Research Article

Quantum Election Protocol Based on Quantum Public Key Cryptosystem

Wenhua Gao and Li Yang

1State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
2School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
3Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

Correspondence should be addressed to Li Yang; yangli@iie.ac.cn

Received 28 January 2021; Accepted 12 March 2021; Published 14 April 2021

Academic Editor: Prosanta Gope

Copyright © 2021 Wenhua Gao and Li Yang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is no quantum election protocol that can fulfill the eight requirements of an electronic election protocol, i.e., completeness, robustness, privacy, legality, unreusability, fairness, verifiability, and receipt-freeness. To address this issue, we employ the general construction of quantum digital signature and quantum public key encryption, in conjunction with classic public key encryption, to develop and instantiate a general construction of quantum election protocol. The proposed protocol exhibits the following advantages: (i) no pre-shared key between any two participants is required, and no trusted third party or anonymous channels are required. The protocol is suitable for large-scale elections with numerous candidates and voters and accommodates the situation in which multiple voters vote simultaneously. (ii) It is the first protocol that dismantles the contradiction between verifiability and receipt-freeness in a quantum election protocol. It satisfies all eight requirements stated earlier under the physical assumptions that there exists a one-way untappable channel from the administrator to the voter and that there is no collusion between any of the three parties in the protocol. Compared with current election protocols with verifiability and receipt-freeness, this protocol relies upon fewer physical assumptions. (iii) This construction is flexible and can be instantiated into an election scheme having post-quantum security by applying cryptographic algorithms conveying post-quantum security. Moreover, utilizing quantum digital signature and public key encryption yields a good result: the transmitted ballots are in quantum states, so owing to the no-cloning theorem, ballot privacy is less likely to be compromised, even if private keys of the signature and public key encryption are leaked after the election. However, in existing election protocols employing classic digital signatures and public key encryption, ballot privacy can be easily violated if attackers obtain private keys. Thus, our construction enhances privacy.

1. Introduction

We employ elections for various work- and life-related scenarios. There are several forms and numerous applications of elections, such as the elections of student cadres in universities, chairmen in companies, and presidents in countries. The aforementioned applications require strict and secure election protocols. Fair and reasonable election systems can accommodate the interests of all aspects of a society and reduce or prevent social conflicts and are conducive to the long-term stability of a society. Furthermore, security problems in elections may affect the stability of a group or even an entire society. Therefore, it is of great significance to study election protocols. With the rapid development of the information age, electronic elections are widely replacing traditional paper voting methods, as they are more in line with our everyday lives and work. Designing a secure and effective electronic election protocol is, at present, a prevalent research topic.

Electronic elections generally consist of three parties: the voters, administrator, and counter. The role of the voter is to forward his/her ballot anonymously to the counter, who counts the ballots and publishes the election results on a bulletin board; the administrator is responsible for helping the election run smoothly. Ever since the concept of elections was first proposed, cryptographers have been
committed to constructing a secure and practical election scheme. The first electronic election protocol was proposed by Chaum [1] in 1981, which was followed by numerous other classic electronic election protocols. These classic election protocols can be mainly divided into three categories: electronic election protocols based on hybrid networks [1–4], electronic election protocols based on homomorphic encryption [5–7], and electronic election protocols based on blind signatures [3, 8–10].

Furthermore, for the security of election protocols, Fujioka et al. [9] proposed that a voting scheme should satisfy all of the following seven requirements: completeness: all valid votes are counted correctly by the system when all parties in the protocol are honest; robustness: inappropriate behaviour of dishonest participants or nonparticipants cannot undermine the conduct of the election, and the system is fault-tolerant; privacy: the vote message of the ballot is secret, and only the corresponding voter and the counter can know it. In addition, no one, except the voters themselves, can associate it with the corresponding voter identity; secrecy: only legitimate voters can vote; unreusability: no voter can vote twice; fairness: during the voting process, the statistics of the votes cast shall not be announced, because the intermediate results of the voting would affect the voting tendency of voters who have not yet voted; verifiability: voters can finally verify whether their votes have been counted correctly. Subsequently, in response to the possibility of vote-buying and coercive vote fraud in the election protocol, Benaloh et al. [11] proposed the requirement of receipt–freeness, that is, the voters cannot prove to a third party the content of their vote. This requirement prevents voters from being bribed or forced to vote. Owing to the contradiction between being receipt-free and verifiable, it is difficult to construct an electronic election with the receipt-freeness property. Presently known receipt-free election schemes are based on the physical assumption that the attackers cannot monitor when voters are voting, because if attackers can monitor, then receipt-free voting cannot be realized. We believe that this assumption is necessary for the receipt-freeness of an election. Most receipt-free classic elections are based on the physical assumption that there is a one-way or two-way untappable channel. In addition, some protocols [4] require large amounts of zero-knowledge proof, which reduces the efficiency of the protocol, while some protocols require an anonymous channel [10] or randomizer [7], which increases the complexity of the protocol.

Most classic electronic election protocols are based on the difficulty assumptions of large integer decomposition or discrete logarithms. The development of quantum computers poses a considerable threat to the security of these protocols. Therefore, constructing quantum election protocols with the property of resisting quantum computer attacks has become a prevalent research topic. Current quantum election protocols are mainly classified into two types: entangled states-based protocols [12–20] and non-entangled states-based protocols [21–23]. Some entangled state-based protocols can only perform a binary vote for “yes” or “no” [13, 14, 16, 17], which is not suitable for scenarios with numerous candidates. Some nonentangled state-based protocols [22, 23] require numerous keys to be pre-shared among the participants. Furthermore, the current quantum election protocols cannot fulfill all eight electoral requirements mentioned above. A quantum election protocol that can resolve the contradiction between receipt-freeness and verifiability has not been developed yet.

To solve the abovementioned problems, considering the advantages of the physical properties of quantum states in the context of an election protocol, we propose the general construction of a quantum election protocol. Our construction is flexible and can thus be instantiated into an election scheme having post-quantum security by applying cryptographic algorithms, which possess post-quantum security properties. The proposed protocol is suitable for scenarios with numerous candidates and can simultaneously achieve all eight election protocol requirements, i.e., completeness, robustness, privacy, legality, unreusability, fairness, receipt-freeness, and verifiability. Our protocol requires fewer assumptions as compared with the classic election protocol.

1.1. Our Contributions. In this study, we utilise the proposed general construction of quantum digital signature [24] and quantum public key encryption [25], in conjunction with classic public key encryption, to develop a general construction of a quantum election protocol.

In our construction, we utilise public key cryptographic algorithms, so no pre-shared key is required for any two participants. The protocol can resist an attack from participants, so there is no need for a trusted third party. Anonymous channels are not required because ballots are delivered with the help of an administrator. The protocol is suitable for large-scale elections with numerous candidates and voters and can accommodate scenarios in which multiple voters vote simultaneously.

The protocol is the first to resolve the conflict between verifiability and receipt-freeness in a quantum election protocol, simultaneously achieving completeness, robustness, privacy, legality, unreusability, fairness, verifiability, and receipt-freeness under only two physical assumptions. The first assumption is that there exists a one-way untappable channel from the administrator to the voter. The second assumption is that there is no collusion between any of the three parties in the protocol. In the actual elections, the administrator and the counter are generally composed of multiple people representing different interested parties; thus, their mutual supervision makes the second assumption easy to implement.

We utilise quantum public key encryption [25] with information theory security, quantum digital signature [24] with post-quantum security, and classic public key encryption with post-quantum security to instantiate the proposed construction into election schemes with post-quantum security. In the existing election protocols with classic digital signature and public key encryption, the transmitted ballots are classic ciphertexts, and the attacker may intercept and copy the ciphertexts. Once the corresponding private keys are known in the future, the attacker
could then decrypt the ciphertexts, thus violating the privacy of the ballots. However, in our construction, the use of quantum digital signature and public key encryption yields a good result: the transmitted votes are delivered in the form of quantum states, which are unknown to the attacker. Therefore, according to the no-cloning theorem, the privacy of the ballots is not compromised even if the corresponding private keys are leaked after the election is completed. As an added benefit, private keys do not need to be kept secret after the election is complete. Furthermore, the keys of the quantum digital signature are classic; thus, classic public key infrastructure (PKI) can be used for key management and distribution.

1.2. Outline of the Paper. The remainder of this paper is organised as follows: Section 2 describes the basic knowledge and definitions of the cryptographic primitives used in the protocol, including public key encryptions and digital signatures, and presents two existing models for quantum public key encryption and quantum digital signatures. Section 3 describes the generic construction of the proposed quantum election. Section 4 analyses the security of the generic construction described in Section 3, including completeness, robustness, privacy, legality, unreusability, fairness, verifiability, and receipt-freeness. Section 5 instantiates the general construction of the election protocol into election schemes with post-quantum security by applying cryptographic algorithms with post-quantum security, analyses the efficiency of the instantiation, and compares the efficiency and security with current protocols. Section 6 summarises our work and presents directions for future work.

2. Preliminaries

For the remainder of this paper, we assume the reader is familiar with the basic notions and notionation of quantum computing. These can be found in textbooks such as [26].

Given \(|r\rangle\) and \(\sum_m a_m |m\rangle\) as input, quantum transformation \(U_f\) computing a function \(f: \{0, 1\}^r \times \{0, 1\}^n \rightarrow \{0, 1\}^r\) is defined as

\[
U_f (|r\rangle \sum_m a_m |m\rangle |0\rangle) = |r\rangle \sum_m a_m |m\rangle |0\oplus f (r, m)\rangle
\]

(1)

where \(\oplus\) denotes bitwise addition. In addition, when given \(|r\rangle\) and \(\sum_m a_m |m\rangle |f (r, m)\rangle\) as input, we can use unitary transformation again and get

\[
U_f (|r\rangle \sum_m a_m |m\rangle |f (r, m)\rangle) = |r\rangle \sum_m a_m |m\rangle |0\rangle
\]

(2)

Unitary transformation implemented via quantum circuits of \(U_f\) is shown in Figure 1.

2.1. Public Key Encryption. Public key encryption algorithms enable participants to transfer information securely without sharing secret key. A classic public key encryption \(PKE = (Gen, Enc, Dec)\) consists of three algorithms, a finite message space \(\mathcal{M}\), and a ciphertext space \(\mathcal{C}\). The key generation algorithm takes a security parameter \(\lambda\) as input and outputs a key pair \((pk_{pke}, sk_{pke})\). The encryption algorithm \(Enc\) takes public key \(pk_{pke}\) and a message \(m \in \mathcal{M}\) as input and outputs either a message \(c \in \mathcal{C}\). The deterministic decryption algorithm \(Dec\) takes \(sk_{pke}\) and a ciphertext \(c\) as input and outputs either a message \(m\) or a special symbol \(\bot\) to indicate that \(c\) is not a valid ciphertext. For correctness, we require that \(Pr[m = Dec(sk_{pke}, c)] = 1 - negl(\lambda)\), where \(negl(\lambda)\) denotes a negligible function. We say that \(PKE\) is deterministic if \(Enc\) is deterministic, while \(PKE\) is probabilistic if \(Enc\) is probabilistic.

2.1.1. Quantum Public Key Encryption. The difference between quantum and classic public key encryption is that several of the six elements, including public and private keys generated by the key generation algorithm, encryption algorithm, decryption algorithm, plaintext, and ciphertext, may be represented in quantum states.

In 2000, Okamoto [27] proposed the concept of quantum public key encryption. Today, the existing quantum public key encryption schemes can be roughly classified into several categories according to different problems that schemes are based on coding [25, 28–32], quantum algorithms [27, 33], indistinguishable quantum states [34–37], induced trapdoor one-way transformations [24], quantum bit rotation [38–41], and interaction between bell state particles [42–46]. In 2015, Wu et al. [47] classified quantum public key encryption into 64 types, according to whether the six elements of public key encryption belong to quantum space or not.

Liang et al. [36] gave a generic quantum public key encryption construction for encrypting classical messages, where a private key corresponds to an exponent of different public keys. Then, Yang et al. [25] slightly improved the abovementioned scheme by using two Boolean functions instead of using one Boolean function. Every public key in [25] is an unknown quantum state for anyone except its generator, so the ciphertext quantum state obtained by encrypting a plaintext is also unknown to its encrypting party. Moreover, every public key is in different quantum state. These may be useful to achieve the receipt-freeness of election protocol. We briefly recall their generic quantum public key encryption construction \(qPKE = (qGen, qEnc, qDec)\). Let \(p\) denote the number of messages to be encrypted, and \(m_i\) denote the \(i\)-th message for \(i \in \{1, 2, \ldots, p\}\).

(i) \(qGen\). (1) Randomly select two functions \(F_1, F_2\) from a set of polynomial computable functions \(\{F: \{0, 1\}^n \rightarrow \{0, 1\}^n\}\). (2) For \(i \in \{1, 2, \ldots, p\}\), randomly select and compute \(k_{i,1} = F_1 (g_i)\) and \(k_{i,2} = F_2 (g_i)\). Then, the quantum state \(|\psi_{k_{i,1}, k_{i,2}}\rangle\) can
be efficiently prepared according to \( k_{i,1} \) and \( k_{i,2} \). (3) Upload all \( (g_j, |\psi_{k_j,k_j}\rangle) \) to the public key register. Function \((F_1, F_2)\) is the private key. (4) In a word, 
\[
qGen(1^n) \rightarrow (pk = \{ pk_i \}_{i \in \{1, 2,..., p\}}, sk = (F_1, F_2)),
\]
where \( pk_i = \{ g_j, |\psi_{k_j,k_j}\rangle \} \).

(ii) \( qEnc \). (1) In order to encrypt the \( i \)-th message \( m_i \), download the \( i \)-th part of the classic and quantum public-key pair \( pk_i = \{ g_j, |\psi_{k_j,k_j}\rangle \} \) from the public key register. (2) Encrypt the classic plaintext \( m_i \) by computing \( |\psi_{\langle m_i \rangle} \rangle = \mathcal{S} (|\psi_{k_j,k_j}\rangle, m_i) \), where \( \mathcal{S} \) is a quantum encryption transformation, then output the ciphertext \( c = (g_j, |\psi_{\langle m_i \rangle} \rangle) \). (3) In a word, 
\[
qEnc (pk_i, m_i) \rightarrow c.
\]

(iii) \( qDec \). (1) Given a ciphertext \( c = (g_j, |\psi_{\langle m_i \rangle} \rangle) \), use private key \( F_1, F_2 \) to compute \( k_{i,1} = F_1(g_j), k_{i,2} = F_2(g_j) \). (2) According to the value of \( k_{i,1}, k_{i,2} \), use \( \mathcal{D} \) to decrypt \( |\psi_{\langle m_i \rangle} \rangle \) and obtain the message \( m_i = \mathcal{D} (k_{i,1}, k_{i,2}, |\psi_{\langle m_i \rangle} \rangle) \), where \( \mathcal{D} \) is a quantum decryption transformation. (3) In a word, 
\[
qDec (sk, c = (g_j, |\psi_{\langle m_i \rangle} \rangle)) \rightarrow m_i.
\]

2.2. Digital Signature. A classic digital signature, \( \text{Signature} = (\text{KeyGen}, \text{Sign}, \text{Verf}) \), consists of three algorithms and a finite message space \( \mathcal{M} \). The key generation algorithm takes a security parameter \( \lambda \) as input and outputs a key pair \( (vk_{sig}, sk_{sig}) \). The signature algorithm \( \text{Sign} \) takes the secret key \( sk_{sig} \) and a message \( m \in \mathcal{M} \) as input and outputs a signature \( \sigma \). The verification algorithm \( \text{Verf} \) inputs \( vk_{sig} \) and a pair \( (m, \sigma) \) and outputs a bit message either zero or one (this means that \( \sigma \) is a legal signature of message \( m \)). For correctness, we require that, pr \( [1 = \text{Verf}(vk_{sig}, (m, \sigma)) \mid \sigma = \text{Sign}(sk_{sig}, m)] = 1 - \text{negl}(\lambda) \) where \( \text{negl}(\lambda) \) denotes a negligible function.

2.2.1. Quantum Digital Signature. The difference between quantum and classic digital signature is that several of the six elements may be represented in quantum states. These elements include the signature key and the verification key generated by the key generation algorithm, the signature algorithm, the verification algorithm, the plaintext, and the signature results of the quantum digital signature. Because of the special properties of quantum states, it is difficult to achieve a true quantum digital signature, and most existing quantum digital signature schemes require the participation of arbitrators. Quantum digital signatures, such as classic digital signatures, achieve both the identity authentication of signer and the integrity verification of message.

With the development of quantum computer and quantum communication, it is necessary to study quantum digital signature. In 2001, Gottesman and Chuang [48] firstly proposed a quantum digital signature scheme for signing classic message. Subsequently, some arbitrated quantum digital signature protocols requiring the participation of trusted third parties [49–54] were proposed in succession. In 2010, Yang et al. [24] proposed an interactive quantum digital signature protocol for signing quantum message based on induced trapdoor one-way transformations. This protocol exploits the nature of quantum entangled states, realising the identity authentication of signer and protecting the integrity of messages without the participation of arbitrators. Subsequently, several quantum digital signature protocols without quantum memory were proposed [55, 56], thus improving the practicability of quantum signatures.

A general construction of quantum digital signature for signing quantum message is proposed by Yang et al. [24]. The signature scheme \( q\text{Signature} \) consists of three algorithms \((q\text{KGen}, q\text{Sign}, q\text{Verf})\) which are described as follows:

(i) \( q\text{KGen} \). Randomly select a trapdoor one-way function \( f: \{0, 1\}^n \rightarrow \{0, 1\}^{n-f} \) which has a trapdoor \( f^{-1} \). Then, \( f \) is the verification key and \( f^{-1} \) is the signature secret key. In a word, 
\[
q\text{KGen}(1^n, 1^t, 1^{t'}, 1^{t''}) \rightarrow (vk_{sig} = f, sk_{sig} = f^{-1}).
\]

(ii) \( q\text{Sign} \). The signer signs a \( n \) bits quantum message \( |\psi\rangle = \sum_m a_m |m\rangle \) to the verifier as follows: (1) the verifier randomly generates a number \( r_{ver} \in \{0, 1\}^t \) and sends it to the signer. (2) The signer randomly generates a number \( r_{sig} \in \{0, 1\}^n \) and computes \( f^{-1}(r_{ver}, r_{sig}) = (r', r'') \), where \( r \in \{0, 1\}^t \) and \( r' \in \{0, 1\}^n \). (3) Then, with \( r \), the signer performs the quantum transformation \( U_f \) on the quantum message \( |\psi\rangle|0\rangle = \sum_m a_m |m\rangle|0\rangle \) and obtains

\[
\sum_m a_m |m\rangle|0\rangle \xrightarrow{(1)} \sum_m a_m |m\rangle|f(r, m)\rangle,
\]
and sends quantum state \( \sum_m a_m |m\rangle|f(r, m)\rangle \) to the verifier. (4) In a word, \( q\text{Sign}(sk_{sig}, |\psi\rangle) \rightarrow \sigma \), where
\[ \sigma = \left( r_{\text{ver}}, r', \sum_{m} a_{m} |m \rangle | f (r, m) \rangle \right). \quad (4) \]

(iii) \( q\text{Ver} f \). (1) The verifier tells the signers that they have received the quantum state. (2) The signer announces \( r \) and \( r' \). (3) The verifier computes \( f (r, r') \) and checks whether the first \( t' \) bits of \( f (r, r') \) equal \( r_{\text{ver}} \). They then perform the transformation,

\[ \sum_{m} a_{m} |m \rangle | f (r, m) \rangle \xrightarrow{2} \sum_{m} a_{m} |m \rangle |0 \rangle, \quad (5) \]

and measure the second quantum register. They accept the signature if and only if the second register is in state \( |0 \rangle \). In a word, \( q\text{Ver} f (vk_{\text{sign}}, (\psi, \sigma)) \rightarrow 1 \) if and only if the second register is in state \( |0 \rangle \).

3. Generic Construction of Quantum Election

In this section, we present a generic construction of quantum election protocol. Section 3.1 gives an overview of the construction and Section 3.2 gives a more detailed description.

3.1. Overview of Our Construction. The proposed election protocol runs as follows. (1) A voter holds his/her vote message and performs quantum public key encryption on it by using the counter’s public key, obtaining the quantum ciphertext which is the ballot of the voter. (2) The ballot is signed by the voter and sent to the administrator. Then, the voter’s signature is verified by the administrator. (3) The ballot is signed by the administrator and sent to the counter. Then, the administrator’s signature is verified by the counter. (4) The ballot is decrypted by using the counter’s private key and verification information is generated by the counter. (5) The information is sent to the voter with the help of the administrator. Finally, the voter can verify whether his/her vote is counted correctly or not.

3.2. Our Concrete Construction. In this section, we give a detailed description of quantum election protocol which consists of four cryptography primitives: classic probabilistic public key encryption \( p\text{PKE} = (p\text{Gen}, p\text{Enc}, p\text{Dec}) \), classic deterministic public key encryption \( d\text{PKE} = (d\text{Gen}, d\text{Enc}, d\text{Dec}) \), quantum digital signature \( q\text{Signature} = (q\text{KGen}, q\text{Sign}, q\text{Ver} f) \), and quantum public key encryption \( q\text{PKE} = (q\text{Gen}, q\text{Enc}, q\text{Dec}) \), where quantum public key encryption and quantum digital signature are two generic constructions proposed by Yang et al. [25] and Yang et al. [24], respectively. Please refer to Sections 2.1 and 2.2 for more details.

3.2.1. Initialization. Let \( p \) denote the number of voters, and \( v_i \) denote the \( i \)-th voter for \( i \in \{1, 2, \ldots, p\} \). Let \( q \) denote the number of legitimate candidates, and \( A_j \in \{0, 1\}^{\alpha_j} \) represent the identity of the \( j \)-th legitimate candidate for each \( j \in \{1, 2, \ldots, q\} \). Moreover, let \( \text{Admin} \) represent the administrator and \( \text{Counter} \) represent the counter. In the protocol, each voter has a unique identity string denoted by \( id_i \), while \( \text{Admin} \) has an identity \( id_{\text{admin}} \). Identity strings are publicly known. The participants carry on the following steps:

(i) Admin. (1) \( \text{Admin} \) runs the algorithm \( q\text{KGen} \) of \( q\text{signature} \) and gets a pair of public and private keys \( (f_{\text{admin}}^{-1}, f_{\text{admin}})^{-1} \). (2) \( \text{Admin} \) runs the algorithm \( p\text{Gen} \) of \( p\text{PKE} \) and gets a pair of public and private keys \( (p_{\text{admin}}, sk_{\text{admin}}) \). (3) \( \text{Admin} \) runs the algorithm \( d\text{Gen} \) of \( d\text{PKE} \) and gets a pair of public and private keys \( (p_{\text{admin}}, sk_{\text{admin}}) \).

(ii) Counter. (1) In order to encrypt \( p \) messages, \( \text{Counter} \) runs the algorithm \( q\text{Gen} \) of \( q\text{PKE} \) and gets a pair of public and private keys \( (pk_{\text{Counter}}, sk_{\text{Counter}}) \), where \( pk_{\text{Counter}} = \{g_j|\psi_{k_{j, 1}, k_{j, 2}}\rangle\}_{j=1, 2, \ldots, p} \) and \( sk_{\text{Counter}} = \{F_1, F_2\} \).

(iii) \( v_i \). \( v_i \) runs the algorithm \( q\text{KGen} \) of \( q\text{signature} \) and gets a pair of public and private keys \( (f_{v_i}, f_{v_i}^{-1}) \).

3.2.2. Election

(1) Preprocessing:

Each voter randomly selects a public key from the quantum public key register. Taking the public key \((g_j, |\psi_{k_{j, 1}, k_{j, 2}}\rangle)\) chosen by voter \( v_i \) as an example and assuming that \( v_i \) wants to vote for candidate \( A_j \), \( v_i \) carries on the following steps:

(i) \( v_i \) downloads one classical and quantum public-key pair \((g_j, |\psi_{k_{j, 1}, k_{j, 2}}\rangle)\) of \( \text{Counter} \) from the public key register.

(ii) \( v_i \) generates a random string \( e_j \in \{0, 1\}^\alpha \), and selects a candidate \( A_j \in \{0, 1\}^\alpha \) to make up a vote message \( T_{ji} \), where \( T_{ji} = A_j \parallel e_j \in \{0, 1\}^\alpha \) represents a simple concatenation of \( A_j \) and \( e_j \).

(iii) By performing quantum encryption transformation \( q\text{Enc} \) on the vote message \( T_{ji} \), \( v_i \) obtains ciphertext \((g_j, |\psi_{k_{j, 1}, k_{j, 2}}\rangle) = q\text{Enc}(|\psi_{k_{j, 1}, k_{j, 2}}\rangle, T_{ji}) \), \(|\psi_{k_{j, 1}, k_{j, 2}}\rangle\) is the quantum ballot and it is clear that it is an \( n \)-bit quantum superposition state which can be denoted as \( \sum_{m} a_{m} |m \rangle \).

(2) Ballot-casting from voter to administrator:

This process is briefly illustrated in Figure 2. For the purpose of authenticating the identity of \( v_i \) and the integrity of ballot, the administrator \( \text{Admin} \) needs to mutually communicate with \( v_i \) so that \( \text{Admin} \) can verify the correctness of the received signature sent by \( v_i \).

(i) If \( v_i \) wants to vote, they will send their \( id_i \) to \( \text{Admin} \).

(ii) Let \( T \) denote the number of \( id_i \) received by \( \text{Admin} \) at the same time. After receiving \( id_i \), \( \text{Admin} \) searches the local data set \( ID \); if \( id_i \in ID \) or \( T > 1 \), the request for voting is rejected. Otherwise, \( \text{Admin} \) randomly generates \( r_{ad_i} \in \{0, 1\}^\alpha \) and sends it to \( v_i \).

(iii) \( v_i \) randomly generates a number \( r_{v_i} \in \{0, 1\}^\alpha \) and computes \( f_{v_i}(r_{ad_i}, r_{v_i}) = (r_{v_1}, r_{v_2}) \), where \( r_{v_1} \in \{0, 1\}^\alpha \) and \( r_{v_2} \in \{0, 1\}^\alpha \).
(iv) With $r_{i,1}$, $v_i$ performs the quantum transformation $U_{f_{v_i}}$ on $|\psi_{i,1}\rangle = |\psi_{k_{i,1},k_{i,2}}\rangle \langle 0|$ and obtains $|\psi_{i,2}\rangle$, where

$$|\psi_{i,1}\rangle = \sum_{m} \alpha_{m} |m\rangle \langle 0| \xrightarrow{U_{f_{v_i}}} |\psi_{i,2}\rangle = \sum_{m} \alpha_{m} |m\rangle f_{v_i}(r_{i,1}, m).$$

(6)

In a word, $q\text{Sign}(f_{v_i}, |\psi_{k_{i,1},k_{i,2}}\rangle) \rightarrow \sigma$, and it is easy to see that $\sigma = (r_{ad,i}, r_{i,1}, r_{i,1}', |\psi_{i,2}\rangle)$.

(v) By running the encryption algorithm $p\text{Enc}$, $v_i$ obtains its ciphertext $c_{v_i} \rightarrow ad = p\text{Enc}(pk_{ad}, (r_{i,1}, r_{i,1}', g_{i}))$. Then, $v_i$ sends $(c_{v_i} \rightarrow ad, |\psi_{i,2}\rangle, id_i)$ to Admin.

(vi) After receiving $(c_{v_i} \rightarrow ad, |\psi_{i,2}\rangle, id_i)$ from $v_i$, Admin decrypts $c_{v_i} \rightarrow ad$ with its private key $sk_{ad}$ and obtains the plaintext $(r_{i,1}, r_{i,1}', g_{i})$. As a result, Admin can get the signature $\sigma = (r_{ad,i}, r_{i,1}, r_{i,1}', |\psi_{i,2}\rangle)$ of $(|\psi_{k_{i,1},k_{i,2}}\rangle, \sigma)$. If the output of $q\text{Verf}(f_{v_i}, (|\psi_{k_{i,1},k_{i,2}}\rangle, \sigma))$ is equal to 1, Admin stores $v_i$'s information $(id_i, g_{i})$ into their database. Note that in the computing process of $q\text{Verf}$, there exists the following operation to obtain quantum state:

$$|\psi_{i,2}\rangle = \sum_{m} \alpha_{m} |m\rangle f_{v_i}(r_{i,1}, m) \xrightarrow{U_{f_{v_i}}} |\psi_{i,1}\rangle = \sum_{m} \alpha_{m} |m\rangle |0\rangle.$$

(7)

When all voters have completed this stage with Admin, the protocol proceeds to the next stage.

(3) Ballot-casting from administrator to counter:

This process is briefly illustrated in Figure 3. For the purpose of authenticating the identity of Admin and the integrity of ballot, the Counter needs to mutually communicate with Admin so that Counter can verify the correctness of the received signature sent by Admin.

(i) If Admin wants to send $v_i$'s ballot to Counter, they firstly send $id_{ad}$ to Counter.

(ii) After receiving $id_{ad}$, Counter randomly generates $r_{ad,i} \in \{0, 1\}^n$ and sends it to Admin.

(iii) randomly generates $r_{ad,i} \in \{0, 1\}^n$ and computes $f_{ad}(r_{ad,i}) = (r_{ad,i}, r_{ad,i}^{'}, |\psi_{i,3}\rangle)$, where $r_{ad,i} \in \{0, 1\}^n$ and $r_{ad,i}^{'}, |\psi_{i,3}\rangle \in \{0, 1\}^n$.

(iv) With $r_{ad,i}$, Admin performs the quantum transformation $U_{f_{ad}}$ on $|\psi_{i,1}\rangle$ and obtains $|\psi_{i,3}\rangle$, where

$$|\psi_{i,1}\rangle = \sum_{m} \alpha_{m} |m\rangle \langle 0| \xrightarrow{U_{f_{ad}}} |\psi_{i,3}\rangle = \sum_{m} \alpha_{m} |m\rangle f_{ad}(r_{i,2}, m).$$

(8)

In a word, $q\text{Sign}(f_{v_i}, |\psi_{i,1}\rangle) \rightarrow \sigma_{ad}$, and it is easy to see that $\sigma_{ad} = (r_{ad,i}, r_{ad,i}^{'}, |\psi_{i,3}\rangle)$. 

(v) Admin sends $(\sigma_{ad}, g_{i})$ to Counter.

(vi) If the output of $q\text{Verf}(f_{ad}, (|\psi_{i,2}\rangle, \sigma_{ad}))$ equals 1, Counter can make sure the identity of Admin. Note that in the process of the above computing $q\text{Verf}$, there exists the following operation to obtain quantum state $|\psi_{i,3}\rangle$,

$$|\psi_{i,3}\rangle = \sum_{m} \alpha_{m} |m\rangle f_{ad}(r_{i,2}, m) \xrightarrow{U_{f_{ad}}} |\psi_{i,1}\rangle = \sum_{m} \alpha_{m} |m\rangle |0\rangle.$$

(9)

(vii) With the quantum state $|\psi_{i,1}\rangle = \sum_{m} \alpha_{m} |m\rangle |0\rangle$, Counter can obtain voter $v_i$'s ballot $|\psi_{k_{i,1},k_{i,2}}\rangle = \sum_{m} \alpha_{m} |m\rangle$ and can get the ciphertext $c = (g_{i}, |\psi_{k_{i,1},k_{i,2}}\rangle)$. 

Figure 2: Ballot-casting from voter to administrator.
After Counter receives and verifies all voters’ ballots from Admin, the election enters the next stage.

(4) Ballot-counting and result-publishing:

Given the quantum ciphertexts \( c = (g_i, |\psi_{i, ki}^T\rangle) \), Counter decrypts it and publishes the election result.

(i) Counter decrypts the ciphertext \( c \) by using \( \text{qDec} \), i.e., \( \text{qDec}(sk_i, c) = T_{ji} = A_j \| e_i \| e_i' \).

(ii) If the pre-(\( n - s \)) bits of the decryption result do not represent a qualified candidate, Counter rejects the ballot. Otherwise, Counter accepts the decryption result \( T_{ji} \) and corresponding \( g_i \).

(iii) After accepting all the decryption results from \( p \) voters, Counter computes the voting result of every voter \( v_i \) in the form of \( A_i \| (e_i \oplus \text{dEnc}(pk_{\text{ab}}, f_j)) \), where \( f_j \in \{0, 1\}^t \) is generated randomly by Counter. Then, Counter collects all the voting results and publishes them on the bulletin board.

(5) Confirming vote and terminating the election:

Counter sends message to \( v_i \) with the help of Admin so that \( v_i \) can verify whether their vote is counted correctly. Under the physical assumption that there is a one-way noneavesdropping channel from the administrator to the voter, the election satisfies the requirement of receipt – freeness.

(i) By running the encryption algorithm \( \text{dEnc} \), Counter obtains \( c_{\text{te} \rightarrow \text{ab}, ji} = \text{dEnc}(pk_{\text{ab}}' (e_i \oplus f_j)) \) and sends \( (c_{\text{te} \rightarrow \text{ab}, ji}, g_i) \) to Admin.

(ii) Admin utilises \( g_i \) as a label to find the corresponding \( v_i \) and sends \( c_{\text{te} \rightarrow \text{ab}, ji} \) to \( v_i \) by the one-way untappable channel from Admin to \( v_i \).

(iii) When all voters receive the corresponding ciphertext, Admin announces their private key \( sk_{\text{ab}}' \). Then, \( v_i \) obtains \( e_i \oplus f_j = \text{dDec}(sk_{\text{ab}}', c_{\text{te} \rightarrow \text{ab}, ji}) \). Finally, \( v_i \) gets \( f_j \) with the knowledge of \( e_i \). With \( e_i \) and \( f_j \), \( v_i \) can compute and obtain \( A_j \| (e_i \oplus dEnc(pk_{\text{ab}}', f_j)) \). Then, \( v_i \) can verify whether they have successfully voted. If \( v_i \) does not succeed in the verification, they can request a re-election. If all voters can make sure that their votes have been successfully voted, the election is complete and terminated.

4. Security Analysis

We assume that there exists no collusion between any two of the three parties in the protocol. In actual elections, the administrator and the counter are generally composed of multiple people representing different interest groups, so their mutual supervision makes this assumption easy to hold. Both participants and nonparticipants can potentially attack the protocol. We define an attack as participants as an internal attack and an attack from nonparticipants as an external attack. In this section, we prove that our proposed election protocol satisfies the eight requirements, namely, completeness, robustness, privacy, legality, unreusability, fairness, verifiability, and receipt-freeness.

4.1. Completeness

Theorem 1 (completeness). In the protocol, all valid votes are counted correctly when all participants are honest.

Proof. (sketch). According to the definition of completeness, all valid votes are counted correctly when all parties in the protocol are honest. A qualified voter \( v_i \) for \( i \in \{1, 2, \ldots, p\} \) encrypts the vote message \( A_j \| e_i \) which is a simple concatenation of binary string \( T_{ji} \), representing a legal candidate and binary string \( e_i \) generated randomly by \( v_i \), with a classic and quantum public-key pair \( (g_i, |\psi_{ki, ki}^T\rangle) \) which is selected randomly by \( v_i \) from Counter’s public key pool. After the ballot \( |\psi_{ki, ki}^T\rangle \) is signed and verified twice, Counter finally obtains the ciphertext \( (g_i, |\psi_{ki, ki}^T\rangle) \). Then, Counter can obtain the vote message of \( v_i \)’s ballot by decrypting the ciphertext with their private key \( (F_1, F_2) \).
Hence, it is clear that the proposed protocol satisfies completeness.

4.2. Robustness

**Theorem 2** (robustness). The inappropriate behaviour of dishonest participants or nonparticipants cannot disrupt the election; i.e., the protocol is fault-tolerant.

**Proof.** We analyse the robustness of the protocol by considering the inappropriate behaviour of participants including voters, administrator and counter, and nonparticipants.

(i) When a dishonest voter \( v_j \) seeks to disrupt the election, they may have three strategies. In the first case, \( v_j \) does not vote, that is, in the ballot-casting from voter to administrator stage, \( v_j \) does not send \( id_j \) or \( (e_{v_j} \rightarrow \text{ad} \mid \psi_{j,2}, \text{id}_j) \) to Admin. However, it is clear that Admin will be aware of this missing vote. In the second case, \( v_j \) sends an invalid message group \( (e_{v_j} \rightarrow \text{ad} \mid \psi_{j,2}, \text{id}_j)_{\text{false}} \) to Admin in the ballot-casting from voter to administrator stage. While the running result of \( q\text{Verf} \) achieved by Admin will not equal 1, this wrong action will be detected by. In the third case, \( v_j \) generates a random binary string \( A_{\text{false}} \), which does not represent a qualified candidate and encrypts \( A_{\text{false}} \| e_j \) with Counter’s public key \( (g_j, \psi_{k_1, k_2}) \). However, Counter will find out the illegal \( A_{\text{false}} \) in the ballot-counting and result-publishing stage and reject the ballot. In a word, the voter \( v_j \) cannot succeed in disrupting the election.

(ii) If the internal attacker Admin wants to disrupt the election, they may have two strategies. In the first case, Admin tampers with the quantum state \( \mid \psi_{1,1} \rangle \) from a legitimate voter \( v_i \) in the ballot-casting from administrator to counter stage. However, because Admin does not know the candidate that \( v_i \) is going to vote for, their random tampering will result in a random decrypting of the result obtained by Counter in the ballot-counting and result-publishing stage. Then, Counter will reject the ballot with a high probability. Because the probability that Admin successfully guesses the candidate that is voted for by \( v_i \) is \( 1/q \), the probability that they correctly tamper with the quantum state is also \( 1/q \). Then, the probability that the decrypted result obtained by Counter represents a legitimate candidate is also \( 1/q \), so the probability that the ballot is rejected in the ballot-counting and result-publishing stage by Counter is \( (q-1)/q \). In addition, even if the tampered ballot is accepted by Counter, \( v_i \) can be aware that their vote is not counted correctly in the confirming vote and terminating the election stage. Then, \( v_i \) can request a re-election. In the second case, Admin may substitute the quantum state \( \mid \psi_{1,1} \rangle \). In this manner, Counter will accept the decrypted result, but \( v_i \) will find that their vote has not been counted correctly during the confirming vote and terminating the election stage, and they can request a re-election.

(iii) If Counter wants to destroy the election, they may modify the vote message of the accepted ballot and then publish the modified result in the ballot-counting and result-publishing stage. Suppose that Counter tampers with the vote message \( E_j \| e_j \) of eligible voter \( v_j \) to \( E_j \| \epsilon_j \). To be unnoticed by \( v_j \), Counter must send a \( f_{\text{false}} \) so that \( v_j \) can verify their vote successfully. To achieve that, Counter can find another vote message, such as \( E_j \| e_j \), which is chosen by another legal voter \( v_i \) and make \( f_{\text{false}} \) satisfy \( e_{v_i} \oplus d\text{Enc} (\text{pk}_{\text{ad}}, f_{\text{false}}) = e_{v_j} \oplus d\text{Enc} (\text{pk}_{\text{ad}}, f_{\text{false}}) \). However, because Counter does not have Admin’s private key \( k_{\text{ad}}' \), it is difficult for Counter to provide a \( f_{\text{false}} \) that can make \( v_j \) verify their vote message on the bulletin board successfully. Therefore, the misbehaviour of Counter cannot be successful if the classic deterministic public key encryption \( \text{dPKE} \) is secure.

(iv) If external attackers want to tamper with the ballot from a legitimate voter \( v_i \), they can only attack during the transmission of \( (e_{v_i} \rightarrow \text{ad} \mid \psi_{i,2}, \text{id}_i) \) in the ballot-casting from voter to administrator stage or during the transmission of \( (\text{ad}, g_j) \) in the ballot-casting from administrator to counter stage. However, the security of the quantum signature algorithm \( q\text{Signature} \) enables this attack to be detected by Admin or Counter.

Hence, the protocol is robust.

4.3. Privacy

**Theorem 3** (privacy). In the protocol, the vote message of the ballot is secret, and only the corresponding voter and the counter can know it. In addition, no one except the voters themselves can associate it with the corresponding voter identity.

**Proof.** Nonparticipants of the protocol, the administrator, and the counter are all likely to attack the privacy of ballots, so we analyse the three cases in turn. Meanwhile, to prove our election can achieve privacy, we use the no-cloning theorem [57], which states that an exact copy of an unknown quantum state is impossible to achieve in quantum mechanics.

(i) Resistance to attacks from nonparticipants: if a nonparticipant of the protocol wants to attack the privacy of \( v_i \)'s ballot, the only possible opportunity occurs during the transmission of information in the ballot-casting from voter to administrator stage and ballot-casting from administrator to counter stage. Assuming that the attacker intercepts \( (e_{v_i} \rightarrow \text{ad} \mid \psi_{i,2}, \text{id}_i) \) in the ballot-casting from voter to administrator stage, the security of \( \text{PKE} \) ensures that the attackers cannot decrypt \( e_{v_i} \rightarrow \text{ad} \) to obtain \( (r_{i,1}, r_{i,1}', g_j) \) and cannot obtain the signature \( \sigma = \).
(r_{ad}, r_{id}, r'_{id}, |ψ_{i,2}⟩) of |ψ_{T,i}⟩. Then, the attacker cannot untagle quantum state |ψ_{i,2}⟩ to obtain |ψ_{i,1}⟩ and the ciphertext c = (g_{i}, |ψ_{k_{i},k_{2}}⟩, ) , let alone obtain the vote message of the ballot, because they do not know Counter’s private key (F_{1}, F_{2}). Moreover, if the transmitted ballot is classic, the attacker may copy it during its transmission. Once the corresponding private keys are known in the future, the attacker could then violate the privacy of the ballot. However, in our construction, the ballot is transmitted in the form of a quantum state, and |ψ_{i,2}⟩ is unknown to the attacker. Owing to the no-cloning theorem, after the election is completed, even if the private keys (F_{1}, F_{2}) and sk_{ad} are leaked, the privacy of the ballots is not compromised. If the attacker intercepts (a_{ad}, g_{i}) in the ballot-casting from administrator to counter stage, they can untagle |ψ_{i,3}⟩ with r_{i,2} to obtain |ψ_{i,1}⟩ and the ciphertext c = (g_{i}, |ψ_{k_{i},k_{2}}⟩, ). However, they cannot decrypt the ciphertext to obtain v_{i}'s vote message, because the attacker does not know Counter’s private key (F_{1}, F_{2}). In this case, |ψ_{i,1}⟩ is unknown to the attacker. Thus, the attacker cannot copy this state and obtain the vote message of the ballot even if the private key (F_{1}, F_{2}) is leaked after the election.

(ii) Resistance to an attack from Admin: Admin can obtain the ciphertext c = (g_{i}, |ψ_{k_{i},k_{2}}⟩) in the stage of ballot-casting from voter to administrator, but the security of qPKE ensures that they cannot obtain the vote message of the ballot, because Counter’s private key (F_{1}, F_{2}) is unknown. Similar to the earlier analysis, c = (g_{i}, |ψ_{k_{i},k_{2}}⟩) is unknown to Admin, so they cannot copy the ciphertext c. Going further, they cannot obtain the vote message of the ballot even if the private key (F_{1}, F_{2}) is leaked after the election.

(iii) Resistance to an attack from Counter: if Counter wants to attack the privacy of v_{i}'s ballot, they may attack in the stages of ballot-casting from voter to administrator and ballot-casting from administrator to counter. In the stage of ballot-casting from voter to administrator, Counter may intercept (c_{v}, a_{ad}, id_{v}) during the transformation and attempt to obtain v_{i}'s vote message via the intercepted information. However, similar to this type of attack from nonparticipants, the security of pPKE ensures that Counter cannot obtain (r_{i,1}, r'_{i,1}, g_{i}) and obtain the signature σ = (r_{ad}, r_{id}, r'_{id}, |ψ_{i,2}⟩) of |ψ_{k_{i},k_{2}}⟩. Thus, Counter cannot untagle quantum state |ψ_{i,2}⟩ to obtain |ψ_{i,1}⟩ and the ciphertext c = (g_{i}, |ψ_{k_{i},k_{2}}⟩, ). Therefore, it is impossible for Counter to use the private key (F_{1}, F_{2}) to decrypt the ciphertext to obtain v_{i}'s vote message. In this stage, |ψ_{i,2}⟩ is unknown to Counter; thus, they cannot copy it and associate v_{i}'s ballot content with v_{i}'s identity even if Admin’s private key sk_{ad} is leaked after the election. Moreover, it is worth noting that Counter cannot utilise g_{i}, which is obtained in the stage of ballot-casting from administrator to counter, as a label of v_{i}'s identity to associate v_{i}'s vote message with v_{i}'s identity because g_{i} is encrypted by pPKE in the stage of ballot-casting from voter to administrator. In the stage of ballot-casting from administrator to counter, even though Counter can finally obtain v_{i}'s vote message, they do not know which voter the decrypted ballot comes from because Admin replaces the vote identity id_{v} at this stage.

As a result, the protocol can satisfy the privacy requirement during an election, and the privacy will not be threatened even if corresponding private keys are leaked after the election.

4.4. Legality

Theorem 4 (legality). In the protocol, only legitimate voters can vote.

Proof. (sketch). The security of qSignature ensures that only the vote cast by a legitimate voter v_{i} for i ∈ [1, 2, ..., p] can be accepted. In the stage of ballot-casting from voter to administrator, v_{i} performs the quantum algorithm to sign their ballot |ψ_{k_{i},k_{2}}⟩ with their private key f^{-1}_{v_{i}}, i.e., qSignature (f^{-1}_{v_{i}}, |ψ_{k_{i},k_{2}}⟩) —⇒ σ. Then, Admin verifies the signature with v_{i}'s public key f_{v_{i}}, i.e., qVerify (f_{v_{i}}, (|ψ_{k_{i},k_{2}}⟩, σ)). The Admin accepts the information from v_{i} only if the output of qVerify equals 1, which ensures that Admin only accepts the ballot from legitimate voter v_{i}. Similarly, Counter only accepts the ballot from Admin. Therefore, only legitimate voters can vote successfully.

4.5. Unreusability

Theorem 5 (unreusability). In the protocol, no voter can vote twice.

Proof. (sketch). In the stage of ballot-casting from voter to administrator, if the legal voter v_{i} wants to vote, they send id_{v} to Admin at first. The condition that id_{v} ≠ 1D and T ≤ 1 ensures that Admin accepts id_{v} only once, so v_{i} can vote only once. Furthermore, in the case where voter v_{j} pretends to be another legitimate voter v_{i} and sends id_{v_{j}} to Admin, the illegal ballot from v_{j} will not be accepted by Admin because v_{j} has no private key f_{v_{j}} and the security of qSignature ensures that the output of quantum verification algorithm qVerify is not 1. Thus, one legal voter can vote only once.

4.6. Fairness

Theorem 6 (fairness). In the protocol, the intermediate results of the voting do not affect the preference of those who have not yet voted.
Proof. (sketch). After accepting all the vote messages from \(q\) voters, Counter collects the voting result of every voter and then publishes all results on the bulletin board in the stage of ballot-counting and result-publishing. Therefore, no one can obtain the intermediate results of the election, and the protocol is fair. \(\square\)

4.7. Verifiability

**Theorem 7** (verifiability). In the protocol, voters can finally verify whether their votes have been counted correctly.

Proof. (sketch). Counter announces the voting result of every voter \(v_i\) in the form of \(A_j|\{(e_i \oplus dEnc(pk^*_{ab}, f_i))\}\). Subsequently, running the encryption algorithm \(dEnc\), Counter obtains \(c_{e_i} = dEnc(pk^*_{ab}, e_i \oplus f_i)\); then they send \(c_{e_i}\) to \(v_i\) with the help of Admin. After \(v_i\) receives it, Admin publishes their private key \(sk^*_{ad}\). Running the decryption algorithm \(dDec\), \(v_i\) obtains \(e_i \oplus f_i = dDec(sk^*_{ad}, c_{e_i})\). \(v_i\) obtains \(f_i\) with the knowledge of \(e_i\). With \(e_i, f_i\), and \(pk^*_{ad}\), \(v_i\) can compute \(e_i \oplus dEnc(pk^*_{ab}, f_i)\). By searching for \(e_i \oplus dEnc(pk^*_{ab}, f_i)\) on the bulletin board and verifying whether the candidate information corresponding to \(e_i \oplus dEnc(pk^*_{ab}, f_i)\) is \(A_j\), \(v_i\) can verify whether their vote has been counted correctly. \(\square\)

4.8. Receipt-Freeness

**Theorem 8** (receipt-freeness). In the protocol, the voter cannot prove to a third party the content of their vote.

Proof. The protocol can achieve receipt-freeness under the physical assumption that there is a one-way unattapable channel from the authority (i.e., the administrator) to the voter [4, 58]. By a standard exclusive-OR trick, this assumption can be implemented by having a number of one-way channels, assuming that the adversary cannot simultaneously tap every one of them [4].

Suppose that voter \(v_i\) wants to vote for candidate \(A_j\), while there is a briber who requires \(v_i\) to vote for candidate \(A_n\). In this protocol, under the premise that the number of ballots for candidate \(A_n\) displayed on the bulletin board does not affect the judgment of the briber (for example, the number of votes for candidate \(A_n\) is zero, and this case can be ignored in a large-scale election), and \(v_i\) can vote for candidate according to their preferences, but lie to the briber successfully by providing the briber with false evidence to prove that they voted for candidate \(A_n\).

In the pre-voting stage, the briber may have asked voter to provide \(e_i\) as evidence for future verification. In this stage, \(v_i\) performs quantum encryption transformation \(qEnc\) on \(T_{ji}\), obtaining the ciphertext \((g_t, |\psi_{k_i,k_{i',2}}\rangle) = qEnc\).

\((|\psi_{k_i,k_{i',2}}\rangle, T_{ji})\). In \(v_i\’s\ view, the density operator of the quantum public key \(|\psi_{k_i,k_{i',2}}\rangle\) is

\[
\rho_{k_{i'}} = \text{Pr}[F_1, F_2] \sum_{y_1, y_2} \langle y_1 | (T_{ji}^{|\psi_{k_i,k_{i',2}}\rangle}) | y_2 \rangle = \frac{1}{2^n} \sum_{h_1, h_2} \langle \psi_{k_i,k_{i',2}} || T_{ji}^{|\psi_{k_i,k_{i',2}}\rangle} \rangle
\]

4.8. Receipt-Freeness

That is, in \(v_i\’s\ view, the quantum public key \(|\psi_{k_i,k_{i',2}}\rangle\) is in the maximum mixed state. Then, the quantum ciphertext \(|\psi_{k_i,k_{i',2}}\rangle\) is also in the maximum mixed state for \(v_i\). So, even if \(v_i\) sells all known information to the briber, the briber cannot obtain any information about the vote message of \(v_i\) by eavesdropping \((c_{e_i} \rightarrow ad\langle |\psi_{k_i,k_{i',2}}\rangle, id\rangle)\) on the channel from \(v_i\) to Admin in the ballot-casting from voter to administrator stage. Therefore, the step of computing the ciphertext, i.e., \((g_t, |\psi_{k_i,k_{i',2}}\rangle) = qEnc(|\psi_{k_i,k_{i',2}}\rangle, T_{ji})\), avoids the requirement that the channel from the \(v_i\) to Admin is unattapable.

In the stage of confirming vote and terminating the election, \(v_i\) receives \(c_{e_i} = dEnc(pk^*_{ab}, f_i)\) by the one-way untappable channel from Admin and obtains \(e_i \oplus f_i = dEnc(sk^*_{ad}, c_{e_i})\) by performing \(dDec\). Finally, \(v_i\) obtains \(f_i\) with the knowledge of \(e_i\). Now, we show that \(v_i\) cannot prove their vote to the briber according to \(f_i\). First, search for a vote for candidate \(A_n\) on the bulletin board. Assuming the voter \(v_a\) has voted for the candidate \(A_n\), then their corresponding message, which is published by Counter on the bulletin board, is \(A_n\| (e_a \oplus dEnc(pk^*_{ab}, f_a))\), where \(e_a\) and \(f_a\) are binary strings of lengths randomly generated by \(v_a\) and Counter, respectively. As \(sk^*_{ad}\) is known, \(v_i\) can obtain \(f_{false}\) which satisfies \(e_i \oplus dEnc(pk^*_{ab}, f_a) = e_i \oplus dEnc(pk^*_{ab}, f_{false})\). Then, \(v_i\) can obtain \(f_{false}\) by running the \(dDec\) algorithm, i.e., \(f_{false} = dDec(sk^*_{ad}, e_i \oplus e_a \oplus dEnc(pk^*_{ab}, f_a))\). Then, \(v_i\) tells \(f_{false}\) to the briber, such that the briber can find \(A_n\| (e_a \oplus dEnc(pk^*_{ab}, f_{false}))\) on the bulletin board. Therefore, with \(f_i, v_i\) cannot prove to the briber how they voted. In conclusion, the protocol is receipt-free. \(\square\)

5. Instantiation of Our Generic Construction

In this section, we instantiate the general construction of the election protocol.

5.1. Instantiation

5.1.1. Quantum Public Key Encryption

\(qPKE = (qGen, qEnc, qDec)\). To achieve post-quantum security for the protocol, we instantiate the construction of quantum public key cryptography \(qPKE\) with quantum
public key encryption scheme [25] based on conjugate encoding which is information-theoretic secure.

Let $I$ and $Y$ be two of the Pauli matrices and $H$ be the Hadamard transformation, where

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (11)$$

For $k = (k_1, k_2, \ldots, k_n) \in \{0, 1\}^n$, we define $H_k = H^{k_1} \otimes H^{k_2} \otimes \cdots \otimes H^{k_n}$, where $H$ is the Hadamard transformation mentioned as above, $H^0 = I$, $H^1 = H$, and $\otimes$ is the tensor product. Similarly, we also define $Y_k = Y^{k_1} \otimes Y^{k_2} \otimes \cdots \otimes Y^{k_n}$, where $Y$ is one of the Pauli matrices defined as above. Let $p$ denote the number of messages to be encrypted, and $m_i \in \{0, 1\}^n$ denote the $i$-th message for $i \in \{1, 2, \ldots, p\}$.

(i) $qGen$. (1) Randomly select two functions $F_1, F_2$ as private key from a set of polynomial computable functions $\{F : [0, 1]^n \rightarrow [0, 1]^n\}$. (2) Randomly select $g_i \in \{0, 1\}^n$ and compute $k_{i,1} = F_1(g_i)$ and $k_{i,2} = F_2(g_i)$. (3) Apply $H_{k_{i,2}}$ to $|k_{i,2}\rangle$ and get $H_{k_{i,2}}|k_{i,2}\rangle$. Take $(g_i, H_{k_{i,2}}|k_{i,2}\rangle)$ as one public key. (4) In a word, $qGen(1^n) \rightarrow (pk = \{pk_i\}_i^{1,2,\ldots,p})$, $sk = (F_1, F_2)$, where $pk_i = \{g_i, H_{k_{i,2}}|k_{i,2}\rangle\}$.

(ii) $qEnc$. (1) In order to encrypt the classic message $m_i$, download the classic and quantum public-key pair $pk_i = \{g_i, H_{k_{i,2}}|k_{i,2}\rangle\}$ from the public key register. (2) Encrypt $m_i \in \{0, 1\}^n$ by applying $Y_{m_i}$ to $H_{k_{i,2}}|k_{i,2}\rangle$, then get the ciphertext $c = (g_i, Y_{m_i}H_{k_{i,2}}|k_{i,2}\rangle)$. It is clear that $\sigma(m_i, H_{k_{i,2}}|k_{i,2}\rangle) = Y_{m_i}H_{k_{i,2}}|k_{i,2}\rangle$. (3) In a word, $qEnc(pk_i, m_i) \rightarrow c$.

(iii) $qDec$. (1) Given the ciphertext $c = (g_i, Y_{m_i}H_{k_{i,2}}|k_{i,2}\rangle)$, use private key $F_1, F_2$ to compute $k_{i,1} = F_1(g_i), k_{i,2} = F_2(g_i)$. (2) According to $k_{i,1}$, apply $H_{k_{i,1}}$ to $Y_{m_i}H_{k_{i,2}}|k_{i,2}\rangle$ and measure on the basis $\{|0\rangle, |1\rangle\}^n$ to obtain $m_i \oplus k_{i,2}$. Then, with $k_{i,2}$, perform an exclusive OR on $m_i \oplus k_{i,2}$ to get $m_i$. (3) It is clear that the algorithm $\mathcal{D}$ is the whole process of step (2) described as above. (4) In a word, $qDec(sk, c = (g_i, Y_{m_i}H_{k_{i,2}}|k_{i,2}\rangle)) \rightarrow m_i$.

5.1.2. Quantum Digital Signature $qSignature = (qKGen, qSign, qVerf)$ The function of quantum signature in the election protocol is to make the verifier get the information that the signer wants to send authenticatively. To achieve post-quantum security for the protocol, we instantiate the construction of quantum digital signature $qSignature$ with quantum digital signature [24] which is constructed based on the McEliece cryptosystem with post-quantum security [59]. The McEliece secret key consists of a nonsingular $n \times n$ matrix $S$; a generator matrix $G$ size of $n \times t$ for a Goppa code; and a $t \times t$ permutation matrix $P$. The McEliece public key is the $n \times t$ matrix SGP. In the following, $t' = (t/2)$ and $n' = (t/2)$.

(i) $qKGen$. (1) Generate the McEliece public and secret key $(SGP, (S, G, P))$. (2) Let the secret key $sk_{sig}$ be $(S, G, P)$ and the verification key $vk_{sig}$ be $G' = SGP$.

(ii) $qSign$. The signer signs a $n$ bits quantum message $|\psi\rangle = \sum_{m} a_{m|m\rangle}0\rangle$ to the verifier as follows: (1) The verifier randomly generates a number $r_{ver} \in [0, 1]^{n'}$ and sends it to the signer. (2) The signer randomly generates a number $r_{sig} \in [0, 1]^{n'}$. (3) With the knowledge of $sk_{sig} = (S, G, P)$, the signer can get $(r', r'' \equiv r \mod n')$, where $r \in [0, 1]^{n'}, r' \in [0, 1]^{n'}$. (4) Note that $f^{-1}$ is the whole process of step (3) described as above and $f(r', r'') \equiv r'G' \oplus r$. (5) Then, with $r$, the signer performs the quantum transformation $U_f$ on the quantum message $|\psi\rangle0\rangle = \sum_{m} a_{m|m\rangle}0\rangle$, and sends quantum state $\sum_{m} a_{m|m\rangle}f(r, m)\rangle$ to the verifier. (4) In a word, $qSign(sk_{sig}, |\psi\rangle) \rightarrow \sigma$, where

$$\sigma = \left( r_{ver}, r', r'' \sum_{m} a_{m|m\rangle}f(r, m)\rangle \right). \quad (13)$$

(iii) $qVerf$. (1) The verifier tells the signer that they have received the quantum state. (2) The signer announces $r$ and $r'$. (3) The verifier computes $f(r', r'') \equiv r'G' \oplus r$ and checks whether the first $t'$ bits of $f(r', r'')$ equal $r_{ver}$. Then they perform the transformation

$$\sum_{m} a_{m|m\rangle}f(r, m)\rangle \rightarrow \sum_{m} a_{m|m\rangle}0\rangle, \quad (14)$$

and measure the second quantum register. They accept the signature if and only if the second register is in state $0\rangle$. In a word, $qVerf(vk_{sig}, (|\psi\rangle, \sigma)) \rightarrow 1$ if and only if the second register is in state $0\rangle$.

5.1.3. Classic Public Key Encryption pPKE and dPKE. The classic probabilistic public key encryption pPKE in the instantiation can be any classic public key encryption schemes with post-quantum security such as public key
encryption based on coding [59], lattices [60], and multivariate [61].

Bellare et al. [62] proposed a generic deterministic public key encryption construction from probabilistic public key encryption, i.e., if there exists a probabilistic public key encryption $pPKE = (pGen, pEnc, pDec)$, then there is a classic deterministic public key encryption scheme $dPKE = (dGen, dEnc, dDec)$ which is illustrated in Figure 4. In the deterministic PKE construction, $H$ denotes a hash function, such as SHA2 and SHA3.

### 5.2. Efficiency Analysis

According to Ref. [18], quantum bit efficiency of quantum election protocol is defined as

$$n = C/Q,$$ \hspace{1cm} (15)

where the total number of transmitted classic bits (message bits) is $C$ and the total number of quantum bits generated is $Q$.

We can split this protocol into two parts. In the first part, the voter passes the vote message to Counter with the help of Admin. In the second part, Counter passes the verification information to the corresponding voter with the help of Admin. In the first part, the valid classic information transmitted is $T_{ji}$, and its number of bits is $n$. To transmit $T_{ji}$, the $n$ bits quantum public key $H_{k_i}|k_{i,2}$, $n$ bits quantum ciphertext $Y_{m_i}H_{k_j}|k_{i,2}$, and $|\psi_{i,2}$, $|\psi_{i,3}$ are generated, where $|\psi_{i,2}$, $|\psi_{i,3}$ are both $n + t$ bits quantum states generated during the signature process. Hence, in this part, total of $4n + 2t$ quantum bits are generated. In the second part, the valid classic information transmitted is $f_i \oplus e_i$, whose length is $s$, where there is no quantum bit generated. Thus, the number $C$ of valid classic bits transmitted in the protocol is $n + s$, and the number $Q$ of quantum bits generated in the protocol is $4n + 2t$. Subsequently, the efficiency of the protocol is

$$\eta = \frac{n + s}{4n + 2t}.$$ \hspace{1cm} (16)

When $s$ is equal to $t$, the efficiency is greater than 1/4.

### 5.2.1. Performance Comparison

There exist many quantum election protocols based on entangled states [12–14, 16–20]. However, some [13, 14, 16, 17] of these protocols can only perform a binary vote for “yes” or “no.” As a result, we make a performance comparison between our protocol and the quantum election protocols [12, 15, 18–20] which are applicable for scenarios with numerous candidates. Please read Table 1 for more details.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Quantum resource</th>
<th>Efficiency</th>
<th>Interactive times</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZWL20 [15]</td>
<td>Cluster states</td>
<td>1/9</td>
<td>4</td>
</tr>
<tr>
<td>ZXZ17 [19]</td>
<td>Entangled states</td>
<td>1/6</td>
<td>9</td>
</tr>
<tr>
<td>CDY16 [12]</td>
<td>Bell states</td>
<td>1/4</td>
<td>9</td>
</tr>
<tr>
<td>ZZX18 [20]</td>
<td>Cluster states</td>
<td>1/4</td>
<td>4</td>
</tr>
<tr>
<td>WXJ20 [18]</td>
<td>GHZ states</td>
<td>1/3</td>
<td>3</td>
</tr>
<tr>
<td>Ours</td>
<td>Entangled states</td>
<td>&gt; 1/4</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Performance comparison.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Completeness</th>
<th>Robustness</th>
<th>Privacy</th>
<th>Legality</th>
<th>Unreusability</th>
<th>Fairness</th>
<th>Verifiability</th>
<th>Receipt-freeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZWL20 [15]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>ZXZ17 [19]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>CDY16 [12]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>ZZX18 [20]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>WXJ20 [18]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2: Security comparison.
requirements. For the same reason which is described in Section 5.2, we make a security comparison between our protocol and the quantum election protocols [12, 15, 18–20] which are applicable for scenarios with numerous candidates. Please read Table 2 for more details. It is clear that our election protocol is much more secure than other schemes.

6. Conclusion

In this paper, we utilise the general construction of quantum digital signature [24] and quantum public key encryption [25], in conjunction with classic public key encryption, to develop a general construction of a quantum election protocol. The protocol is suitable for large-scale elections with numerous candidates and voters and accommodates scenarios in which multiple voters vote simultaneously. The protocol does not require any two participants to pre-share a secret key, nor does it require anonymous channels or a trusted third party. This construction is the first to achieve the properties of completeness, robustness, privacy, legality, unreusability, fairness, verifiability, and receipt-freeness under attacks from both external and internal agents in the current quantum election protocols. We instantiate the general construction into an election scheme with post-quantum security by applying cryptographic algorithms having post-quantum security. Furthermore, our election protocol has higher security than that of existing protocols. All the keys of the quantum digital signature are classic so that the classic PKI can be used for key management and distribution.

This construction is a theoretical achievement, and any practical application remains a distant goal. The quantum public key cryptosystem has significant room for development in comparison with the relatively well-established conventional public key cryptosystem. Numerous problems remain to be solved, such as developing the quantum PKI.

Data Availability

The data used are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61672517), National Natural Science Foundation of China (Key Program, Grant No. 61732021), National Cryptography Development Fund (Grant No. MMJ20170108), and Beijing Municipal Science & Technology Commission (Grant no. Z19110000719006).

References


