International Journal of Security and Its Applications
Vol.9, No.9 (2015), pp.253-264
http://dx.doi.org/10.14257/ijsia.2015.9.9.22

Quantum Public-key Cryptosystem without Quantum Channels between Any Two Users using Non-orthogonal States

Xiaoyu Li and Yuwen Chen

School of Information Engineering, Zhengzhou University
Iexyli@zzu.edu.cn, 646809863@qq.com

Abstract

A quantum public-key cryptosystem without quantum channels between any two users using non-orthogonal states is provided in this paper. Every user keeps a set of quantum particles in non-orthogonal states in a key management center (KMC) as the public key while he or she keeps the states of them as the private key. By the help of KMC users can accomplish secret communication and message authentication. The laws of quantum physics guarantee the unconditional security of this cryptosystem. No entangled states or complex quantum operations are needed. On the other hand there are no quantum channels needed to connecting any two users. So the public-key cryptosystem is easier to carry out and more robust in practice.

Keywords: *quantum public-key cryptosytem, quantum cryptography, non-orthogonal states, indistinguishability, message authentication*

1. Introduction

Quantum cryptography is the integration of quantum physics and cryptography. Unlike classical cryptography based on the computation complexity, quantum cryptography applies the special properties of quantum system to achieve secure communications. The laws of quantum physics guarantee that quantum cryptographic protocols can be unconditionally secure. In 1984 Bennett and Brassard proposed the first quantum key distribution protocol (so called BB84 protocol) [1], which is the beginning of quantum cryptography. After that people have developed many kinds of QKD protocols, such as the EPR protocol [2], B92 protocol [3], Lo-Chau's protocol [4], and so on [5-13]. Experiments of OKD have also been accomplished successfully for many times. In 1992 Bennett et al first realized BB84 protocol in laboratory [14]. Now QKD protocols in optical fiber have been carried out beyond 150 km [15] while QKD protocols in free space have been carried out over a distance of 1 km [16].

All the QKD protocols mentioned above belong to symmetrical key protocols in which people use the same key to encrypt the plain text and decrypt the cipher text. But symmetrical key cryptosystems are confronted with a difficult problem: how to distribute and manage keys if many users want to communicate with each other? If there are N users, every user needs to share a key with the each one of other N-1 users. So N(N-1)/2 distribution processes must be accomplished before the cryptosystem begin to work. It's a heavy work when N is a large number. Moreover in practice maybe the users don't trust each other at all, which means that they can't establish shared keys by implementing key distribution. In classical cryptography public-key cryptosystem (or public-key algorithm) is a solution to overcome such difficulties, for example, RSA algorithm [17] and so on. In a public-key cryptosystem every user has a (public key, private key) pair. The public key is used to encrypt the plain text while the private key is used to decrypt the corresponding cipher text. But the two keys are independent from each other, that is to say, holding a key is of no help to finding the other key. Every user's public key is kept by a key manage center and open to every one. But every user must keep his or her private key absolutely

ISSN: 1738-9976 IJSIA

secret. When a user Bob wants to send secret information to another user Alice, he asks KMC for Alice's public key and uses it to encrypt the plain text. Only Alice can decrypt the cipher text by her private key. So to achieve secret communications a user needs only share his or her public key with the credible KMC, which reduce the key distribution and key management greatly. Classical public-key cryptosystems have been widely applied in modern society, for example, commercial affairs, military affairs, network communications et al. But Peter Shor proved that RSA algorithm can be cracked in polynomial time on future quantum computer in 1994 [18], that is to say, the classical public-key cryptosystems based on RSA algorithm will lose effectiveness on a quantum computer. So do most classical pubic-key cryptosystems based on several other classical public-key algorithms. Quantum public-key technology may be the best solution. In 2001 Gottesman presented a quantum one-way function to design quantum message authentication protocol [19] which is instructive to develop a public-key system. Nikolopoulos presented the first unconditionally secure quantum public-key protocol in 2008 [20]. It's based on the single-particle rotation of unknown quantum states. Since then several public-key protocols have been provided [21-29].

Until now users need to exchange qubits to complete the communication process in most of current quantum public-key cryptosystems. So it's necessary to maintain a quantum channel between any two users. But it may often be very difficult even impossible in a large network including many users separated in space. In this paper we provide a quantum public-key cryptosystem without quantum channels between any two users using non-orthogonal states. It is based on the indistinguishability of non-orthogonal quantum states. With the help of the key management center, two users can communicate with each other securely. Moreover message authentication can be fulfilled naturally by the public-key cryptosystem. There are no entangled states or complex quantum operations needed. Moreover no quantum channels are needed between any two users. So it's easier to carry out and more robust in practice.

2. Basic Idea

In quantum information science the basic unit of quantum information is called a qubit which can be represented by a two-state quantum system. As known the state space of a qubit is a two-dimension Hilbert space. There are four possible states of a qubit in its state space

$$|0>, \quad |1>, \quad |+>, \quad |->. \tag{1}$$

in which

$$|+>=\frac{1}{\sqrt{2}}(|0>+|1>)$$

$$|->=\frac{1}{\sqrt{2}}(|0>-|1>). \tag{2}$$

Obviously the four states aren't orthogonal to each other. It's known that |0> and |1> form a complete orthogonal basic vector set

$$B_{01}=\{|0>, |1>\} \tag{3}$$

while |+> and |-> form another complete orthogonal basic vector set

$$B_{+-}=\{|+>, |->\}. \tag{4}$$

People can measure a qubit in B_{01} or B_{+-} as required. Quantum mechanics demands that non-orthogonal quantum states can't be discriminated from each other with certainty, that is to say, it's impossible to determine one state of the four states |0>, |1>, |+>, |-> with certainty.

Let's consider a public-key cryptosystem. It includes a key management center (KMC) and N users. A user, for example, Alice creates an n-qubit sequence denoted as Q

$$Q = (q_1 q_2 ... q_n). \quad (5)$$

in which every qubit is in one of the four states |0>, |1>, |+>, |-> at random. Q is just Alice's public key. At the same time Alice denotes the states of these n qubits' as a state sequence

$$\varphi = (|\varphi_1>|\varphi_2>...|\varphi_n>). \quad (6)$$

which is just Alice's private key. Then Alice gives her public key to KMC in which it is open to any user while she keeps her private key absolutely secret so that no one except herself can get it. All users agree to the Coding Rule.

Coding Rule:

$$|0>\rightarrow 0, |1>\rightarrow 1, |+>\rightarrow 0, |->\rightarrow 1 \quad (7)$$

Now let's assume that another user, such as Bob, wants to send a secret message denoted as an n-bit string *P* to Alice. Obviously *P* is called the plain text which Bob send to Alice. To encrypt the plain text into the cipher text, Bob asks KMC for Alice's public key Q. After getting Q he asks Alice for the correct base to measure the qubits through a public classical channel. Then to each qubit q_i in Q, Alice find its state φ_i from φ and determines the basis to measure it according to the following Key Rule.

Key Rule: If the state φ_i of the qubit q_i is |0> or |1>, the measurement basis is B_{01}; if the state φ_i of the qubit q_i is |+> or |->, the measurement basis B_{+-}.

Finally Alice gets a sequence of basis $B = (B_1 B_2 ... B_n)$ in which $B_1, B_2, ..., B_n \in \{B_{01}, B_{+-}\}$. Then Alice sends Bob *B* through the classical channel. When Bob receives *B*, he measures Q according to *B*. The measurement results can be described as the following table.

Table 1. Measurement Result

State	Bob's basis	Bob's measurement result
|0>	B_{01}	|0>
|1>	B_{01}	|1>
|+>	B_{+-}	|+>
|->	B_{+-}	|->

Obviously after finishing the measurements Bob will get φ at last. Moreover the state of any qubit hasn't changed at all. Next Bob records his measurement results according to the Coding Rule. So he gets an n-bit string *K*. Now if Bob wants to send Alice an n-bit string *P*, he performs an XOR operation on *P* and *K* to get a new string *PS* in which

$$PS = P \oplus K. \quad (8)$$

P' is just the cipher text. Then Bob sends *P*' to Alice through the classical channel. When Alice receives *P*', she first produces an n-bit string *K*' form the private key φ which she holds according to the Coding Rule. It's easy to find that *K*'=*K*. Then Alice performs an XOR operation on *K*' and *PS* to get a new string *P*'. Taking into account the property of XOR operation, we have

$$P' = PS \oplus K' = P \oplus K \oplus K' = P. \quad (9)$$

So Alice gets *P* at last. It is just the secret message Bob wants to send her. In section 4 we will prove that by a well-designed scheme no one except Alice and Bob can get the plain

text. So Alice and Bob accomplish a secure secret communication process. Moreover users can achieve message authentication by the help of KMC which we will show it in section 3.

There is still a problem unsolved. Alice's public key is an n-qubit sequence. After a communication process, it has been consumed by Bob, that is to say, one public key can be used for only one time. If many users want to send secret information to Alice or a user wants to send secret information to Alice for many times, KMC must preserve a lot of public keys for Alice. Moreover Bob has gotten φ after a communication process with Alice! If another user encrypts plain text using the public, Bob will be able to decrypt it without being found. So KMC must not keep many copies of one public key. It should keep many different public keys for Alice otherwise Bob will be able to get any message other user sends to Alice in future. In practice KMC should keep M (M>>N) public key for Alice in which every public key should be given a unique id number. So does every user in the cryptosystem.

Now we can design a quantum public-key cryptosystem based on this idea.

3. Quantum Public-key Cryptosystem without Quantum Channels between any Users using Non-orthogonal States

Now we give the quantum public-key cryptosystem as follows.

3.1. Building the Public-key Cryptosystem

There are N users and a key management center (KMC) in the public-key cryptosystem. Any two users can exchange classical information through an authenticated public classical channel. The classical channel is open to everyone, that is to say, everyone can get the information transmitted though it. But it is authenticated so that a user can affirm the identity of the other one who is communicating with him. On the other hand every user can exchange qubits with KMC through an insecure quantum channel which every one can control. But two users needn't to exchange qubits so that no quantum channels are needed between them.

Every user, such as Alice, creates M n-qubit sequence in which one qubit is in the state |0>, |1>, |+>, or |-> at random. One n-qubit sequence is called a public key while all the states of the n qubits form an n-state sequence which is the corresponding private key. So the public key set of a user can be denoted as

$$\left(K_{PK}\right)_{Alice} = \{\,(i, Q_i)_{Alice},\quad i = 1,2,...,M\,\} \tag{10}$$

in which Q_i is an *n*-qubit sequence with the id number *i*. On the other hand the user's private key set can be denoted as

$$\left(K_{PA}\right)_{Alice} = \{\,(i, \varphi_i)_{Alice},\quad i = 1,2,...,M\,\}. \tag{11}$$

in which φ_i is the corresponding *n*-state sequence with the id number *i*. All users' public keys are kept by KMC and open to everyone who wants get them. But a user must keep his private keys absolutely secret by himself so that no other one including KMC can get it.

3.2. Process of Communication

If a user Bob wants to send an *n*-bit string *P* to another user Alice. They perform the following steps.

Step 1: Bob asks KMC for one of Alice's public keys.

Step 2: KMC chooses one public keys $(j, Q_j)_{Alice}$ from Alice's public key set $\left(K_{PK}\right)_{Alice}$ at random and sends it to Bob through the quantum channel.

Step 3: When Bob receives $(j,Q_j)_{Alice}$, he sends the id number *j* to Alice through the classical channel.

Step 4: When Alice receives the id number *j*, she takes out the corresponding private key $(j,\varphi_j)_{Alice}$. Then Alice produces the basis sequence *B* according to the Key Rule.

Step 5: Alice sends her basis sequence *B* to Bob.

Step 6: When Bob receives *B*, he measures $(j,Q_j)_{Alice}$ according to *B*. Then he records his measurement results according to the Coding Rule. Finally Bob gets an n-bit string *K*.

Step 7: Bob performs an XOR operation on *P* and *K* to get the cipher text *PS*. Then Bob sends *PS* to Alice.

Step 8: When Alice receives *PS*, she first produce an n-bit string *K*' from $(j,\varphi_j)_{Alice}$ according to the Coding Rule. It's obvious that *K*'=*K*.

Step 9: Alice performs an XOR operation on *PS* and *K'* to get a string *P*'.

It is easy to find *P*'=*P*. So Alice has gotten the message that Bob sends her.

If Alice wants to send a secret message to Bob, they need only exchange the roles in the communication process above. So any two users can achieve secret communications in this public-key cryptosystem.

3.3. Message Authentication

If Bob sends a secret message *PO* to Alice, he can sign the message to guarantee the reality of *PO*. What Bob needs to do is to attach a classical message (the authentication token) with the original message that he wants send to Alice. To produce the authentication token, Bob performs the following steps.

Step 1: Bob produces an abstract *PA* from *PO* which he wants to send Alice by a hash algorithm, such as SHA-1 algorithm. The length of *PA* is m.

Step 2: Bob chooses one of his private keys at random, such as $(i,\varphi_i)_{Bob}$. Then he produces an n-bit string *ST* form $(i,\varphi_i)_{Bob}$ according to the Coding Rule. Next Bob chooses the first m bits from *ST*. Finally Bob gets an m-bit string *PK*.

Step 3: Bob performs an XOR operation between *PA* and *PK*. Finally he gets an m-bit string *PS* which is just the authentication token.

Step 4: Bob attaches *PS* and the id number i with *PO*. So he gets a string *PX* which is the plain text to be submitted to Alice.

Notice that the length of *PX* should be n. So the length of the original message *PO* added with the length of k should be n-m. If *PO* can't satisfy it, we can always make it by dividing it into several parts or adding supplementary bits to it.

Then Bob and Alice can finish the communication as the steps in section 3.2.

After Alice gets the plain text *PX*, she can extract the original message *PO*, the authentication token *PS* and the id number i. To verify the authentication, she does the following steps.

Step 1: Alice gets Bob's no. i public key $(i,Q_i)_{Bob}$ from KMC through the quantum channel.

Step 2: Alice asks Bob for the measurement basis sequence through the classical channel.

Step 3: Bob produces the basis sequence *B* according to $(i,\varphi_i)_{Bob}$ and sends it to Alice through the classical channel.

Step 4: Alice measure $(i,Q_i)_{Bob}$ in *B*. Then she takes the first m measurement results and records them according to the Key Rule. Finally she gets an m-string *PK*' which is just equal to *PK*.

Step 5: Alice performs an XOR operation on *PK*' and *PS*. So she gets an m-bit string *PA*'.

Step 6: Alice produces the abstract *PA* of *PO* using SHA-1 algorithm just as Bob does.

Step 7: Alice compare *PA*' and *PA*. If they are identical, the verification succeeds. Alice can be sure that the message *PO* is really from Bob.

On the other hand by the SHA-1 algorithm Alice can also assure that the message *PO* hasn't been tampered by any eavesdroppers.

4. Security of the Public-key Cryptosystem

This quantum public-key cryptosystem is secure. If one user Bob sends a secret message to another user Alice, no other one including KMC can get the message. At the same time message authentication is also secure. No one can pretend to be Bob to send fake message to Alice. We prove it as follows.

We assume that an eavesdropper, such as Eve, wants to get the message transmitted from Bob to Alice.

4.1. Impossibility for Eve to get the Message

Eve can listen to both the classical channel and the quantum channel. She can catch all the classical information transmitted between Alice and Bob while she can catch Alice's private key $(j,Q_j)_{Alice}$ when KMC sends it to Bob. Obviously Eve can get the cipher text *PS* sent form Bob to Alice in step 8 of section 3.2. At the same time she also knows that the cipher text is produced by no. j public key with the plain text *P*. Eve knows that *PS* is produced by *P* XOR *K*. So what Eve needs to do is to get *K*.

When KMC sends Alice's public key $(j,Q_j)_{Alice}$ to Bob in step 1, Eve may catch it and try to do something to get the key. We can prove that Eve can't succeed.

First Eve can't measure $(j,Q_j)_{Alice}$ because she doesn't know the correct basis sequence *B* which Alice will produce and declare in step 3. Now if Eve measures $(j,Q_j)_{Alice}$ without the correct basis sequence *B*, she can just get some random measurement results. It's of no use to help Eve in getting *K*. Since Bob doesn't receive $(j,Q_j)_{Alice}$ because Eve has intercepted it, he won't send any message to Alice asking for the basis sequence *B* at all, or in other words, Alice will never send *B* to Bob. Of course Eve can't get *B*. Moreover the states of $(j,Q_j)_{Alice}$ will collapse if Eve chooses wrong basis, that is to say, the string *K* which Bob gets will be different from the string *K*' which Alice produces from $(j,\varphi_j)_{Alice}$. If Eve sends $(j,Q_j)_{Alice}$ which has been measured to Alice, Alice can't decrypt the cipher text at all. So she can affirm Eve's existing at once and abandon the communication. So Eve fails in getting the secret message without being found.

Second Eve may think that she can make a copy of $(j,Q_j)_{Alice}$ without measuring $(j,Q_j)_{Alice}$. Then she sends $(j,Q_j)_{Alice}$ to Bob and listens to the communicated information between Alice and Bob so that she can get *B* just as Alice. Next Eve does the same steps as Alice. Finally she can also get the message *P*. Such strategy of attack is infeasible at all. Quantum no-cloning theorem forbids anyone to clone an unknown quantum state. So it's impossible for Eve to make a copy of $(j,Q_j)_{Alice}$, not

mention to getting the basis sequence *B* and the plain text *P*. Whatever Eve does, the probability to get the plain text *P* for her is no more than the probability she just guesses it. Since *P* is an n-bit string, the probability for Eve to get *P* is no more than

$$P_{error} = \left(\frac{1}{2}\right)^n . \tag{12}$$

If n=1000, we have

$$P_{error} = \left(\frac{1}{2}\right)^{1000} \approx 10^{-300}. \tag{13}$$

It's a number too small to imagine. So Eve is sure to fail.

Third Eve may take the strategy of entanglement attack. After receiving $(j, Q_j)_{Alice}$, to each qubit (denoted as qubit 1) she creates an auxiliary qubit (denoted as qubit E) in the state |0>. Then Alice performs a CNOT (controlled NOT) operation on the two qubits in which the control qubit is qubit 1 and the target qubit is qubit E. Next Eve sends $(j, Q_j)_{Alice}$ to Bob. When Alice sends B to Bob through the classical channel, Eve also gets B. Then Eve measures the auxiliary qubit sequence with the aim to get *K*. But it's impossible for Eve to get *K* because Eve's measurement results are sure to be different from Bob's. Let's prove it as follows. After Eve's CNOT operation the state of whole two-qubit system comprise of qubit 1 and qubit E changes as table 2.

Table 2. Entanglement Attack

Before CNOT	After CNOT	basis
$\|0>_1\|0>_E$	$\|0>_1\|0>_E$	B_{01}
$\|1>_1\|0>_E$	$\|1>_1\|1>_E$	B_{01}
$\|+>_1\|0>_E$	$\|0>_1\|0>_E + \|1>_1\|1>_E$	B_{+-}
$\|->_1\|0>_E$	$\|0>_1\|0>_E - \|1>_1\|1>_E$	B_{+-}

When Eve gets *B*, she measures qubit E according to *B*. Then she produces a string according the Coding Rule from her measurement results. On the other hand Alice produces *K*' according to the Coding Rule in step 8. If the state of qubit 1 is |0> or |1>, Eve will get the same bit as Bob and achieves her goal. But if the state of qubit 1 is |+> or |->, to each qubit the probability that Eve gets the same bit as Bob is only 1/2. So the average probability that Eve gets the same bit as Bob is 3/4. Since there are n bits in *K*', the probability for Eve to get it is no more than

$$P_{error} = \left(\frac{3}{4}\right)^n . \tag{14}$$

If n=1000,

$$P_{error} = \left(\frac{3}{4}\right)^{1000} \approx 10^{-125}. \tag{15}$$

So such attack also fails.

4.2. Impossibility for KMC to Get the Message

In this public-key cryptosystem KMC can't get the message sent from Bob to Alice, either. We prove it as follows.

First KMC will be on the same boat as Eve when Bob sends the cipher text *PS* to Alice. KMC can get *PS*, too. But without *K* it's impossible for it to recover the plain text *P* from *PS* just as Eve.

Second since KMC keeps Alice's public keys, it may try to get something about the key to help itself in getting the plain text. But KMC doesn't hold the private key $(j,\varphi_j)_{Alice}$, that is to say, the states of all the qubits in $(j,Q_j)_{Alice}$ are unknown to KMC, either. Obviously KMC also can't measure $(j,Q_j)_{Alice}$ to help it to get the secret message because it can do nothing more than Eve. We have proved that Eve can't succeed. So does KMC.

4.3. Security against Fake Message Attack by Eavesdroppers

Eve may catch the cipher text *PS* sent from Bob to Alice and sends a fake message *FP* to Alice. It can be proved to be impossible as follows.

As known the cipher text *PS* is sent through the public classical channel. It's easy for Eve to catch *PS*. But Eve has no *K* to encode *FP*. Notice it, Alice will decode the cipher text FPS encoded from *FP* which Eve sends her with *K*' (*K*'=*K*). No mater what Eve does to treat *FP* to get *FPS*, the probability that Alice just gets *FP* after decoding FPS with *K*' is no more than the probability that Alice guesses every bit of *K*' or *K* correctly. So the probability that Eve make Alice to accept a fake message is

$$P_{error}=\left(\frac{1}{2}\right)^n \quad . \tag{16}$$

If n=1000,

$$P_{error}=\left(\frac{1}{2}\right)^{1000}\approx 10^{-300}. \tag{17}$$

It's an extreme small probability so that we can say that Eve can't succeed.

On the other hand Eve may catch the public key $(j,Q_j)_{Alice}$ when it's sent form KMC to Bob. But she still has no way in making Alice to receive a fake message because the states of the qubits in $(j,Q_j)_{Alice}$ are unknown to her. Moreover Eve will be found as long as she measure the qubit in $(j,Q_j)_{Alice}$. It's easy to prove that the probability Eve escaping from being found is equal to that in equation (15) in subsection 4.1, or in other words, it's about 10^{-125}.

So we can conclude that Eve can't have Alice to accept a distort message.

4.4. Security against Fake Message Attack by KMC

Although KMC keeps Alice's public keys, it can't make Alice to accept a fake message, either. If Bob asks KMC for Alice's public key $(j,Q_j)_{Alice}$, how can KMC do? First KMC can't measure $(j,Q_j)_{Alice}$ because Alice and Bob will find it and abandon the process of communication, which has been proved in subsection 4.3. Second quantum no-cloning theorem forbid KMC to produce a copy of $(j,Q_j)_{Alice}$. So what KMC can do is no more than what Eve can do. We have proved that Eve can't make Alice to accept a fake message. So KMC can't make it, either.

4.5. Security of Message Authentication

Message authentication in this cryptosystem is easy to be proved to be secure. No one can counterfeit Bob to send a message to another user. To produce the authenticated token, Bob produces *ST* from his private $(i,\varphi_i)_{Bob}$ according to the

Coding Rule and chooses the first m bits of *ST*. So he gets a random binary string *PK*. The authenticated token *PS* is produced by $PK \otimes PA$ in which *PA* is the abstract of the plain text *PO*. When Alice receives the cipher text and decrypts it, she will extract *PS* and the id number *i*. Then she ask KMC for Bob' public key $(i, Q_i)_{Bob}$ and chooses out the first m qubits. By measuring these qubits according to *B* Alice gets *PK*'. Obviously *PK*= *PK*'. Then Alice performs $PK' \otimes PA$ to get *PS*'. If *PS*'= *PS*, the digital signature is verified. It's easy to find that *PK*= *PK*' is necessary to produce the correct authentication token. But no one except Bob holds Bob's private key $(i, \varphi_i)_{Bob}$. This means that Eve can't get *PK* (or *PK*') from $(i, \varphi_i)_{Bob}$ as Bob does. Without *PK* (or *PK*') the probability that Eve just produce the correct authentication token *PS* is at most

$$P_{error} = \left(\frac{1}{2}\right)^m . \tag{23}$$

If m=100,

$$P_{error} = \left(\frac{1}{2}\right)^{100} \approx 10^{-30} . \tag{24}$$

So no one can forge Bob's authentication token, which also means that Alice can assure that the message is from Bob. On the other hand SHA-1 algorithm guarantees that the message is real and intact because the probability using another message to produce the same abstract is negligible.

4.6 Security against Forward Search Attack

In classical public-key cryptosystem forward search attack is a powerful strategy of attack. Eve uses Alice's public key to encrypt as many plain texts as possible and save the (plain text, cipher text) pairs in her database. Next Eve listens to the channel and catches all the cipher texts sent to Alice. Then Eve searches the cipher text in her database. If she does find a matched (plain text, cipher text) pair, she is sure that the plain text is just the secret message which is being sent to Alice. But forward search attack is invalid in this quantum public-key cryptosystem because Alice has many (public key, private key) pairs in which one pair can be used for only one time. If Eve encrypts the plain text *P* using one of Alice's public keys $(j, Q_j)_{Alice}$ to get the cipher text *PS*, the (public key, private key) pair has been consumed. If a user, for example Bob wants to send a secret message to Alice, he has to choose another public key $(j', Q_{j'})_{Alice}$ to encrypt the plain text, that is to say, it's impossible for Eve to find the cipher text in her database at all. So forward search attack is impossible to succeed.

4.7. Security against Resend Attack

In classical public-key cryptosystem Eve can catch the cipher text and make a copy of it. Then Eve resends the cipher text again to Alice when she wants. Alice can decrypt it successfully without finding anything wrong. So Eve has made Alice to receive a repeated message even though she knows nothing about the message. But our quantum public-key cryptosystem is immune to such attack. When Alice receives a cipher text which is sent from Eve as a resent message, she has no correct private keys to decrypt it because the key has been consumed after the first communication process. So Alice can't recover the plain text at all. What Alice can get is not a repeated message but a random string. So she will see through Eve's cheating at once.

4.8. Security against Chosen Plain Text Attack

If Eve can produce the cipher text of random plain text, by comparing many (plain text, cipher text) pairs she may find some regular patterns which can helps Eve to get the key. This is called chosen plain text attack. It is a serious threat to classical public-key cryptosystem. But our quantum public-key cryptosystem is immune to chosen text attack because a (public key, private key) can be used for one time. Every cipher text is produced from a unique public key. There are nothing correlations among different (plain text, cipher text) pairs. So the quantum public-key cryptosystem is secure against chosen text attack.

5. Feasibility Analysis of the Public-key Cryptosystem

First in this public-key cryptosystem what users and KMC have to do are performing the single particle measurement on qubits and transmitting qubits through a quantum channel, which have been mature technology for many years. No entangled states and complex quantum operations are needed. So it's easier to carry out in practice.

Second in this public-key cryptosystem any two users needn't exchange qubits at all. What they need to do is to communicate with each other though a public classical channel. So it not only saves much resource but also avoids many costs for maintain the quantum channels, which make it easier to carry out and more robust in practice. This is a big advantage of this quantum public-key cryptosystem.

Third as known all quantum cryptographic protocols depends on the special properties of quantum system. But in reality a quantum system will occur decoherence over time inevitably, which make it to degenerate into a classical system and lose quantum coherence. So any quantum cryptographic protocol will collapse once decoherence takes place. This difficulty can be solved by using quantum systems which has a long decoherence time as the carrier of qubit, such as photon. Another solution is to replace the public keys periodically. Secret communication can be completed successfully so long as it finishes before the public keys undergo decoherence.

6. Discussion and Conclusion

There is a weak point of this public-key cryptosystem. A public key can be used for only one time. This means that KMC must keep many public keys for one user. Once a user's public keys run out, no one can sends message to him again. This problem can be overcome by developing public-key cryptosystem in which the public key is reusable. We will study it in future work.

In this paper we provide a quantum public-key cryptosystem without quantum channels between any two users using non-orthogonal states. N users can achieve secret communications by the help of a key manage center (KMC). The principles of quantum mechanics guarantee that the public-key cryptosystem is unconditionally secure. On the other hand message authentication can be accomplished. There are no quantum channels needed between any two users. So the quantum public-key cryptosystem is easier to carry out and more robust in practice.

Acknowledgements

This work is supported by Natural Science Foundation of China (Grants 61472412), Natural Science Foundation of the Education Department of Henan Province of China (Grants 14A520012) and Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2014JM2-6103). The authors wish to thank Ruqian Lu for directing us into this research.

References

[1] C. H. Bennett and G. Brassard, Proceedings of IEEE International conference on Computers, Systems and Signal Processing, (**1984)** December; Bangalore, India.
[2] A. K. Ekert, "Physical Review Letters", vol. 67, no. 6, **(1991**), p. 661.
[3] C. H. Bennett, G. Brassard and N. D. Mermin, "Physical Review Letters", vol. 68, no. 5, **(1992), p.** 557.
[4] H.-K. Lo and H. F. Chau, "Science", vol. 283, **(1999), p.** 2050**.**
[5] A. Cabello, "Physical Review Letters", vol. 85, no. 26, **(2000**), p. 5635.
[6] T. Nguyen, M. A. Sfaxi, S. Ghernaouti-Hélie, "Journal of Networks", vol. 1, no. 5, **(2006),** p. 9.
[7] R. Namiki and T. Hirano, "Physical Review A", vol. 74, no. 3, **(2006),** p. 032301.
[8] B. Qi, Y. Zhao, X. F. Ma, H. K. Lo and L. Qian, "Physical Review A", vol. 75, no. 5, **(2007),** p. 052304.
[9] R. Matsumoto, "Physical Review A", vol. 76, no. 6, **(2007),** p. 062316.
[10] Y. Zhao, B. Qi and H. K. Lo, "Physical Review A", vol. 77, no. 5, **(2008),** p. 052327.
[11] T. Choi and M. S. Choi, "Journal of Physics: Condensed Matter", vol. 20, **(2008),** p. 275242.
[12] K. Horodecki, M. Horodecki, P. Horodecki, D. Leung and J. Oppenheim, "IEEE Transaction on Information Theory", vol. 54, no. 6, **(2008),** p. 2604.
[13] J. Barrett, R. Colbeck and A. Kent, "Physical Review A", vol. 86, **(2012),** p. 062306.
[14] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, "Journal of Cryptology", vol. 5, no. 1, **(1992),** p. 3.
[15] T. Kimura, Y. Nambu, T. Hatanaka, A. Tomita, H. Kosaka and K. Nakamura, arXiv:quant-ph/0403104 **(2004).**
[16] W. T. Buttler, "Physical Review Letters", vol. 81, no. 15, **(1998),** p. 3283.
[17] R. Rivest, A. Sharmir and L. Adleman, "Communications of ACM", vol. 21, no. 2, **(1978),** p. 120**.**
[18] P. W. Shor, "Proceedings of 35th Annual IEEE Symposium on Foundations of Computer Science", **(1994)** November; Santa Fe, US.
[19] D. Gottesman and I. Chuang, arXiv:quant-ph/0105032 (**2001).**
[20] G. Nikolopoulos, "Physical Review A", vol. 77, **(2008),** p. 032348.
[21] G. Nikolopoulos and L. Ioannou, "Physical Review A", vol. 79, **(2009),** p. 042327.
[22] L. Ioannou and M. Mosca, arXiv:quant-ph/0903.5156 **(2009)**.
[23] L. Ioannou and M. Mosca, Proceedings of the 6th Conference on the Theory of Quantum Computation, Communication and Cryptography, **(2011)** May 24-26; Madrid, Spain.
[24] U. Seyfarth, G. Nikolopoulos and G. Alber, "Physical Review A", vol. 85, **(2012),** p**.** 022342**.**

Authors

Xiaoyu Li, He was born in Nanyang, China in 1974. He received the Ph. D. degree in computer software and theory from Institute of Computing Technology, Chinese Academy of Sciences, China in 2004. He majors in quantum information and quantum computing; mobile computing. He is an associate professor at School of Information Engineering, Zhengzhou University, China.

Yuwen Chen, He was born in Luoyang, China in 1988. Now he is a graduate student at School of Information Engineering, Zhengzhou University. He majors in quantum information and quantum computing.