Quantum election scheme based on anonymous quantum key distribution*

Zhou Rui-Rui(周瑞瑞)[a)b)] and Yang Li(杨 理)[a)†]

[a)] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*

[b)] *Graduate University of Chinese Academy of Sciences, Beijing* 100049, *China*

(Received 7 January 2012; revised manuscript received 22 March 2012)

An unconditionally secure authority-certified anonymous quantum key distribution scheme using conjugate coding is presented, based on which we construct a quantum election scheme without the help of an entanglement state. We show that this election scheme ensures the completeness, soundness, privacy, eligibility, unreusability, fairness, and verifiability of a large-scale election in which the administrator and counter are semi-honest. This election scheme can work even if there exist loss and errors in quantum channels. In addition, any irregularity in this scheme is sensible.

Keywords: quantum election, quantum key distribution, conjugate coding

PACS: 03.65.–w, 03.67.–a **DOI:** 10.1088/1674-1056/21/8/080301

1. Introduction

Achieving security and privacy simultaneously is a problem that must be solved in a voting system if the system is to be used for serious large-scale government elections.[1] In a traditional paper ballots method, the voting system often uses a locked box to solve this problem: after filling out the ballots, the voters put the ballots into the box one by one so as to guarantee the privacy of the voters. With the advent of the information age, more and more daily life matters are transacted through a digital communication network. An electronic voting scheme is appealing as an alternative to the traditional paper ballots method.

The first cryptographic voting protocol was suggested by Chaum in 1981.[2] This protocol uses public key cryptography and rosters of digital pseudonyms to conceal the identity of votes, but it does not guarantee that the identity of voters cannot be traced. Later Chaum proposed a protocol that provided unconditionally security against tracing the voting.[3] However, elections based on this protocol can be disrupted by a single voter who can break RSA. The authors in Ref. [4] suggested an unconditionally secure solution of the so-called dinning cryptographer's problem that can be used to guarantee unconditionally the anonymity of the voters. Privacy in this scheme is assured, but the protocol cannot avoid a voter voting an arbitrary number times, so cheating is easy. Solving the privacy and fairness problems at the same time seems to be a drawback of cryptographic voting schemes. Over the years, much attention has been paid to electronic voting schemes.[5−11] In practice, an electronic voting scheme can use blind signature, homomorphic-encryption, and other classical cryptographic techniques. For example, a secure electronic voting scheme based on blind signature was presented in Ref. [9], and the authors in Ref. [5] presented a practical secret voting scheme which is suitable for large scale election and solves the privacy and fairness problems at the same time. However, the security of these schemes depends on the solving of certain difficult mathematical problems, such as factoring large numbers and solving discrete logarithms. It could be easily broken with the emergence of quantum computers, thus allowing an attacker to disturb the election.

The physics of quantum cryptography pioneers a new way of election. Different from the classical cryptography, the security of quantum cryptography is based on the fundamental laws of quantum physics to provide unconditional security.[12−15] For example, quantum key distribution can use the laws of physics

*Project supported by the National Natural Science Foundation of China (Grant No. 61173157) and the Strategy Pilot Project of Chinese Academy of Sciences (Grant No. XDA06010702).

†Corresponding author. E-mail: yangli@gucas.ac.cn

© 2012 Chinese Physical Society and IOP Publishing Ltd

as the basis for a scheme to distribute secure cryptographic keys.[16−18] Hence, more and more researchers are showing their interest in quantum election.

As in an election scheme it is more desirable that the identity of a voter, rather than his vote, be kept secret, quantum anonymous transmission[19] seems to be an inspiration for quantum election. A few quantum election protocols in the context of this notion have been proposed recently: Vaccaro *et al.*, proposed quantum protocols for voting and surveying;[20] a key feature of their schemes is the use of entangled states to ensure that the votes are anonymous and to allow the votes to be tallied. Two similar protocols suggested the voters apply local operations to an entangled state to guarantee the anonymity of participants in the voting procedures,[21,22] reference [22] also presented a method of preventing voters from casting more than one vote. Another similar protocol based on quantum mechanics had been proposed in Ref. [23], which used quantum mechanics to maintain the privacy of the voters. In 2011, Horoshko and Kilin proposed a new protocol for quantum anonymous voting,[24] their protocol protected both the voters from a curious tallyman and all the participants from a dishonest voter in an unconditional way. All the mentioned protocols provide unconditional anonymity of voting, but every voter in these protocols can only decide to vote for two alternatives, which may be impractical in an election that has many candidates. Unlike these schemes that used quantum entanglement for obtaining anonymity, the authors in Refs. [25] and [26] presented quantum voting schemes that involved many candidates. In Ref. [26] a blank ballot was an unknown quantum state for a voter, and the voter randomized his ballot to guarantee anonymity. But there was still a problem: as there was no way for a voter to verify whether his/her ballot had been counted correctly, the voter could not make sure that he/she had voted successfully.

In this paper, we present a quantum election scheme based on anonymous quantum key distribution (AQKD). Our scheme should be easier to implement in comparison with the schemes in Refs. [20]–[25] and [27], because we do not use quantum entanglement and the reading of the result does not need complicated measurement. In addition, the candidates in our scheme are not limited to two alternatives. Moreover, our scheme can ensure that each eligible voter can make sure that his/her ballot has been counted successfully compared with the scheme in Ref. [26].

The rest of this paper is organized as follows. In the next section, we present the unconditionally secure AQKD protocol. In Section 3, we present our quantum election scheme in detail. In Section 4, we analyze the security of the scheme. Subsequently, we discuss the comparison between our scheme and the previous traditional and quantum election schemes in Section 5. Finally, we make conclusions in Section 6.

2. Anonymous quantum key distribution

2.1. AQKD protocol

To start with, we present a protocol AQKD$(\cdot)$ for anonymous quantum key distribution with unconditional security.

We use the notation $\|$ to denote the concatenation of strings and $E_k[M]$ to denote an unconditionally secure symmetric encryption algorithm which encrypts message M by using the one-time-use key k, and the length of k is equal to the length of M.

Suppose someone U_i wants to anonymously establish a key with a certain entity Charlie with the help of another entity Bob. For this he uses the following protocol:

Protocol AQKD(U_i)

Prerequisite: U_i has established a 10-m-bit key $P_{\rm bi}$ with Bob by directly contacting or using an unconditionally secure quantum key distribution protocol.[28−30] Where $P_{\rm bi} = (p_i\|x_i\|y_i\|z_i)$. We denote $S_i = (x_i\|y_i\|z_i)$. Here the entity Bob acts as the administrator; he only helps the users who pre-share private key P_{b_*} with him to run the AQKD protocol. In other words, Bob will certify the authority of each user U_i at the beginning of the protocol.

Step 1 U_i sends p_i to Bob via a secure classical channel.

Step 2 Bob checks whether p_i' is correct after he receives it; if it is correct, Bob establishes an m-bit key k_c and a $(9+1)m$-bit key $k_{\rm bc}$ with Charlie by directly contacting or using an unconditionally secure quantum key distribution protocol. Then he computes

$$X = E_{k_{\rm bc}}[k_c\|S_i] = E_{k_{\rm bc}}[k_c\|(x_i\|y_i\|z_i)] \qquad (1)$$

and sends it to Charlie via a secure classical channel.

Step 3 When Charlie receives X', he uses $k_{\rm bc}$ to decrypt it. After gaining the outcome $(k_c'\|S_i')$, he checks whether k_c' is equal to k_c. If the check succeeds,

Charlie can make sure that his outcome S_i' is equal to Bob's plaintext S_i.

Step 4 U_i randomly chooses two strings r_{i1}, $r_{i2} \in \{0,1\}^{3m}$ to generate his qubits $|\alpha_i\rangle$ by the method of conjugate coding:[31]

$$|\alpha_i\rangle = H^{r_{i1}}|r_{i2}\rangle = \otimes_{j=1}^{3m}(H^{r_{i1}^j}|r_{i2}^j\rangle), \tag{2}$$

where r_{i1}^j and r_{i2}^j denote the j-th bit of r_{i1}, r_{i2}, and $H^{r_{i1}} = \otimes_{j=1}^{3m} H^{r_{i1}^j}$,

$$H^0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad H^1 = H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Then U_i anonymously transmits $|\alpha_i\rangle$ to Charlie via an error-free quantum channel.

Step 5 Charlie randomly chooses the measuring basis to perform single-particle measurements on the qubits $|\alpha_i\rangle'$ that he received. After gaining the outcome r_{i2}', he randomly chooses a subset σ_i of the set $\{1,2,\ldots,3m\}$, where $\sigma_i = \{d_i^1, d_i^2, \ldots, d_i^m\}$, with $1 \leqslant d_i^1 < d_i^2 < \cdots < d_i^m \leqslant 3m$; then generates the string $f_i \in \{0,1\}^{3m}$, where

$$f_i^j = \begin{cases} 0, & \text{for } j \in \sigma_i, \\ 1, & \text{for } j \notin \sigma_i, \end{cases} \tag{3}$$

where $j \in \{1,2,\ldots,3m\}$ throughout. Then he computes

$$F_i = (r_{i2}')^{d_i^1}(r_{i2}')^{d_i^2}\ldots(r_{i2}')^{d_i^m}, \tag{4}$$

and publicly publishes the group (F_i, f_i) with the corresponding measuring basis of the qubits $|\alpha_i\rangle'$.

Step 6 U_i extracts $\sigma_i = \{d_i^1, d_i^2, \ldots, d_i^m\}$ through the string f_i that is published by Charlie and computes

$$F_i' = (r_{i2})^{d_i^1}(r_{i2})^{d_i^2}\ldots(r_{i2})^{d_i^m}, \tag{5}$$

if F_i' satisfies the following conditions:

$$\begin{aligned} &\text{if Charlie measures } |\alpha_i^{d_i^j}\rangle' \text{ correctly,} \\ &\text{then } F_i'^{d_i^j} = F_i^{d_i^j}, \end{aligned} \tag{6}$$

where $j \in \{1,2,\ldots,m\}$ throughout, U_i generates the string $r_{i3} \in \{0,1\}^{2m}$, where

$$r_{i3}^{e_i^j} = \begin{cases} 1, & \text{for Charlie measuring } |\alpha_i^{e_i^j}\rangle' \text{ correctly,} \\ 0, & \text{for Charlie measuring } |\alpha_i^{e_i^j}\rangle' \text{ incorrectly,} \end{cases} \tag{7}$$

where $j \in \{1,2,\ldots,2m\}$ throughout, with

$$\begin{aligned} &\{e_i^1, e_i^2, \cdots, e_i^{2m}\} = \{1,2,\cdots,3m\} - \sigma_i, \text{ and} \\ &e_i^1 < e_i^2 < \cdots < e_i^{2m}. \end{aligned}$$

Subsequently U_i computes

$$Y_i = E_{z_i}[y_i \| r_{i3}], \tag{8}$$

and anonymously sends (x_i, Y_i) to Charlie via a secure classical channel.

Step 7 When he receives (x_i', Y_i'), Charlie checks whether he has accepted x_i' before. If he has not accepted x_i' before, he verifies whether x_i' is equal to x_i that was recorded in his database. If it is not, Charlie rejects to accept Y_i'; otherwise Charlie uses z_i to decrypt Y_i'. With the outcome $(y_i' \| r_{i3}')$, Charlie checks whether y_i' is equal to y_i. If it is true, Charlie uses r_{i3}' to extract the final key G_i, which is made up by the measurement outcome of the qubits that Charlie has chosen the correct measuring basis.

Now the protocol is completed, U_i establishes a key string G_i with Charlie anonymously.

2.2. Security analysis

As the qubits $|\alpha_i\rangle$ in the protocol is an unknown quantum state to others, it is impossible to make a copy. An attacker has to measure the initial qubits if he wants to get some useful information. However, as the attacker does not know the encoding basis of the qubits, he will choose the wrong measuring basis with the probability 50% for each qubit, then introduce no less than 25% error rate in the string F_i' and to be detected by U_i in Step 6. The probability that others except U_i and V_i get the correct x-bit key G_i is not more than $(1/2)^x$, when x is large enough, the probability is negligible. So nobody except U_i and Charlie knows the value of G_i. As Charlie is not aware of U_i's identity ID_i, U_i has established an x-bit key G_i with Charlie anonymously.

2.3. Practicability analysis

The protocol can also work even if there exist loss and errors in quantum channels. In order to estab-

lish an error-free key, Charlie can publish all the serial numbers and measuring basis of the qubits that he has received, then publish the check string F_i and its serial numbers f_i in Step 5. If the error rate of U_i's string F_i' is acceptable, U_i can make sure that the key distribution is successful. After verifying that the key distribution is successful, U_i extracts r_{i3} and computes the key string G_i which contains acceptable errors from r_{i2} by using the serial numbers and measuring basis that were published by Charlie, and subsequently uses G_i and a key redistribution protocol[32] to establish the final error-free key K_{ic} given below.

(i) U_i randomly chooses a string as the final key K_{ic}, then uses a classic error correction coding (ECC) to encode it. We use D_i to denote the error correction code of K_{ic}. Then U_i computes

$$Y_i = E_{z_i}[y_i \| r_{i3}], \quad Z_i = G_i \oplus D_i \tag{9}$$

and sends (x_i, Y_i, Z_i) to Charlie.

(ii) When he receives (x_i', Y_i', Z_i'), Charlie checks whether he has accepted x_i' before. If he has not accepted x_i' before, he verifies whether x_i' is equal to x_i that was recorded in his database. If it is not, Charlie rejects Y_i'; otherwise Charlie uses z_i to decrypt Y_i'. With the outcome $(y_i' \| r_{i3}')$, Charlie checks whether y_i' is equal to y_i. If it is, Charlie uses r_{i3}' to extract G_i', then he computes

$$D_i' = Z_i' \oplus G_i' = D_i \oplus e, \tag{10}$$

then Charlie decodes D_i' to obtain the final error-free key K_{ic}.

Obviously, if the error correction ability of the chosen error correction code is strong enough to correct the errors in D_i', Charlie will obtain the correct K_{ic} at last.

Generally, a quantum key distribution protocol is a key expansion protocol, in which the two parties in the protocol should pre-share a short key, which is used for identification and authentication of classical communication, to distribute a new longer key. In our AQKD protocol, U_i can anonymously establish a key with Charlie with the help of Bob under the condition that there is no key shared between U_i and Charlie in advance. In addition, the final key between U_i and Charlie is private to others (Bob included). The AQKD protocol plays a unique role in some cases (such as in an anonymous election scheme) although the final key that is generated by the protocol is shorter than the keys that have been used up between U_i and Bob, and Bob and Charlie during the protocol.

3. Quantum election scheme

3.1. Security definition and notations

There are many properties that a perfect voting scheme should have. Reviewing the properties that are described in the literature,[3,5,8] we define a secure quantum election scheme which should have the following properties.

i) Completeness: All valid votes are counted correctly.

ii) Soundness: The dishonest voter cannot disrupt the voting.

iii) Privacy: The content of the ballots is invisible to others.

iv) Eligibility: Only eligible voters are permitted to vote.

v) Unreusability: Each eligible voter can vote successfully only once.

vi) Fairness: Nothing must affect the voting.

vii) Verifiability: Each eligible voter can check whether his/her ballot has been counted successfully.

We use ID_i to represent the identity of the eligible voter V_i.

3.2. Quantum election scheme

A complete voting process in our election scheme includes four phases: initial phase, authentication phase, key distribution phase, and voting phase. Several voters V_j, $j = 1, 2, \ldots, N$, the voting administrator Bob, and the counter Charlie are also involved.

In order to ensure the successful implementation of the election, we assume the two parties Bob and Charlie in our scheme are semi-honest (honest but curious). In addition, they are two independent parties, which means they will not collaborate to trace a ballot.

3.2.1. Initial phase

The initial phase is as follows: before the authentication phase, the voting administrator Bob will publish a set $\mathscr{S} \subset \{0,1\}^s$. Each element of the set is an s-bit string that is randomly chosen by Bob to represent an eligible candidate. In addition, Bob will announce the corresponding candidate of each element. In the election scheme, an eligible voter V_j chooses the corresponding element of one candidate as his ballot v_j. We assume s is large enough to ensure that the

probability that a random s-bit string is an element of set $\mathscr{S}$ is negligible.

The voting administrator Bob establishes a key k_{bj}, where $k_{bj} = (k_j \| a_j \| b_j \| c_j)$, with every eligible voter V_j, $j = 1, 2, \ldots, N$, by directly contacting or using an unconditionally secure quantum key distribution protocol.[26−28] All the four parts of k_{bj} are selected uniquely for V_j.

All this work should be completed in advance.

3.2.2. Authentication phase

When the eligible voter V_j wants to vote, he sends a request by sending the group (ID_j, k_j) to Bob.

When Bob receives (ID_j, k_j) and the corresponding qubits, he checks whether V_j has successfully applied for voting before. If V_j has applied successfully before, Bob rejects his request; otherwise, Bob verifies the string k_j, if it is correct, Bob accepts V_j's request and records the corresponding $(i, ID_j, a_j b_j c_j)$ in a table with number i (See Table 1).

At the end of the authentication phase, Bob announces the number of the verified voters (we denote it by n) and publishes a set that contains all the verified ID_j, $j \in \{1, 2, \ldots, N\}$. Now the scheme turns to the key distribution phase.

Table 1. Bob's list of the verified voters.

Entry	Identity	Related information
1	ID_t	$S_1 : (a_t, b_t, c_t)$
⋮	⋮	⋮
i	ID_j	$S_i : (a_j, b_j, c_j)$
⋮	⋮	⋮
n	ID_l	$S_n : (a_l, b_l, c_l)$

3.2.3. Key distribution phase

After dealing with all the applied voter V_j's requests (ID_j, k_j), $j \in \{1, 2, \ldots, N\}$, Bob helps each verified voter V_j to execute an AQKD protocol to establish a $2s$-bit key K_{ic} between V_j and Charlie by running $\mathbf{AQKD}(U_i)$ with $U_i = V_j$.

Table 2. List of Charlie's key strings.

(K'_{*c})		Remark (a_*)
K'_{*cL}	K'_{*cR}	
K'_{1cL}	K'_{1cR}	a_t
⋮	⋮	⋮
K'_{icL}	K'_{icR}	a_j
⋮	⋮	⋮
K'_{ncL}	K'_{ncR}	a_l

After establishing key strings with all the verified voters, Charlie records them in a table (see Table 2) and publicly announces that the key distribution phase is completed, and the scheme turns to the voting phase.

Here $K'_{*c} = K'_{*cL} \| K'_{*cR}$, and $K'_{*cL}, K'_{*cR} \in \{0, 1\}^s$.

3.2.4. Voting phase

When the verified voter V_j makes sure that he has anonymously established a $2s$-bit key k_{ic} with Charlie successfully, he chooses an element from set $\mathscr{S}$ as his ballot v_j and anonymously sends $(K_{icL}, E_{K_{icR}}[v_j])$ to Charlie.

After receiving $(K_{icL}, E_{K_{icR}}[v_j])$, Charlie checks whether K_{icL} is in the first column of Table 2. If it is, Charlie checks whether K_{icL} has been accepted before. If not, Charlie extracts the corresponding group $(K'_{icL}, K'_{icR}, a_j)$, where $K'_{icL} = K_{icL}$, from Table 2 and uses K'_{icR} to decrypt $E_{K_{icR}}[v_j]$. If the outcome $v_j \in \mathscr{S}$, Charlie counts v_j and accepts K_{icL}.

After dealing with all the groups $(K_{icL}, E_{K_{icR}}[v_j])$, Charlie deletes the lines in which the first item K_{*cL} has not ever been accepted by him from Table 2. Then Charlie publishes a set which contains all the corresponding strings $\{a_j\}$ of the accepted K'_{icL}. Both Bob and the verified voter V_j can verify whether V_j has voted successfully by searching the value of a_j in this set. Bob communicates with each verified voter V_i that has not voted successfully to establish a new k_{bi}, then allows these failed voters for a new round of election.

When all the verified voters vote successfully, Charlie counts the number of each candidate's ballots. Subsequently, he randomly arranges all the accepted groups (K_{icL}, v_j) and publicly publishes them (See Table 3). The scheme is now completed.

Table 3. List of ballots (for public).

Ballot	Additional information
⋮	⋮
v_j	K_{icL}
⋮	⋮

4. Security analysis of the election scheme

In this section, we are to discuss the security of the quantum election scheme. We will prove that the

scheme has the properties of completeness, soundness, privacy, eligibility, unreusability, fairness, and verifiability when the administrator and the counter are semi-honest.

4.1. Completeness

The completeness means that all the valid ballots in the scheme are counted correctly. In this quantum election scheme, if all the three participants (the voters, the voting administrator, and the counter) are honest, after receiving an eligible voter V_j's voting request, Bob will send the messages S_i to Charlie honestly in the key distribution phase; when V_j makes sure that the anonymous key distribution protocol is successful, he will honestly send the messages $(K_{icL}, E_{K_{icR}}[v_j])$ to Charlie; After extracting V_j's ballot v_j from the cipher $E_{K_{icR}}[v_j]$, Charlie will count it honestly. So the result of the election is trustable.

4.2. Soundness

When a dishonest voter V_j wants to disrupt the voting, he may keep omitting the valid group $(K_{icL}, E_{K_{icR}}[v_j])$ or sending an invalid group $(K_{icL}, E_{K_{icR}}[v_x])$, where v_x represents an invalid candidate, to Charlie in the voting phase, then applying for a new round of voting. In the first case, V_j will be detected by Bob, because before a new round of voting Bob has to establish a new key with V_j. If V_j fails too many times, Bob will detect him/her. In the second case, V_j will be detected by Charlie in the voting phase. Therefore, the dishonest voter cannot disrupt the election.

4.3. Privacy

In the key distribution phase, the voter V_j sends his quantum blank vote $|\alpha_i\rangle$ to the counter Charlie. As $|\alpha_i\rangle$ is independent of v_j before the voting phase, neither an attacker nor a curious Bob can gain anything about v_j even if they get the entire state $|\alpha_i\rangle$. As the anonymous quantum key distribution protocol in this scheme is unconditionally secure, only Charlie and V_j know the correct K_{ic} if V_j ensures that his anonymous quantum key distribution is successful; although Charlie knows the value of v_j, he cannot gain the corresponding ID_j. Hence, nobody except V_j can match ID_j with the ballot v_j.

In this scheme it is possible to assume that Bob and Charlie are independent parties, i.e., they will not cooperate to trace the ballots.

4.4. Eligibility

The eligibility of a voting scheme means only eligible voters are permitted to vote, and it is impossible for an ineligible voter to forge a valid ballot. As described in the authentication phase, Bob will check the identity of a voter and reject his/her voting request if the voter cannot pass through the identity authentication, therefore only eligible voters are permitted to vote.

If an ineligible voter V_e wants to vote successfully, he has only two ways; one is to impersonate an eligible voter V_j in the voting phase, the other is to generate a valid ballot in the voting phase. In the first case, suppose V_e has intercepted V_j's qubits $|\alpha_i\rangle$, he replaces $|\alpha_i\rangle$ by $|\alpha_e\rangle$ and sends $|\alpha_e\rangle$ to Charlie. In order to vote successfully, V_e has to compute

$$Y_e = E_{c_j}[b_j \| r_{e3}]$$

and send the group (a_j, Y_e) to Charlie in the key distribution phase. However, V_e has no way of obtaining the value of (a_j, b_j, c_j), thus it is impossible for him to compute the correct Y_e, so he cannot impersonate V_j successfully; in the second case, V_e may monitor the classical channel to gain a verified voter V_j's group $(K_{icL}, E_{K_{icR}}[v_j])$, and send his group $(K_{icL}, E_{K_{icR}}[v_e])$ to Charlie. If his group reaches earlier than V_j's group, Charlie will accept $(K_{icL}, E_{K_{icR}}[v_e])$. However, it is impossible for V_e to compute $E_{K_{icR}}[v_e]$, for he does not have the encryption key K_{icR}. At the same time, as s is large enough, the probability that a random s-bit string v_* be a valid ballot is negligible. Therefore it impossible for an attacker to forge a valid vote.

In this paper we assume the administrator Bob and the counter Charlie are semi-honest. Actually, a malicious administrator Bob or a malicious counter Charlie may also try to forge valid votes: as Bob knows (a_j, b_j, c_j), he can easily impersonate the eligible voter V_j and vote successfully; in addition, a malicious Charlie may tamper with the ballot of an eligible voter. Although these irregularities will be discovered by V_j at the end of the scheme, there is no way to distinguish between a dishonest voter and a malicious Bob or a malicious Charlie. To avoid this, we can use a distributed scheme, in which several independent parties, e.g., several candidates of the election, collaborate to act as the voting administrator Bob and another group of several independent parties collaborate to act as the voting counter Charlie. They will not

cooperate with forging invalid votes or tracing valid ballots. The distributed scheme ensures the credibility of the voting administrator and the counter.

4.5. Unreusability

In the authentication phase, before responding to an eligible voter V_j's voting request, Bob will check whether V_j has successfully applied for voting before to avoid accepting V_j's repeated voting request. In addition, before he accepts a group $(K_{icL}, E_{K_{icR}}[v_j])$, Charlie will check whether the corresponding K_{icL} has been accepted before to avoid accepting the group twice. Moreover, before a new round of voting, Charlie will delete the lines in which the first item has not ever been accepted from Table 2 to avoid accepting an overdue group $(K_{icL}, E_{K_{icR}}[v_j])$ that had not been sent by V_i in the former round of voting. Hence, each eligible voter can vote successfully only once.

4.6. Fairness

Fairness means that nothing must affect the voting, especially that the counting of ballots does not affect the voting. In this scheme, the voting phase is done after the authentication phase and key distribution phase, and Charlie will not disclose the intermediate result of the election to others before the whole scheme is completed. Therefore the previous voters will not affect the subsequent voters. So this scheme is fair for all voters.

4.7. Verifiability

At the end of the scheme, Charlie will publish Table 3 publicly to all the voters. Each eligible voter V_j can search his group (K_{icL}, v_j) in this table. If (K_{icL}, v_j) is in the table, V_j can make sure that his vote has been counted correctly.

5. Discussion

In a traditional electronic voting scheme, there is no reliable way of knowing whether someone has successfully eavesdropped in the channel. Unlike the traditional electronic voting schemes that were based on classical cryptography, any eavesdropping behavior of quantum information in our scheme will be discovered. As the quantum blank votes in the scheme are unknown qubits to others, it is impossible to make a copy of them, so the attacker has to measure the initial qubits if he wants to gain some useful information. However, as the attacker does not know the encoding basis of the qubits, it is impossible for him to perform a correct measurement; moreover, his incorrect measurement will change the initial quantum blank vote and will then be detected by the voter in the key distribution phase. Once an eligible voter detects that his quantum blank vote has been changed, he will not send the encrypted ballot to the counter. In addition, as the quantum blank vote that transformed in the quantum channel is independent of the voter's choice, an attacker cannot get any useful information even if he intercepts the entire states. As a result, the attacker cannot get any available information.

Compared with the previous quantum voting schemes in Refs. [20]–[25] and [27], our scheme should be easier to implement because we do not use quantum entanglement. In view of the problem of verifiability that was not mentioned in Ref. [26], we also discussed the verifiability of our scheme to make sure that every voter can check whether his/her vote has been counted successfully.

In this paper we present an unconditionally secure quantum election scheme. As there is no reliable way to prevent the participants from copying the classical messages, we assume the administrator and counter in the scheme are semi-honest, and we use a distributed scheme to ensure the credibility of the voting administrator and the counter. A malicious voter cannot disrupt the voting and any irregularity in the scheme is sensible. The scheme can work even if there exists in the quantum channels a certain loss and the error rate.

6. Conclusion

We presented an unconditionally secure quantum election scheme based on anonymous quantum key distribution. Our scheme ensures the completeness, the soundness, the privacy, the eligibility, the unreusability, the fairness, and the verifiability of the quantum election scheme in which the administrator and counter are semi-honest. A malicious voter cannot disrupt the voting and any irregularity in the scheme is sensible. The key technique of our scheme is using unknown quantum states to execute authority-certified anonymous quantum key distribution with unconditional security. Different from previous quantum voting schemes, we do not use quantum entanglement,

and each eligible voter can check whether his/her vote has been counted successfully.

References

[1] Marius I and Ionut P 2011 *Computer Science Master Research* **1** 67

[2] Chaum D 1981 *Commun. ACM* **24** 84

[3] Chaum D 1988 *Advances in Cryptology-Eurocrypt* 1988 (Berlin: Springer-Verlag) p. 177

[4] Chaum D 1988 *J. Cryptology* **1** 65

[5] Fujioka A, Okamoto T and Ohta K 1993 *Lecture Notes in Computer Science* **718** 244

[6] Salomaa A 1996 *Public-Key Cryptography* 2nd edn. (Berlin: Springer-Verlag) pp. 200–202

[7] Sako K and Killian J 1995 *Advances in Cryptology-Crypto* (Berlin: Springer-Verlag) p. 411

[8] Cranor L F and Cytron R K 1996 *Washington University Computer Science Technical Report*

[9] Ibrahim S, Kamat M, Salleh M and Aziz S R A 2003 *Proceedings of the 4-th National Conference on Telecommunication Technology* p. 193

[10] Jafari S, Karimpour J and Bagheri N 2011 *Int. J. Comput. Sci. Eng.* **3** 2191

[11] Kumar S and Walia E 2011 *Int. J. Comput. Sci. Eng.* **3** 1825

[12] Boykin P O and Roychowdhury V 2003 *Phys. Rev.* A **67** 042317

[13] Chen K and Lo H K 2005 *arXiv*: 0404133 [quant-ph]

[14] Deng F G, Li X H, Zhou H Y and Zhang Z J 2005 *Phys. Rev.* A **72** 044302

[15] Wang T Y and Wen Q Y 2010 *Chin. Phys.* B **19** 060307

[16] Gisin N, Ribordy G, Tittel W and Zbinden H 2002 *Rev. Mod. Phys.* **74** 145

[17] Zhou Y Y and Zhou X J 2011 *Acta Phys. Sin.* **60** 100301 (in Chinese)

[18] Jiao R Z, Zhang C and Ma H Q 2011 *Acta Phys. Sin.* **60** 110303 (in Chinese)

[19] Christandl M and Wehner S 2005 *Advances in Cryptology-Asiacrypt* **3788** 217

[20] Vaccaro J A, Spring J and Chefles A 2007 *Phys. Rev.* A **75** 012333

[21] Dolev S, Pitowsky I and Tamir B 2006 *arXiv*: 0602087 [quant-ph]

[22] Hillery M, Ziman M, Buzek V and Bielikov M 2006 *Phys. Lett.* A **349** 75

[23] Bonanome M, Buzek V, Hillery M and Ziman M 2011 *Phys. Rev.* A **84** 022331

[24] Horoshko D and Kilin S 2011 *Phys. Lett.* A **375** 1172

[25] Li Y and Zeng G 2009 *Chin. Opt. Lett.* **7** 152

[26] Okamoto T, Suzuki K and Tokunaga Y 2008 *NTT Thchnical Review* **6**

[27] Li Y and Zeng G 2008 *Opt. Rev.* **15** 219

[28] Bennett C H and Brassard G 1984 *Advances in Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, December 10–12, 1984 Bangalore, India, p. 175

[29] Ekert A K 1991 *Phys. Rev. Lett.* **67** 661

[30] Bennett C H 1992 *Phys. Rev. Lett.* **68** 3121

[31] Wiesner S 1983 *ACM Sigact News* **15** 78

[32] Yang L, Wu L A and Liu S H 2002 *Acta Phys. Sin.* **51** 961 (in Chinese)