CMC, vol.59, no.1, pp.149-165, 2019

Quantum Homomorphic Signature with Repeatable Verification

Tao Shang[1, *], Zhuang Pei[2], Ranyiliu Chen[3] and Jianwei Liu[1]

Abstract: In January 2015, the first quantum homomorphic signature scheme was proposed creatively. However, only one verifier is allowed to verify a signature once in this scheme. In order to support repeatable verification for general scenario, we propose a new quantum homomorphic signature scheme with repeatable verification by introducing serial verification model and parallel verification model. Serial verification model solves the problem of signature verification by combining key distribution and Bell measurement. Parallel verification model solves the problem of signature duplication by logically treating one particle of an EPR pair as a quantum signature and physically preparing a new EPR pair. These models will be beneficial to the signature verification of general scenarios. Scheme analysis shows that both intermediate verifiers and terminal verifiers can successfully verify signatures in the same operation with fewer resource consumption, and especially the verified signature in entangled states can be used repeatedly.

Keywords: Quantum homomorphic signature, repeatable verification, serial model, parallel model, bell measurement.

1 Introduction

With the rapid development of quantum computing technology [Liu, Xu, Yang et al. (2018)] and the feasibility of new quantum bit preparation technologies [Wang, Yang and Mousoli (2018)], quantum cryptographic protocols like quantum signature can play the full role of unconditional security. Quantum signature is a combination of quantum theory and classical digital signature. Unlike quantum-secure signatures which are based on classical hard problems against quantum adversaries, quantum signature can provide unconditionally secure signature by taking advantage of quantum effects. It has been paid much attention and many quantum signature schemes have been proposed.

In 2001, Gottesman et al. [Gottesman and Chuang (2001)] proposed the first quantum signature scheme based on quantum one-way functions and quantum Swap-test. In this scheme, the public key can only be used once for signing merely one bit of message each

[1] School of Cyber Science & Technology, Beihang University, Beijing, 100083, China.
[2] Institut Fresnel, Ecole Centrale de Marseille, Marseille, 13451, France.
[3] School of Electronic & Information Engineering, Beihang University, Beijing, 100083, China.
[*]Corresponding Author: Tao Shang. Email: shangtao@buaa.edu.cn.

CMC. doi:10.32604/cmc.2019.05360

time. In 2002, Zeng et al. [Zeng and Keitel (2002)] proposed a pioneering arbitrated quantum signature (AQS) protocol which can be used to sign both classical message and quantum one. This scheme uses the correlation of Greenberger-Horne-Zeilinger (GHZ) triplet states and quantum one-time pads to ensure the security. As a necessary and important technique, probabilistic comparison of two unknown quantum states [Barnett, Chefles and Jex (2003); Filippov and Ziman (2012)] was also introduced to verify the validity of a signature. This work provides an elementary model to sign a quantum message. Although it was mentioned that both known and unknown quantum states could be signed, there were some corresponding comments about whether it was suitable for unknown messages [Curty and Lutkenhaus (2008); Zeng (2008)]. Then a variety of quantum signature schemes were proposed.

As a kind of efficient signature, homomorphic signature [Johnson, Molnar, Song et al. (2002)] allows intermediate verifiers to generate a new signature by directly manipulating the original signatures of received messages without encryption operation. It has drawn much attention and many schemes have been proposed in classical cryptography. Classical homomorphic signature schemes are used to protect classical information in communication networks. However, it is believed that homomorphic signature of quantum information is more meaningful and difficult than its counterpart in classical cryptography. In 2015, Shang et al. [Shang, Zhao, Wang et al. (2015)] creatively treated entanglement swapping as a homomorphic operation and proposed the first quantum homomorphic signature (QHS) scheme. The scheme is additively homomorphic and can generate quantum signatures for classical messages, but only one verifier is allowed to verify the signature once in this scheme. In 2017, Shang et al. [Li, Shang and Liu (2017)] further proposed the quantum homomorphic signature scheme for continuous variables. In 2018, Shang et al. [Shang, Li and Liu (2018)] analyzed the measurement-device independency of the signature scheme. In fact, it is actually required that more than one verifier needs to verify a signature many times. The solution to such problems will be beneficial to the application of quantum homomorphic signature to general scenarios.

In this paper, from the viewpoint of repeatable verification of quantum signature in general scenarios, we propose a new quantum homomorphic signature scheme with repeatable verification which assures intermediate verifiers as well as terminal verifiers can verify signatures repeatedly.

2 Related works

2.1 Quantum homomorphic signature scheme

It is crucial for Shang et al.'s scheme [Shang, Zhao, Wang et al. (2015)] to treat one particle of an EPR pair as a quantum signature and entanglement swapping as a quantum homomorphic operation. The signed message is classical information and the corresponding signature is quantum information.

As shown in Fig. 1, the messages that the signers A_1 and A_2 want to send are the classical bits X_1 and X_2, respectively. The messages and the corresponding signatures will first be

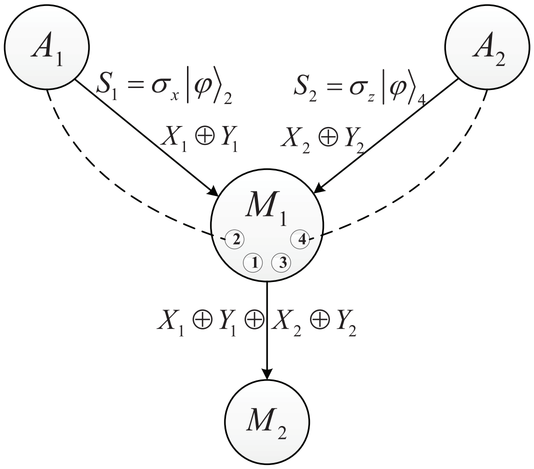
A_1
A_2
$S_1 = \sigma_x |\varphi\rangle_2$
$S_2 = \sigma_z |\varphi\rangle_4$
$X_1 \oplus Y_1$
$X_2 \oplus Y_2$
M_1
2
4
1
3
$X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$
M_2

Figure 1: Quantum homomorphic signature scheme

sent to the aggregator M_1. By performing entanglement swapping on the received quantum signatures, M_1 can generate the homomorphic signature of the encoded information $X_1 \oplus X_2$. The encoded information and its signature will finally be sent to the verifier M_2 for verification.

The scheme consists of four algorithms, namely *Setup, Sign, Combine* and *Verify*.

(1) ***Setup***

Step 1: quantum key distribution. A_1 (A_2) chooses two classical bits Y_1 (Y_2) as its secret key and shares this key with M_2 by the quantum key distribution protocol. Here, an improved BB84 protocol with authentication [Beige, Englert and Kurtsiefer (2002)] is used to ensure the security.

Step 2: EPR pair distribution. M_1 prepares two EPR pairs:

$$\begin{aligned} \left|\phi^+\right\rangle_{12} &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)_{12} \\ \left|\phi^+\right\rangle_{34} &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)_{34} \end{aligned} \tag{1}$$

M_1 sends particles 2 and 4 (namely $|\varphi\rangle_2$ and $|\varphi\rangle_4$) to A_1 and A_2, respectively.

(2) ***Sign***

After receiving the particle from M_1, A_1 (A_2) chooses a unitary operator according to the result of $X_1 \oplus Y_1$ ($X_2 \oplus Y_2$), and performs a corresponding operation on the particle 2 (4). The particle 2 (4) after the operation is viewed as the signature of the information X_1 (X_2).

The unitary operator corresponding to the result of $X_i \oplus Y_i$ is chosen as follows:

$$\begin{aligned} X_i \oplus Y_i = 00 \quad &\rightarrow \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ X_i \oplus Y_i = 01 \quad &\rightarrow \quad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ X_i \oplus Y_i = 10 \quad &\rightarrow \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ X_i \oplus Y_i = 11 \quad &\rightarrow \quad -i\sigma_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned} \tag{2}$$

After the phase of *Sign*, the two EPR pairs become:

$$\begin{aligned} \left|\psi'\right\rangle_{12} &= U(X_1 \oplus Y_1)^{(2)} \left|\phi^+\right\rangle_{12} \\ \left|\psi'\right\rangle_{34} &= U(X_2 \oplus Y_2)^{(4)} \left|\phi^+\right\rangle_{34} \end{aligned} \tag{3}$$

(3) ***Combine***

Step 1: A_1 (A_2) sends the encrypted information $X_1 \oplus Y_1$ ($X_2 \oplus Y_2$) and its signature $|\psi'\rangle_2$ ($|\psi'\rangle_4$), namely the particle 2 (4), to the aggregator M_1.

Step 2: M_1 performs a Bell measurement on the particles (1, 3), the measurement result is noted as $|\psi''\rangle_{13}$. According to entanglement swapping, the particles (2, 4) will fall into a certain state $|\psi''\rangle_{24}$. Here, the particle 4, namely $|\psi''\rangle_4$, is exactly the signature of the information $X_1 \oplus X_2$.

Step 3: M_1 sends the classical information $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ and the particles (1, 2, 3, 4) (namely $|\psi''\rangle_{13} \otimes |\psi''\rangle_{24}$) to the verifier M_2.

(4) ***Verify***

When M_2 gets the classical information and all the particles from M_1, it can verify the signature in the following steps:

Step 1: M_2 first performs a Bell measurement on the particles (1, 3) to get $|\psi''\rangle_{13}$, and then performs a Bell measurement on the particles (2, 4) to get $|\psi''\rangle_{24}$.

Step 2: By comparing $|\psi\rangle_{24}$ and $|\psi''\rangle_{24}$, M_2 will get a unitary operator $U(Z)$ such that $|\psi''\rangle_{24} = c(Z) \cdot U(Z)^{(4)} |\psi\rangle_{24}$, in which $|c(Z)| = 1$. $|\psi\rangle_{24}$ is the result of entanglement swapping without performing unitary operators on the particles 2 and 4.

Step 3: M_2 compares Z with $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$. If $Z = X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$, M_2 accepts the signature; otherwise, M_2 denies the signature.

2.2 Main problems

For the quantum homomorphic signature scheme, there still exist several problems for general scenarios.

(1) ***An intermediate verifier cannot verify signatures.***

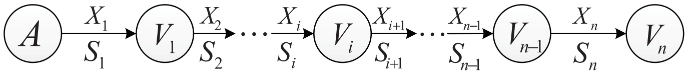

Figure 2: Successive signature verification

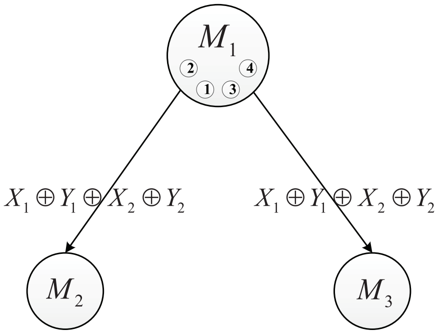
M_1
2
4
1
3
$X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$
$X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$
M_2
M_3

Figure 3: Signature duplication

As shown in Fig. 1, the aggregator M_1 in this scheme can also be viewed as an intermediate verifier. While it can generate a quantum homomorphic signature from two received signatures S_1 and S_2, it cannot verify the latter ones in advance. If the message $X_i \oplus Y_i$ or its signature S_i was changed during transmission, this change will not be found until all information arrives at the terminal verifier M_2. Such problem is not evident in Shang et al's scheme since the information passes through only two verifiers. As the number of intermediate verifiers increases, the problem will become really serious.

As we can see in Fig. 2, the signer A first generates a signature S_1 of the information X_1, then the information and its signature passes through a series of intermediate verifiers (or verifier groups) till they finally arrive at the terminal verifier V_n. Since the intermediate verifiers (verifier groups) cannot verify the signatures, all errors during the transmission can only be found by the terminal verifier V_n, which not only causes the waste of resources, but also reduces the efficiency of communication. The more intermediate verifiers (verifier groups) there are, the more serious the problem will be.

(2) ***The duplication of quantum signature is not provided.***

Let us take a look at the case of Fig. 3. Here, we add a terminal verifier M_3. Since the classical information $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ could be easily duplicated, M_1 can send it to M_2 and M_3 at the same time. However, as there is only one share of quantum particles, just one verifier (M_2 or M_3) can fulfill the verification of a signature. In order to assure that multiple verifiers can verify signatures, we should introduce the duplication of quantum signatures.

(3) ***The signature between M_1 and M_2 can be forged.***

Assume that an attacker can intercept the classical information $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ and the particles (1, 2, 3, 4), then it can forge the signature. Here we take an example to illustrate it.

Example 1: Suppose that the state of the particles (1, 3) after the entanglement swapping is $|\psi''\rangle_{13} = |\psi^+\rangle_{13}$, as a result $|\psi''\rangle_{24} = c \cdot U(X_1 \oplus Y_1 \oplus X_2 \oplus Y_2)^{(4)} |\psi^+\rangle_{24}$. The verifier accepts the signature as long as the received classical message Z matches the Bell state $|\psi''\rangle_{24} = c \cdot U(Z)^{(4)} |\psi^+\rangle_{24}$ (here, $Z = X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$). For this reason, an attacker can forge the signature in a simple way.

An attacker just needs to replace the classical bits $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ by a corrupt data E while preparing two entangled particles (5, 6) such that $|\psi\rangle_{56} = c \cdot U(E)^{(4)} |\psi^+\rangle_{24}$. Obviously, the verifier would confirm the signature according to the received information E and the particles (1, 3, 5, 6). In other words, the attacker has forged the signature successfully.

3 Proposed QHS scheme

With regard to the problems in Section 2, we provide corresponding solutions. Firstly, we construct the serial verification model, and introduce key distribution and Bell measurement for intermediate verifiers to verify signatures. Secondly, we construct the parallel verification model, and realize the duplication of signatures by logically treating one particle of an EPR pair as a quantum signature and physically preparing a new EPR pair.

3.1 Serial verification model

Generally, in a homomorphic signature scheme, an intermediate verifier must verify the received signatures at first before generating a new homomorphic signature. For this reason, we intend to realize the verification of quantum homomorphic signature for intermediate verifiers. Unlike classical homomorphic signatures who use public and private key pairs, the original quantum homomorphic signature scheme cannot take effect in that way. So we introduce key distribution to guarantee the verification of quantum homomorphic signature for intermediate verifiers.

We define serial verification as the case in which a message and its signature successively pass through a series of intermediate verifiers (or verifier groups) and finally reach terminal verifiers. In general, the serial verification model has an inverted pyramid-shaped structure as shown in Fig. 4.

As we can see in Fig. 4, A_1, A_2 and A_3 are the signers; B_1 and B_2 are the intermediate verifiers (the first intermediate verifier group); C_1 is the terminal verifier (the second intermediate verifier group). The signed messages and their signatures will pass through the intermediate verifiers and finally arrive at the terminal verifier.

Compared to the original quantum homomorphic signature scheme, our scheme has made some changes. The main ideas are described as follows:

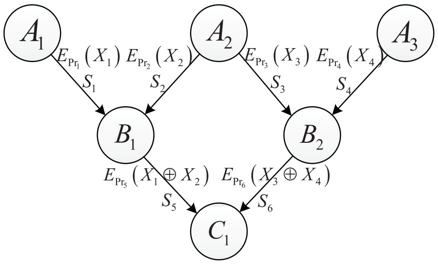

Figure 4: Serial verification model

1) In the original quantum homomorphic signature scheme, the EPR pairs are prepared by the first verifier group (e.g., M_1 in Fig. 1). On one hand, this means that the members of the first verifier group must be trusted nodes. On the other hand, this causes the dissimilarity of the verification operation between the verifiers. In order to assure the repeatability of our scheme, i.e., all the verifiers can fulfill the verification of signatures in the same operation, the EPR pairs are prepared by the original signers. In fact, such change makes no difference to the generation of signatures and does not affect the properties of our scheme.

2) To solve the forgery problem in the original quantum homomorphic signature scheme, we transmit the encrypted classical information $E_{Pr_i}(X_i)$ instead of $X_i \oplus Y_i$. Here, Pr_i is the private key of a signer for signing the message X_i and E is the encryption algorithm. To fulfill the verification of a signature, a signer has to send its public key Pb_i to its corresponding verifier so that the verifier can get $X_i = D_{Pb_i}(E_{Pr_i}(X_i))$ with the decryption algorithm D. In our scheme, (Pr_i, Pb_i) is called a signature key pair, and Y_i is called an encryption key.

According to the serial verification model, our quantum homomorphic signatures scheme can be described in the following steps.

(1) ***Setup***

Step 1: secret key generation and distribution. The signer first generates a signature key pair (Pr_i, Pb_i). Then it keeps the private key Pr_i and sends the public key Pb_i to its corresponding verifier by the quantum key distribution protocol such as an improved BB84 protocol with authentication [Beige, Englert and Kurtsiefer (2002)]. In particular, A_1 sends its public key Pb_1 to B_1; A_2 sends Pb_2 to B_1 and B_2; A_3 sends Pb_4 to B_2. In order to simplify the description and help the understanding of our scheme, we number the secret key pairs according to the messages. For example, the signer A_2 only needs to generate one secret key pair (Pr_2, Pb_2), but we also call it (Pr_3, Pb_3) when it is used to encrypt and decrypt the information X_3. In the following part, we will not point out this usage unless necessary.

The encryption key Y_i is used for the confidentiality of a message and can also be shared by an improved BB84 protocol with authentication.

Step 2: EPR pair preparation. A_1 prepares an EPR pair $|\phi^+\rangle_{12}$; A_2 prepares two EPR pairs $|\phi^+\rangle_{34}$ and $|\phi^+\rangle_{56}$; A_3 prepares an EPR pair $|\phi^+\rangle_{78}$. Here, $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

(2) ***Sign***

A_1 calculates the result of $X_1 \oplus Y_1$, and according to this result it performs a unitary operation on the particle 2. After this operation, the particle 2 can be viewed as the signature of X_1 (Note that the particles 1 and 2 are still entangled). Similarly, we can get the signatures of X_2, X_3, and X_4.

After the phase of *Sign*, the states of the entangled particles become

$$\begin{aligned}
\left|\psi'\right\rangle_{12} &= U(X_1 \oplus Y_1)^{(2)} \left|\phi^+\right\rangle_{12} \\
\left|\psi'\right\rangle_{34} &= U(X_2 \oplus Y_2)^{(4)} \left|\phi^+\right\rangle_{34} \\
\left|\psi'\right\rangle_{56} &= U(X_3 \oplus Y_3)^{(6)} \left|\phi^+\right\rangle_{56} \\
\left|\psi'\right\rangle_{78} &= U(X_4 \oplus Y_4)^{(8)} \left|\phi^+\right\rangle_{78}
\end{aligned} \tag{4}$$

The signers then send the encrypted information $E_{Pr_i}(X_i)$ and the signature particles to their corresponding verifiers. To be concrete, A_1 sends the classical information $E_{Pr_1}(X_1)$ and the quantum particles (1, 2) to B_1; A_2 sends $E_{Pr_2}(X_2)$ and the particles (3, 4) to B_1; A_2 sends $E_{Pr_3}(X_3)$ and the particles (5, 6) to B_2; A_3 sends $E_{Pr_4}(X_4)$ and the particles (7, 8) to B_2.

(3) ***Verify original signatures***

When B_1 receives the information $E_{Pr_1}(X_1)$ and its signature S_1 (namely the particles (1, 2)), it first gets the information $X_1 = D_{Pb_1}(E_{Pr_1}(X_1))$ with the help of the public key Pb_1 and calculates the exclusive OR result $X_1 \oplus Y_1$. Then it performs a Bell measurement on the particles (1, 2). If the measurement result equals to $U(X_1 \oplus Y_1)^{(2)} |\phi^+\rangle_{12}$, B_1 accepts the signature; otherwise, it denies the signature.

B_1 and B_2 can verify the other signatures in the same way.

(4) ***Combine***

After the intermediate verifiers have verified the received signatures, they can generate the quantum homomorphic signature of the encoded message by entanglement swapping. The details of entanglement swapping are described in Shang et al. [Shang, Zhao, Wang et al. (2015)].

While B_1 measures the particles (1, 3) to get $|\psi''\rangle_{13}$, the particles (2, 4) will collapse to a certain state $|\psi''\rangle_{24}$. Here, $|\psi''\rangle_4$ is the generated quantum homomorphic signature which is also the signature of the message $X_5 = X_1 \oplus X_2$.

B_2 can generate the signature of $X_6 = X_3 \oplus X_4$ in the same way.

(5) ***Verify homomorphic signatures***

Step 1: generation and distribution of secret keys. B_1 calculates a new encryption key

$Y_5 = Y_1 \oplus Y_2$ and shares it with C_1 by the improved quantum key distribution protocol. B_1 generates a signature key pair (Pr_5, Pb_5), then it keeps the private key Pr_5 and sends the public key Pb_5 to C_1 by the improved quantum key distribution protocol.

Step 2: translation from quantum states to classical bits. B_1 translates the Bell measurement result of the particles (1, 3) to classical bits according to the following rules:

$$\begin{aligned} \left|\phi^+\right\rangle_{13} &\rightarrow 00 \\ \left|\phi^-\right\rangle_{13} &\rightarrow 01 \\ \left|\psi^+\right\rangle_{13} &\rightarrow 10 \\ \left|\psi^-\right\rangle_{13} &\rightarrow 11 \end{aligned} \tag{5}$$

Assume that the measurement result of the particles (1, 3) in the entanglement swapping is $|\psi''\rangle_{13} = |\psi^+\rangle_{13}$, then the state of the particles (2, 4) will be $|\psi''\rangle_{24} = c \cdot U(X_1 \oplus Y_1 \oplus X_2 \oplus Y_2)^{(4)} |\psi^+\rangle_{24}$. In this case, the measurement result of the particles (1, 3) will be translated into the classical bits 10.

By sending the classical bits, B_1 only needs to send 2 particles other than 4 particles to the terminal verifier C_1, so does B_2. We can imagine that no matter how many verifiers there are, each intermediate verifier just needs to send 2 particles to its successor. Thus the number of quantum particles is reduced along with the verification of signatures.

Step 3: transmission of related information. First, B_1 sends the classical bits 10 to C_1. To ensure the security, we use the improved quantum key distribution protocol. Then, B_1 sends the encoded and encrypted information $E_{Pr_5}(X_5) = E_{Pr_5}(X_1 \oplus X_2)$ and the signature particles (2, 4) to C_1.

Step 4: verification of quantum homomorphic signature. As we just mentioned, if the state of the particles (1, 3) is $|\psi''\rangle_{13}$, the particles (2, 4) will be in the corresponding state $|\psi''\rangle_{24}$. According to the classical bits shared with B_1, C_1 can easily derive $|\psi''\rangle_{13}$. C_1 calculates $X_5 = D_{Pb_5}(E_{Pr_5}(X_5))$ and $X_5 \oplus Y_5 = X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$. Then it performs a Bell measurement on the particles (2, 4). If the result equals to $|\psi''\rangle_{24}$, it accepts the signature; otherwise, it denies the signature.

The signature of $X_6 = X_3 \oplus X_4$ can be verified in the same way.

When the number of verifiers in the serial verification model increases, we just need to repeat the above process and all verifiers can fulfill the verification of signatures.

3.2 Parallel verification model

As we mentioned earlier, the original scheme lacks the duplication of signatures, which makes it impossible for all verifiers to realize the verification in the case where an intermediate verifier is followed by more than one successors. So we add the operation of signature duplication and construct the parallel verification model. Here parallel verification refers to the situation in which a message and its signature have to be sent to more than one verifiers at the same time.

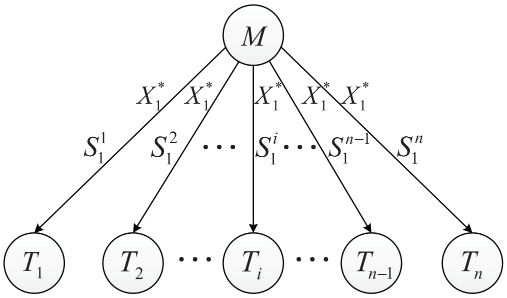

Figure 5: Parallel verification model

Fig. 5 shows a scenario of the parallel verification model. Here, we have ignored the signers and some intermediate verifiers to make the model more clear. M is an intermediate verifier and has N successors $T_1, T_2, \cdots, T_n$.

Assume that, after the combination of signatures, M holds an encoded and encrypted message $X_1{}^* = E_{Pr_1}(X_1)$, a new encryption key Y_1, and two pairs of entangled particles (1, 3) and (2, 4), with the particle 4 representing X_1's signature S_1. Here, Pr_1 is the private key of M and the corresponding public key is Pb_1. By the steps of *Verify*, all the n successors of M can achieve the verification of the signature S_1.

The signature scheme for the parallel verification model is similar to that for the serial verification model. The first four algorithms of *Setup*, *Sign*, *Verify original signatures*, and *Combine*, are the same, and only the algorithm *Verify homomorphic signatures* is different. So we just present the algorithm *Verify* homomorphic signatures*.

(5) ***Verify* homomorphic signatures***

Step 1: distribution of secret keys. M shares the key Y_1 and its public key Pb_1 with its n successors by the improved quantum key distribution protocol.

Step 2: translation from quantum states to classical bits. Assume that the state of the particles (1, 3) is $|\psi\rangle_{13} = |\psi^+\rangle_{13}$, then it will be translated to be the classical bits 10.

Step 3: signature duplication. As the signature S_1 is represented by the quantum particle 4, we cannot copy it independently without destroying it. However, if we treat the particles (2, 4) as a whole, we can duplicate the signature by preparing new EPR pairs in the same state.

According to the assumption of Step 2, the state of the particles (2, 4) will be $|\psi\rangle_{24} = c \cdot U(X_1 \oplus Y_1)^{(4)} \, |\psi^+\rangle_{24}$. Then we just need to prepare $n-1$ new EPR pairs such that the state of the i_{th} EPR pair is $|\psi\rangle_{ab}^{i} = |\psi\rangle_{24}$, with $i \in [2, n]$. Now we have n shares of the signature particles, so all the n successors of the intermediate verifier M can achieve the verification of the signature S_1.

Step 4: transmission of related information. First, M sends the encoded and encrypted information $X_1{}^*$ and the classical bits 10 to its n successors. Then, it sends the i_{th} EPR

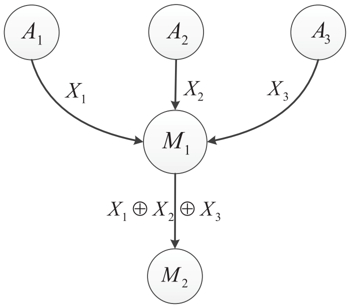

Figure 6: The situation of receiving more than two signatures

pair to its successive verifier T_i. Here, $i \in [1, n]$.

Step 5: signature verification. After receiving the information and the EPR pair $|\psi\rangle_{ab}^{i}$ from M, T_i first derives the state $|\psi^+\rangle_{13}$ according to the classical bits 10. Then it calculates $X_1 = D_{Pb_1}(X_1{}^*)$ and $X_1 \oplus Y_1$, and performs a Bell measurement on the received EPR pair $|\psi\rangle_{ab}^{i}$. If $|\psi\rangle_{ab}^{i} = |\psi\rangle_{24} = c \cdot U(X_1 \oplus Y_1)^{(b)} |\psi^+\rangle_{ab}^{i}$, T_i accepts the signature; otherwise, it denies the signature. Thus, all the n successors of M can verify the signature of X_1.

3.3 Application to general scenarios

Apart from the serial verification model and the parallel verification model, we should also consider the situation in which an intermediate verifier receives more than two signatures at the same time. Although this situation has already been discussed [Shang, Zhao, Wang et al. (2015)], there is still need to illuminate it in our scheme.

As shown in Fig. 6, A_1, A_2 and A_3 are the signers or the intermediate verifiers, M_1 is an intermediate verifier, and M_2 is a successor of M_1. The messages sent to M_1 are X_1, X_2 and X_3 with their corresponding signature particles (1, 2), (3, 4) and (5, 6). In this situation, M_1 can generate the quantum homomorphic signature by the following steps.

Step 1: M_1 receives the messages and signatures from A_1, A_2 and A_3, and then verifies the signatures by Bell measurement. Now the particles (1, 2), (3, 4) and (5, 6) are all in Bell states.

Step 2: M_1 performs a Bell measurement on the particles (1, 3) to get $|\psi''\rangle_{13}$, then the state of the particles (2, 4) will be $|\psi''\rangle_{24} = c_1 \cdot U(X_1 \oplus Y_1 \oplus X_2 \oplus Y_2)^{(4)} |\psi\rangle_{24}$.

Step 3: M_1 performs a Bell measurement on the particles (2, 5) to get $|\psi''\rangle_{25}$, then the state of the particles (4, 6) will be $|\psi''\rangle_{46} = c_1 \cdot c_2 \cdot U(X_1 \oplus Y_1 \oplus X_2 \oplus Y_2 \oplus X_3 \oplus Y_3)^{(6)} |\psi\rangle_{46}$. Now the particle 6 is the signature of the encoded message $X_1 \oplus X_2 \oplus X_3$. Then it can send the particles (4, 6) and the classical bits related to $|\psi''\rangle_{25}$ to M_2 for verification.

Based on the above models, we can apply our quantum homomorphic signature scheme to general scenarios now.

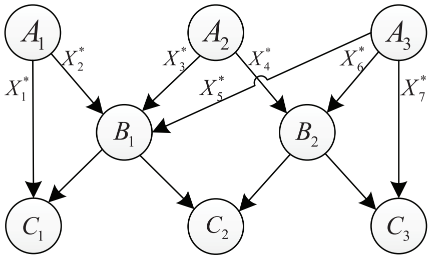

Figure 7: General scenario

Fig. 7 shows a general scenario based on serial verification model and parallel verification model. Let us have a look at how the quantum homomorphic signature scheme can be accomplished in this situation.

(1) ***Setup***

Step 1: secret key generation and distribution. A_1 generates a signature key pair (Pr_1, Pb_1) and sends its public key Pb_1 to C_1. Similarly, A_1, A_2 share their public keys Pb_2, Pb_3 with B_1, respectively. A_2 shares its public key Pb_4 with B_2. Then A_3 shares the public keys Pb_5, Pb_6, and Pb_7 with B_1, B_2 and C_3, respectively. Similarly, the signers share an encryption key Y_i with their corresponding signer.

Step 2: EPR pair preparation. A_1 prepares two EPR pairs $|\phi^+\rangle_{12}$, $|\phi^+\rangle_{34}$. A_2 prepares two EPR pairs $|\phi^+\rangle_{56}$, $|\phi^+\rangle_{78}$. A_3 prepares three EPR pairs $|\phi^+\rangle_{9,10}$, $|\phi^+\rangle_{11,12}$, and $|\phi^+\rangle_{13,14}$.

(2) ***Sign***

A_1 performs a unitary operation on the particle 2 according to the result of $X_1 \oplus Y_1$ to generate the signature of X_1. Similarly, A_1, A_2, A_3 can generate the signatures of X_2, X_3, $\cdots$, X_7. Then the signers send the encrypted information $X_i{}^* = E_{Pr_i}(X_i)$ and the signature particles ($2i-1$, $2i$) to their corresponding verifiers.

(3) ***Verify original signatures***

After receiving the classical information and the quantum particles from A_1, C_1 calculates $X_1 = D_{Pb_1}(E_{Pr_1}(X_1))$ and $X_1 \oplus Y_1$. Then it performs a Bell measurement on the particles (1, 2). If the measurement result is $|\psi''\rangle_{12} = c \cdot U(X_1 \oplus Y_1)^{(2)} |\phi^+\rangle_{12}$, C_1 accepts the signature of X_1; otherwise, it denies the signature.

The signatures of X_2, X_3, $\cdots$, X_7 can be verified in the same way.

(4) ***Combine***

B_1 receives three signatures at the same time. Then it can generate the signature of the encoded information $X_2 \oplus X_3 \oplus X_5$ by applying the method in Fig. 6 after verifying the received signatures. B_2 can generate the signature of the encoded information $X_4 \oplus X_6$ in the same way.

(5) ***Verify* homomorphic signatures***

When B_1 and B_2 have generated the homomorphic signatures, they have to send them to the terminal verifiers for verification. Note that both B_1 and B_2 has two successors, so their signatures need to be duplicated. Then the terminal verifiers can achieve the verification of signatures by the same steps as those in the parallel verification model.

Actually, serial verification model can also be found in this scenario. If C_1 and C_3 were ignored, the rest part can be treated as serial verification model.

4 Scheme analysis

4.1 Security analysis

Our signature scheme should achieve three basic proprieties, i.e., verifiability, undeniability and unforgeability.

1) *Verifiability:* A verifier is able to verify the validity of a signature after receiving it from its corresponding signer.

2) *Undeniability:* Once a signer has signed a message, it cannot deny its signature later.

3) *Unforgeability:* No one can generate a valid signature of a certain signer except for itself.

It should be emphasized that in our scheme an intermediate verifier can also be viewed as a signer for its successive verifiers. When we analyze the security of a signature scheme, we generally pay attention to two important security requirements, i.e., undeniabiliy and unforgeability. In this part, the security analysis of our scheme will be based on the serial verification model, because the only difference between serial verification model and parallel verification model is the phase of *Verify homomorphic signatures*, and such difference does not affect the security of our scheme.

In order to prove the undeniability and the unforgeability, we first give the following two lemmas.

Lemma 1: *The key Y_i and the public key Pb_i are shared by a signer and its corresponding verifier(s) securely.*

Proof: In our scheme, a signer shares the key Y_i and the public key Pb_i with its corresponding verifier(s) by the quantum key distribution protocol such as an improved BB84 protocol with authentication [Beige, Englert and Kurtsiefer (2002)], which has been proved to be unconditionally secure. Hence the key Y_i and Pb_i are shared securely. This means that any attacker cannot capture the key Y_i or the public key Pb_i.

Lemma 2: *It is impossible to calculate the key Y_i and the public key Pb_i by means of classical message and its corresponding quantum signature.*

Proof: As shown in Fig. 8, there are two cases in which an attacker can capture a classical message and its corresponding quantum signature particles. We will prove that in both cases the attacker cannot calculate the key Y_i or the public key Pb_i. The details are described as follows:

1) *If an attacker captures the classical message $E_{Pr_i}(X_i)$ and its corresponding quantum*

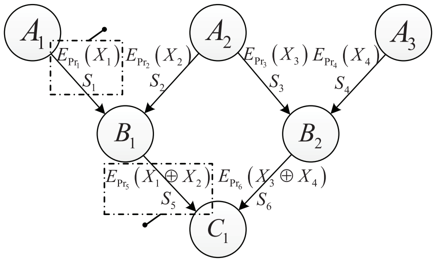

Figure 8: Attack model aimed at signature key

signature S_i which are sent by an original signer, it cannot obtain the key Y_i or the key Pb_i.

Assume that an attacker captures the classical message $E_{Pr_1}(X_1)$ and the signature S_1. Here, $S_1 = Sign(X_1) = |\psi'\rangle_{12} = c \cdot U(X_1 \oplus Y_1)^{(2)} |\phi^+\rangle_{12}$. By performing a Bell measurement on the particles (1, 2), the attacker can get $X_1 \oplus Y_1$. But, with the information $E_{Pr_1}(X_1)$ and $X_1 \oplus Y_1$, it can get neither Pr_1 nor Y_1.

2) *If an attacker captures the classical message $E_{Pr_i}(X_i)$ and its corresponding quantum signature S_i which are sent by an intermediate signer, it cannot obtain the key Y_i or the key Pb_i.*

Assume that an attacker captures the classical message $E_{Pr_5}(X_5)$ (remember that $X_5 = X_1 \oplus X_2$) and the signature S_5 sent by the intermediate signer B_1. Here, $S_5 = Sign(X_1 \oplus X_2) = |\psi''\rangle_{24} = c_1 \cdot U(X_1 \oplus Y_1 \oplus X_2 \oplus Y_2)^{(4)} |\psi\rangle_{24}$. Unlike in the previous case, this time the attacker cannot even get $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ by performing a Bell measurement. In order to get $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$, the attacker has to know $|\psi\rangle_{24}$, which is the entanglement swapping result without performing unitary operations on the particles 2 and 4. However, $|\psi\rangle_{24}$ depends on the result of $|\psi''\rangle_{13}$ which is transmitted securely from B_1 to C_1 by two classical bits. So it is impossible for the attacker to obtain $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$, much less the key $Y_5 = Y_1 \oplus Y_2$. Obviously, the attacker cannot obtain the public key Pb_5, either.

Property 1: *Anyone cannot forge a signature.*

Proof: As mentioned above, here we consider both the forgery of a third-party attacker and the forgery of a verifier. Therefore, the proof is divided into two parts.

1) *Any third-party attacker cannot forge a signature.*

In the original homomorphic signature scheme, the aggregator sends the information $X_1 \oplus Y_1 \oplus X_2 \oplus Y_2$ and the particles (1, 2, 3, 4) to the verifier. We have shown that in this case a third-party attacker can forge the signature by preparing a corrupt data Z and two entangled particles (5, 6) with $|\psi\rangle_{56} = c \cdot U(Z)^{(4)} |\psi\rangle_{24}$.

In our scheme, we have made some changes to solve that problem. One is that we transmit $E_{Pr_i}(X_i)$ instead of $X_i \oplus Y_i$, and the other is that we transmit the Bell measurement result of an EPR pair by two classical bits. Now we will prove that these changes can prevent any third-party attacker from obtaining the secret keys Y_i and Pr_i which are indispensable to generate a valid signature.

Suppose that a third-party attacker wants to forge a signature of the signer B_1. It prepares a corrupt data Z and two entangled particles (a, b). In order that the corrupt data and the particles pass the verification, the state of the particles (a, b) should satisfy $|\psi\rangle_{ab} = c \cdot U(Z \oplus Y_5)^{(4)} |\psi\rangle_{24}$. However, it is impossible for the attacker to prepare the particles (a, b) in the right state because Lemma 1 and Lemma 2 show that any attacker cannot obtain the key Y_5. So a third-party attacker cannot forge a signature.

In the second case, we will show that even with the key Y_5 an attacker cannot forge a signature either.

2) *Any verifier cannot forge a signature.*

Here, suppose that the verifier C_1 wants to forge a signature of the signer B_1. Compared with a third-party attacker, the verifier C_1 can get the key Y_5 and the public key Pb_5 of its corresponding signer B_1. With the key Y_5, C_1 can prepare a corrupt data Z and two entangled particles (a, b) with $|\psi\rangle_{ab} = c \cdot U(Z \oplus Y_5)^{(4)} |\psi\rangle_{24}$. In order that the corrupt data and the particles pass the verification, C_1 needs to process the data Z first before sending it to a verifier. After receiving the processed data Z^* and the particles (a, b) from C_1, a verifier calculates $D_{Pb_5}(Z^*) \oplus Y_5$ and performs a Bell measurement on the particles (a, b) for verification. The forged signature will pass the verification if and only if $D_{Pb_5}(Z^*) \oplus Y_5 = Z \oplus Y_5$, which means $Z^* = E_{Pr_5}(Z)$. This is impossible because the private key Pr_5 is only kept by B_1. Thus, any verifier cannot forge a signature.

Property 2: *Any signer cannot deny its signature.*

Proof: Suppose that a verifier receives an encrypted message X^* and the corresponding signature particles from a signer S. The verifier will first calculate $D_{Pb_i}(X^*) \oplus Y_i$, where Pb_i is the public key of the signer S. Then it will perform a Bell measurement on the signature particles to verify the signature of the information X_i. Once the data and the particles pass the verification, we can derive that $D_{Pb_i}(X^*) \oplus Y_i = X_i \oplus Y_i$, which means that $X^* = E_{Pr_i}(X_i)$. As the private key Pr_i is only kept by the signer S and no one can forge a signature, we can conclude that the message X^* and the corresponding signature particles are generated by S. So the signer S cannot deny its signature.

In addition, our quantum signature scheme is also additively homomorphic, which can be proved just as in Shang et al. [Shang, Zhao, Wang et al. (2015)].

4.2 Resource consumption analysis

We will analyze two types of resource consumption.

(1) ***Consumption of quantum resource***

Here, the quantum resource consumed mainly refers to the EPR pairs. As for our scheme, new signatures are generated from the old ones, so there is no need to prepare extra EPR pairs except in the situation where a signature needs to be duplicated. Therefore, the EPR pairs consumed come from two parts: ne part is the EPR pairs used to generate the original signatures, and the other part is those used to duplicate the signatures. Assume that there are m original messages to be signed at the very beginning and n signatures to be duplicated during the implementation of the scheme, then the number of the EPR pairs consumed in our scheme will be $m + n$.

By contrast, if we use an ordinary quantum signature scheme other than the homomorphic one, the consumption of the EPR pairs could be extremely large. Whenever a message is sent, a signature, namely an EPR pair, is needed. If N messages are sent during the whole process, the number of the EPR pairs consumed will be N.

Let us take the parallel verification model as an example. In Fig. 5, we can easily obtain that $m + n = 5, N = 7$. In general, N could be much larger than $m + n$.

(2) ***Consumption of secret keys***

We will first analyze the consumption of signature key pairs. In fact, every signer in our scheme has to generate its own signature key pair. Therefore, if there are in total p signers in our scheme, the number of the signature key pairs consumed will be p. The consumption of signature key pairs will be the same in an ordinary quantum signature scheme.

Different from the signature key pairs, the number of encryption keys is only determined by the number of original signers because all the other encryption keys are calculated from the original ones. Therefore, if there are q original signers in our scheme, the total number of the signature key pairs consumed will be q. As for an ordinary quantum signature scheme, the number of the encryption keys consumed could be p or q. If the verifiers calculate new encryption keys from the original ones, the number of the encryption keys consumed will be q. However, if the verifiers prepare a new signature key every time, the number will be p. Generally, p is much larger than q.

5 Conclusion

In this paper, we proposed a new quantum homomorphic signature scheme with repeatable verification, which can be used in general scenarios. A serial verification model was provided to solve the problem of signature verification for intermediate verifiers. A parallel verification model was provided to solve the problem of signature duplication for multiple terminal verifiers. These models will be beneficial to the signature verification of general scenarios. Scheme analysis shows that our scheme consumes much less quantum resource compared with ordinary quantum signature schemes.

Acknowledgement: This project was supported by the National Natural Science Foundation of China (No. 61571024) and the National Key Research and Development Program of China (No. 2016YFC1000307) for valuable helps.

References

Barnett, S. M.; Chefles, A.; Jex, I. (2003): Comparison of two unknown pure quantum states. *Physics Letters A*, vol. 307, no. 4, pp. 189-195.

Beige, A.; Englert, B.; Kurtsiefer, C. (2002): Secure communication with a publicly known key. *Acta Physica Polonica A*, vol. 101, pp. 357.

Curty, M.; Lutkenhaus, N. (2008): Comment on arbitrated quantum-signature scheme. *Physical Review A*, vol. 77, no. 4, pp. 1-4.

Filippov, S. N.; Ziman, M. (2012): Probability-based comparison of quantum states. *Physical Review A*, vol. 85, no. 85, 062301.

Gottesman, D.; Chuang, I. (2001): Quantum digital signatures. *Technical Report*.

Johnson, R.; Molnar, D.; Song, D. X.; Wagner, D. A. (2002): Homomorphic signature schemes. *RSA Conference, Lecture Notes in Computer Science*, vol. 2271, pp. 244-262.

Li, K.; Shang, T.; Liu, J. W. (2017): Continuous-variable quantum homomorphic signature. *Quantum Information Processing*, vol. 16, no. 10, pp. 246.

Liu, W. J.; Xu, Y.; Yang, C. N.; Gao, P. P.; Yu, W. B. (2018): An efficient and secure arbitrary n-party quantum key agreement protocol using bell states. *International Journal of Theoretical Physics*, vol. 57, no. 1, pp. 195-207.

Shang, T.; Li, K.; Liu, J. W. (2018): Measurement-device independency analysis of continuous-variable quantum digital signature. *Entropy*, vol. 20, no. 4, pp. 291.

Shang, T.; Zhao, X. J.; Wang, C.; Liu, J. W. (2015): Quantum homomorphic signature. *Quantum Information Processing*, vol. 14, no. 1, pp. 393-410.

Wang, M. M.; Yang, C.; Mousoli, R. (2018): Controlled cyclic remote state preparation of arbitrary qubit states. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 321-329.

Zeng, G. H. (2008): Reply to ‘comment on arbitrated quantum-signature scheme’. *Physical Review A*, vol. 77, no. 1, pp. 1-5.

Zeng, G. H.; Keitel, C. H. (2002): Arbitrated quantum-signature scheme. *Physical Review A*, vol. 65, no. 4, pp. 12-17.