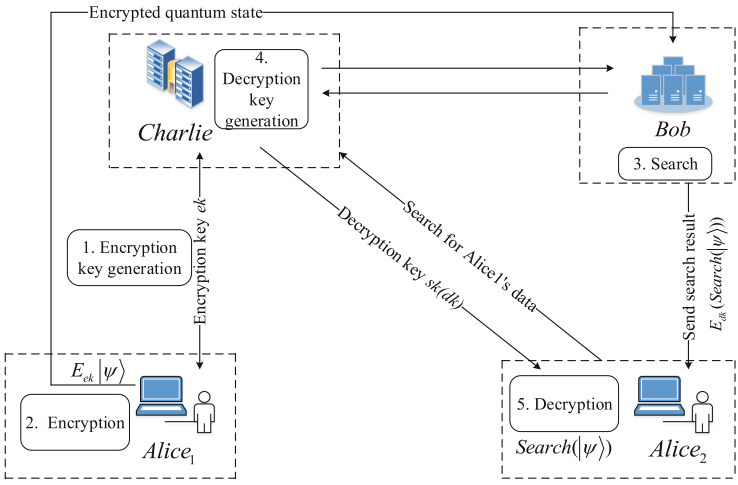

Fig. 5. The process of our scheme

4 An Example of Two-Qubit Quantum Search

Suppose $Alice_1$ has a set 000, 001, 010, 011, 100, 101, 110, 111 and $Alice_2$ wants to find the items of 001 and 011 from this set. *Bob* uses Grover algorithm to find their items, and the circuit of search computation is shown in Fig. 6. Although there are some $T^\dagger$ gates in this circuit, the decryption key update rule is as same as T gate, and the S gates in the Fig. 2 g are replaced with $S^\dagger$ gates. The example proceeds in seven steps provided below.

1. $Alice_1$ sends a number 3 to *Charlie*.
2. *Charlie* sends a string of 6 random binary bits back to $Alice_1$ by BB84 protocol, where $|+\rangle$, $|+_y\rangle$ stands for 0, and $|-\rangle$, $|-_y\rangle$ stands for 1. The 6 bits of the binary string act as $ek = (x_0, z_0)$.
3. $Alice_1$ encrypts her superposition state $|\psi\rangle = |+\rangle_1|+\rangle_2|+\rangle_3|-\rangle_4$ with X and Z gates, and sends encrypted state $E_{ek}|\psi\rangle = (\otimes_{k=1}^{4} X^{x_0(k)} Z^{z_0(k)})(|+\rangle_1|+\rangle_2 |+\rangle_3|-\rangle_4)$ (The fourth qubit does not need to be encrypted, i.e., $x_0(4) = 0$, $z_0(4) = 0$.)
4. $Alice_2$ wants *Bob* to search on $E_{ek}|\psi\rangle$, and *Charlie* compute the decryption key synchronously. During the search, the circuit in Fig. 6 has seven $T^\dagger$ and T gates. *Charlie* needs to randomly generates 14-bit (y_i, d_i) $(y_i, d_i \in \{0, 1\}; 1 \leqslant i \leqslant 7)$ to control S^y (or $S^{\dagger y}$) and Z^d (see in Fig. 2g), which can determine each state of 7 auxiliary qubits from $\{|+\rangle, |+_y\rangle, |-\rangle, |-_y\rangle\}$. *Charlie* sends these 7 auxiliary qubits and 7 related bits $w_i (1 \leqslant i \leqslant 7)$ to *Bob*. For other Clifford gates, *Charlie* performs the same operation as Algorithm 1.
5. When the search is completed, *Bob* sends the search result state $E_{dk}(Search(|\psi\rangle))$ to $Alice_2$.
6. *Charlie* sends the encrypted decryption key $sk(dk)$ to $Alice_2$ by QKD, where $sk(dk)$ can only be decrypted by $Alice_2$.

7. $Alice_2$ uses sk to decrypt the encrypted decryption key $sk(dk)$ and use dk to decrypt the state $E_{dk}(Search(|\psi\rangle))$ to get the search result $Search(|\psi\rangle)$ (i.e., $Alice_2$ performs $X^{x_{26}}$ and $Z^{z_{26}}$ gates on $Search(E_{ek}\,|\psi\rangle)$. The circuit of search computation in Fig. 6 has 26 gates. Therefore, the number of executing Algorithm 1 is 26 and $dk = (x_{26}, z_{26})$.).

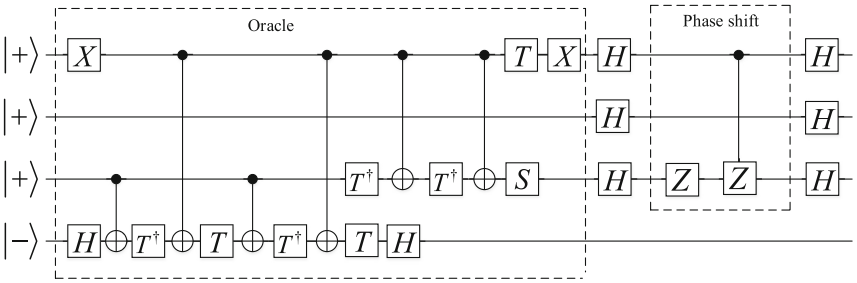

Fig. 6. The circuit of Grover algorithm to search $|001\rangle$ and $|011\rangle$ from $|+\rangle\,|+\rangle\,|+\rangle$.

5 Security Analysis

Suppose Bob is dishonest and wants to get data information from $E_{ek}\,|\psi\rangle$. He firstly needs to ek about the encrypted data, when he gets $E_{ek}\,|\psi\rangle$ sent by $Alice_1$. Since $Alice_1$ only sends the encrypted data to Bob, there is no other information interaction between $Alice_1$ and Bob, Bob cannot get any information about ek from $Alice_1$. Except $Alice_1$, only $Charlie$ has information about ek. Especially, there is some information about ek in an auxiliary qubit (i.e., one of $\{|+\rangle, |+_y\rangle, |-\rangle, |-_y\rangle\}$) and a related key bit w (i.e., x in Fig. 2g) when $Charlie$ sends them to Bob. Since $w = x_0 \oplus y$, Bob only needs to know the value of y. However, he is unable to determine the value of y when he uses $\{|+\rangle, |-\rangle\}$ or $\{|+_y\rangle, |-_y\rangle\}$ measurement base to measure this auxiliary qubit. So Bob cannot get any information about the encrypted data.

Suppose an eavesdropper $Dave$ attempts to decrypt the encrypted data by eavesdropping on the key transforming (including ek and dk). Since the key transforming relies on BB84 protocol, both parties in the communication can detect the presence of the eavesdropper. Therefore, the security of the encrypted can be guaranteed.

As analysed in above, our scheme can protect the privacy of the encrypted data (Fig. 7).

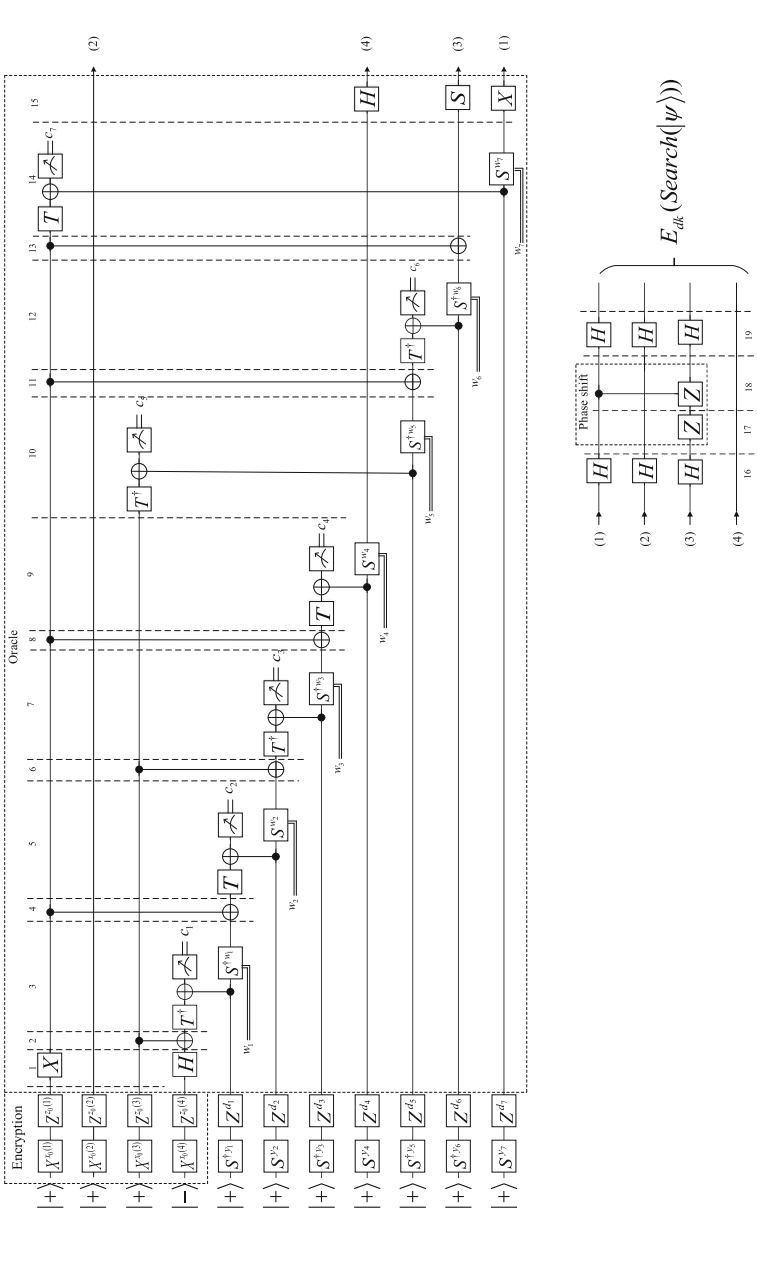

Fig. 7. The quantum search on $E_{ek}\,|\psi\rangle$. In the circuit, w_i $(1 \leqslant i \leqslant 7)$ represents that the intermediate key (corresponding to this qubit and X gate) XOR y_i. For example, when performing 5^{th} level in the circuit, five gates have been performed before, so the intermediate key for the fourth qubit is $(x_5(4), z_5(4))$. Then, $w_2 = x_5(4) \oplus y_2$

6 Conclusion

In this paper, we propose a quantum searchable encryption scheme for cloud data based on delegating quantum computing. In our scheme, the quantum cloud center, composed of the key server and the data server, can provide storage and search services for key management and encrypted data. The clients only need limited quantum ability to encrypt or decrypt the data. And the decrypted search result is equivalent to the original data, which achieves the purpose of searchable encryption. Moreover, we give an example of our scheme to verify the feasibility of our scheme. Besides, the security of our scheme is analysed in detail, which can protect the privacy of the data. Furthermore, certification of the client's legality will be our next work.

References

1. Cao, Y., Kaiwartya, O., Zhuang, Y., Ahmad, N., Sun, Y., Lloret, J.: A decentralized deadline-driven electric vehicle charging recommendation. IEEE Syst. J. **13**(3), 3410–3421 (2019)
2. Qi, L., et al.: Finding all you need: web APIs recommendation in web of things through keywords search. IEEE Trans. Comput. Soc. Syst. (2019). https://doi.org/10.1109/TCSS.2019.2906925
3. Qie, X., Jin, S., Yue, W.: An energy-efficient strategy for virtual machine allocation over cloud data centers. J. Netw. Syst. Manage. **27**(4), 860–882 (2019). https://doi.org/10.1007/s10922-019-09489-w
4. Qi, L., Chen, Y., Yuan, Y., Fu, S., Zhang, X., Xu, X.: A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web **23**(2), 1275–1297 (2019). https://doi.org/10.1007/s11280-019-00684-y
5. Bösch, C., Hartel, P., Jonker, W., Peter, A.: A survey of provably secure searchable encryption. ACM Comput. Surv. **47**(2), 1801–1851 (2014)
6. Xu, X., Liu, Q., Zhang, X., Zhang, J., Qi, L., Dou, W.: A blockchain-powered crowdsourcing method with privacy preservation in mobile environment. IEEE Trans. Comput. Soc. Syst. (2019). https://doi.org/10.1109/TCSS.2019.2909137
7. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE Symposium on Security and Privacy, S&P 2000, Berkeley, CA, USA, pp. 44–55 (2000)
8. Goh, E.: Secure indexes. IACR Cryptol. ePrint Arch. **2003**, 216 (2003)
9. Curtmola, R., Garay, J., Kamara, S. Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: Proceedings of 13th ACM Conference on Computer and Communications Security, USA, pp. 79–88 (2006)
10. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30
11. Xia, Z., Wang, X., Zhang, L., Qin, Z., Sun, X., Ren, K.: A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(11), 2594–2608 (2016)

12. Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F.: Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans. Parallel Distrib. Syst. **27**(9), 2546–2559 (2016)
13. Chen, L., Lee, W.-K., Chang, C.-C., Choo, K.-K.R., Zhang, N.: Blockchain based searchable encryption for electronic health record sharing. Future Gener. Comput. Syst. **95**, 420–429 (2019)
14. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: 50th Annual IEEE Symposium on Foundations of Computer Science, Atlanta, GA, pp. 517–526 (2009)
15. Kong, X., Li, Q., Wu, C., Yu, F., He, J., Sun, Z.: Multiple-server flexible blind quantum computation in networks. Int. J. Theor. Phys. **55**(6), 3001–3007 (2016)
16. Arrighi, P., Salvail, L.: Blind quantum computation. Int. J. Quantum Inf. **4**(5), 883–898 (2006)
17. Tan, X., Zhou, X.: Universal half-blind quantum computation. Ann. Telecommun. **72**(9), 589–595 (2017)
18. Fisher, K.A.G., et al.: Quantum computing on encrypted data. Nat. Commun. **5**, 3074 (2014)
19. Broadbent, A.: Delegating private quantum computations. Can. J. Phys. **93**(9), 941–946 (2015)
20. Kashefi, E., Pappa, A.: Multiparty delegated quantum computing. Cryptography **1**(2), 12 (2017)
21. Nielsen, M.A., Chuang, I.: Quantum Computation and Quantum Information, 10th edn. Cambridge University Press, New York (2002)
22. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: International Conference on Computers, Systems & Signal Processing, Bangalore, India, pp. 175–179. Springer (1984)