Efficient quantum digital signatures over long distances with likely bit strings

Ji-Qian Qin,[1] Zong-Wen Yu,[2] and Xiang-Bin Wang[1,3,4,5,6,*]

[1]*State Key Laboratory of Low Dimensional Quantum Physics, Department of Physics, Tsinghua University, Beijing 100084, China*
[2]*Data Communication Science and Technology Research Institute, Beijing 100191, China*
[3]*Jinan Institute of Quantum Technology, SAICT, Jinan 250101, China*
[4]*Shanghai Branch, CAS Center for Excellence and Synergetic Innovation Center in Quantum Information and Quantum Physics, University of Science and Technology of China, Shanghai 201315, China*
[5]*Shenzhen Institute for Quantum Science and Engineering, and Physics Department, Southern University of Science and Technology, Shenzhen 518055, China*
[6]*Frontier Science Center for Quantum Information, Beijing 100084, China*

Quantum digital signatures (QDSs) can provide information-theoretic security of messages against forgery and repudiation. Compared with previous QDS protocols that focus on signing one-bit messages, hash function–based QDS protocols can save quantum resources and are able to sign messages of arbitrary length. Using the idea of likely bit strings, we propose an efficient QDS protocol with hash functions over long distances. Our method of likely bit strings can be applied to any quantum key distribution–based QDS protocol to significantly improve the signature rate and dramatically increase the secure signature distance of QDS protocols. In order to save computing resources, we propose an improved method where Alice participates in the verification process of Bob and Charlie. This eliminates the computational complexity relating to the huge number of all likely strings. We demonstrate the advantages of our method and our improved method with the example of sending-or-not-sending QDS. Under typical parameters, both our method and our improved method can improve the signature rate by more than 100 times and increase the signature distance by about 150 km compared with hash function–based QDS protocols without likely bit strings.

I. INTRODUCTION

Digital signature [1] is a useful technique for guaranteeing the security of message transfer among the signer and receivers, i.e., it can provide security of the message against forging and repudiation. Forgery means that a receiver accepts a message from a forger rather than from the signer, and repudiation means that the signer can repudiate his or her signature. Different from classical digital signatures, quantum digital signatures (QDSs) [2] can provide information-theoretic security that does not depend on computational complexity.

With the removal of the two assumptions of quantum memory and secure quantum channels [3–5], QDS becomes practicable [6–13]. Recently, there have been many breakthroughs [14–22] in quantum key distribution (QKD). Based on mature QKD devices, one can sign one-bit messages with bit strings generated by quantum communication of QKD [5]. The sending-or-not-sending

(SNS) protocol [19] can tolerate large misalignment error, while retaining measurement device–independent (MDI) security and a high key rate of twin-field (TF) QKD [18] on the scale of the square root of channel transmittance. Naturally, SNS QDS has higher efficiency over long distances [23,24] compared with BB84 (the quantum key distribution scheme developed by Bennett and Brassard in 1984) QDS and MDI QDS.

Compared with one-bit messages, QDS of multibit messages is more practicable. It would be inefficient to simply iterate a QDS based on one-bit messages to sign multibit messages. In classical digital signature, one can use the hash function method to sign multibit messages [25,26] with classical secure keys. Naturally, if we replace the classical secure keys here by the secure final keys from QKD, one can have a type of QDS with quantum security. In this type of implementation, the secure distance of QDS is just that of QKD. To further improve the secure distance and signature rate of QDS, here, using the idea of likely bit strings [5], we propose to use the raw key without any error correction or private amplification to sign multibit
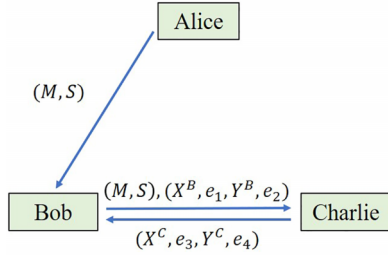
*xbwang@mail.tsinghua.edu.cn

FIG. 1. Our efficient QDS over long distances with likely bit strings. Alice is a signer and sends the message and the corresponding signature (M, S) to a receiver, Bob. Bob forwards (M, S) to another receiver Charlie, along with the bit strings and the corresponding bit-flip error rates (X^B, e_1, Y^B, e_2). Then, Charlie sends the bit strings and the corresponding bit-flip error rates (X^C, e_3, Y^C, e_4) to Bob. Bob (Charlie) determines whether to accept the message M based on the result of the comparison function $\mathbb{C}^B$ (11) ($\mathbb{C}^C$ (18)).

messages. Note that the method given in Ref. [5] is for one-bit messages, whereas we design a new hash function–based QDS protocol to sign messages of arbitrary length. Moreover, different from Refs. [27–29], which use final keys (with error correction and private amplification) as well as hash functions to sign and verify messages of arbitrary length, using the idea of likely bit strings, we perform the QDS protocol with raw keys as well as hash functions. We then propose an improved method where Alice participates in the verification process of Bob and Charlie. This improved method removes the computational complexity such as listing the huge number of hashing values of all likely strings.

II. QUANTUM DIGITAL SIGNATURE WITH LIKELY BIT STRINGS

Consider a QDS protocol with one signer and two receivers. As shown in Fig. 1, Alice is a signer and Bob and Charlie are two receivers. In order to avoid misunderstanding, we emphasize that, different from the usual use of Charlie for an unreliable relay's name in MDI QKD, here we use Charlie for one receiver's name. Alice signs the m-bit message M, and Bob (Charlie) decides whether to accept M based on the result of comparison between the actual digests and the expected digests, where the generation of digests requires hash functions [26] derived from associated bit strings among the signer and receivers. In our QDS protocol, instead of using final keys of QKD to generate hash functions, we use raw keys to generate hash functions.

Alice and Bob (Charlie) perform a key-generating protocol (KGP) to generate the associated bit strings X_A^B and X^B (X_A^C and X^C), where X_A^B (X_A^C) belongs to Alice and X^B (X^C) belongs to Bob (Charlie). The KGP is the quantum

communication of QKD, and there are many kinds, including BB84 [30], MDI [14–16,31], SNS TF [18,19], etc. After performing the KGP, Alice, Bob, and Charlie each have bit strings $\{X^A, X^B, X^C\}$ with n bits, where $X^A = X_A^B \oplus X_A^C$, and $\oplus$ denotes bit addition modulo 2. They use the same process to generate another set of associated bit strings $\{Y^A, Y^B, Y^C\}$ with $2n$ bits, where $Y^A = Y_A^B \oplus Y_A^C$ and Y_A^B, Y^B (Y_A^C, Y^C) are generated in KGP between Alice and Bob (Charlie). We denote the bit-flip error rate between bit strings X_A^B and X^B, Y_A^B and Y^B, X_A^C and X^C, and Y_A^C and Y^C as e_1, e_2, e_3, and e_4, respectively.

A. Signing stage

Alice uses local n-bit quantum random numbers p^A to randomly generate an irreducible polynomial p of degree n [25,28]. After that, based on the irreducible polynomial and the local bit string X^A, Alice generates a linear feedback shift register (LFSR)–based Toeplitz matrix [26] H^A with n rows and m columns, where n and m are the lengths of X^A and message M, respectively. Alice acts the matrix H^A on the message M to generate the hash value h,

$$H^A \cdot M = h, \tag{1}$$

where h is a vector of n bits. It should be noted that, in practice, the LFSR-based Toeplitz matrix and the hash value can be computed simultaneously in order to reduce the time consumption. In order to make it easier to understand, we present it in a step-by-step manner in this work.

Based on the above steps, Alice generates a digest D of message M,

$$D = (h, p^A), \tag{2}$$

where D is a $2n$-bit string consisting of h with n bits and p^A with n bits. Next, Alice encrypts the digest D with a bit string Y^A with $2n$ bits to generate the signature S of the message M,

$$S = D \oplus Y^A. \tag{3}$$

At the end of the signing stage, Alice sends the message along with the corresponding signature (M, S) to a receiver for verification.

B. Verification stage

The receivers Bob and Charlie generate a series of likely bit strings based on the local bit strings and the bit-flip error rates. They use these likely bit strings for verification.

1. Verification of Bob

As shown in Fig. 1, when Bob receives the message and the corresponding signature (M, S) from Alice, he forwards them to Charlie and sends the local bit strings

and the corresponding bit-flip error rates (X^B, e_1, Y^B, e_2) to Charlie. Then, Charlie sends the local bit string and the corresponding bit-flip error rates (X^C, e_3, Y^C, e_4) to Bob.

Based on the bit strings X^B, X^C and the corresponding bit-flip error rates e_1, e_3, Bob generates a series of likely bit strings $\{K_i^{XB}\}$ locally,

$$\{K_i^{XB}\} = f(X^B \oplus X^C, e_1, e_3), \tag{4}$$

where $i = 1, 2, \ldots, n_x$, and n_x is the number of likely bit strings,

$$n_x = 2^{nH_2(E_x)}, \tag{5}$$

in which $H_2(\cdot)$ is the binary Shannon entropy. The function $f(X^B \oplus X^C, e_1, e_3)$ means that, based on the bit string $X^B \oplus X^C$ and the corresponding bit-flip error rate E_x, which has a maximum value of $(e_1 + e_3)$, Bob can generate likely bit strings $\{K_i^{XB}\}$ relative to X^A, i.e., $X^A \in \{K_i^{XB}\}$.

Similarly, Bob can generate a series of likely bit strings $\{K_j^{YB}\}$ relative to Y^A,

$$\{K_j^{YB}\} = f(Y^B \oplus Y^C, e_2, e_4), \tag{6}$$

where $j = 1, 2, \ldots, n_y$, and n_y is the number of likely bit strings,

$$n_y = 2^{2nH_2(E_y)}. \tag{7}$$

The function $f(Y^B \oplus Y^C, e_2, e_4)$ means that, based on the bit string $Y^B \oplus Y^C$ and the corresponding bit-flip error rate E_y, which has a maximum value of $(e_2 + e_4)$, Bob can generate likely bit strings $\{K_j^{YB}\}$ relative to Y^A, i.e., $Y^A \in \{K_j^{YB}\}$. Of course, the bit-flip error rates E_x and E_y must satisfy $E_x, E_y \in (0, 0.5)$.

After generating the likely bit strings locally, Bob decrypts the signature S with likely bit strings $\{K_j^{YB}\}$ to obtain n_y expected digests,

$$S \oplus \{K_j^{YB}\} = \{(h_j^B, p_j^B)\}, \tag{8}$$

where $j = 1, 2, \ldots, n_y$.

Bob then judges whether the expected hash values received from Alice have been generated from one likely bit string according to the expected hash values he received and his own raw keys. Bob can do this job in the following way, for example: Bob generates $n_x n_y$ LFSR-based Toeplitz matrices [26] $\{H_{i,j}^B\}$ based on likely bit strings $\{K_i^{XB}\}$ and $\{p_j^B\}$. After that, Bob performs $n_x n_y$ LFSR-based Toeplitz matrices $\{H_{i,j}^B\}$ on the message M to obtain

$n_x n_y$ actual hash values,

$$\{H_{i,j}^B\} \cdot M = \{\mathbb{h}_{i,j}^B\}, \tag{9}$$

which together with $\{p_j^B\}$ form actual digests,

$$\{(\mathbb{h}_{i,j}^B, p_j^B)\}. \tag{10}$$

Finally, Bob compares expected digests $\{(h_j^B, p_j^B)\}$ (8) with actual digests $\{(\mathbb{h}_{i,j}^B, p_j^B)\}$ (10) using the following comparison function $\mathbb{C}^B$,

$$\mathbb{C}^B = \begin{cases} 1 & \exists \, C_{i,j}^B = 1, \\ 0 & \text{else}, \end{cases} \tag{11}$$

where $C_{i,j}^B$ is

$$C_{i,j}^B = \begin{cases} 1 & h_j^B = \mathbb{h}_{i,j}^B, \\ 0 & \text{else}. \end{cases} \tag{12}$$

Obviously, this comparison function implies that Bob makes at most $n_x n_y$ comparisons. If $\mathbb{C}_B = 1$, Bob accepts the message M; otherwise, Bob rejects the message M.

2. Verification of Charlie

Charlie uses similar steps to generate the following two sets of likely bit strings $\{K_i^{XC}\}$ and $\{K_j^{YC}\}$,

$$\{K_i^{XC}\} = f(X^B \oplus X^C, e_1, e_3), \tag{13}$$

$$\{K_j^{YC}\} = f(Y^B \oplus Y^C, e_2, e_4), \tag{14}$$

where $i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_y$. Then, Charlie decrypts the signature S with $\{K_j^{YC}\}$ to obtain n_y likely expected digests,

$$S \oplus \{K_j^{YC}\} = \{(h_j^C, p_j^C)\}. \tag{15}$$

Charlie then judges whether the expected hash values received from Bob have been generated from one likely bit string according to the expected hash values he received and his own raw keys. Charlie can do this job in the following way, for example: Based on likely bit strings $\{K_i^{XC}\}$ and $\{p_j^C\}$, Charlie can obtain $n_x n_y$ LFSR-based Toeplitz matrices [26] $\{H_{i,j}^C\}$. After that, Charlie performs $\{H_{i,j}^C\}$ on the message M to obtain $n_x n_y$ actual hash values,

$$\{H_{i,j}^C\} \cdot M = \{\mathbb{h}_{i,j}^C\}, \tag{16}$$

which together with $\{p_j^C\}$ form the actual digests,

$$\{(\mathbb{h}_{i,j}^C, p_j^C)\}. \tag{17}$$

Finally, Charlie compares expected digests $\{(h_j^C, p_j^C)\}$ (15) with actual digests $\{(\mathbb{h}_{i,j}^C, p_j^C)\}$ (17) using the following

comparison function $\mathbb{C}^C$,

$$\mathbb{C}^C = \begin{cases} 1 & \exists \, C_{i,j}^C = 1, \\ 0 & \text{else}, \end{cases} \tag{18}$$

where $C_{i,j}^C$ is

$$C_{i,j}^C = \begin{cases} 1 & h_j^C = \mathbb{h}_{i,j}^C, \\ 0 & \text{else}. \end{cases} \tag{19}$$

Obviously, this comparison function implies that Charlie makes at most $n_x n_y$ comparisons. If $\mathbb{C}_C = 1$, Charlie accepts the message M; otherwise, Charlie rejects the message M.

III. SECURITY ANALYSIS

In the QDS protocol with three participants, at most one participant is malicious due to the majority principle. We analyze the security of our QDS protocol in terms of forgery attacks, repudiation attacks, and robustness [5,28].

First, we consider that Bob is malicious, i.e., he wants to launch a forgery attack. A successful forgery of Bob means that he can make Charlie accept a tampered message M', where $M' \neq M$. In our protocol, Bob needs to send the message and the corresponding signature (M, S) to Charlie before Charlie sends X^C and Y^C to Bob. Therefore, if Bob successfully guesses X^C and Y^C, then, combined with X^B and Y^B, he can tamper with the message M for Charlie to accept M'. The probability of successfully guessing the bit string X^C and Y^C in Charlie's hand is

$$p_g \leq 2^{-nH_2(p_e)}, \tag{20}$$

where p_e is the minimum rate at which Bob can make errors when guessing X^C and satisfies

$$H_2(p_e) = \Delta_1[1 - H_2(e^{\text{ph}})], \tag{21}$$

in which $H_2(\cdot)$ is the binary Shannon entropy, and Δ_1 and e^{ph} are the proportion of effective bits of a single-photon state in X^C and the phase-flip error rate of X^C, respectively. We define the bits generated in time windows of KGP where the clicks of detectors satisfy certain conditions as effective bits. For example, in SNS KGP [19], a bit generated in a time window with only one detector clicks is an effective bit.

In addition to guessing the bit strings of Charlie, Bob has another method of forgery, which is to directly construct a set of tampered messages and corresponding signatures (M', S') for Charlie to accept, where $M' \neq M$ and $S' \neq S$ or $S' = S$. Because of the collision probability of LFSR-based Toeplitz matrices and the characteristics of our comparison function, Charlie may accept the tampered

message M' with a very small probability. The collision refers to the situation where different messages correspond to the same hash value. The collision probability of LFSR-based Toeplitz matrices is at most $m/2^{n-1}$ [26], where m is the length of the message and n is the degree of irreducible polynomials used to generate the LFSR-based Toeplitz matrices. In our protocol, Charlie decides whether to accept the message based on the comparison function $\mathbb{C}_C$ (18), which compares the expected digests with the actual digests at most $n_x n_y$ times. For one fixed set of (i, j), the upper bound of the probability that the expected hash value $h_j^{C,M'}$ and actual hash value $\mathbb{h}_{i,j}^{C,M'}$ of M' satisfy $h_j^{C,M'} = \mathbb{h}_{i,j}^{C,M'}$ is $m/2^{n-1}$. For other $(n_x n_y - 1)$ sets of (i, j), the upper bound of the probability that the hash values of M' satisfy $h_j^{C,M'} = \mathbb{h}_{i,j}^{C,M'}$ is $(m/(2^{n-1}))^2$. According to our comparison function, Charlie will reject M' only if the results of all $n_x n_y$ comparisons are not equal. As long as there is one equal comparison result, Charlie will accept M'. Therefore, the probability that Charlie accepts (M', S') is

$$p_h \leq 1 - \left(1 - \frac{m}{2^{n-1}}\right)\left[1 - \left(\frac{m}{2^{n-1}}\right)^2\right]^{n_x n_y - 1}. \tag{22}$$

Combining the above two scenarios, the probability of Bob's successful forgery is

$$p_f = \{p_g, p_h\}_{\max}. \tag{23}$$

Next, we consider that Alice is malicious, i.e., she wants to deny her signature. If Alice can make Bob accept the message M and Charlie reject M, then she can deny the signed message, which is a successful repudiation attack. In our protocol, when Bob and Charlie are both honest, they generate the same possible bit strings locally, i.e., $\{K_i^{XB}\} = \{K_i^{XC}\}$ and $\{K_j^{YB}\} = \{K_j^{YC}\}$. Then, they generate the same LFSR-based Toeplitz matrices locally as well. Therefore, if Bob accepts the message M, then Charlie must also accept M. So, the probability of successful repudiation is

$$p_{re} = 0. \tag{24}$$

Finally, we need to analyze the robustness of our QDS protocol, i.e., the case where Bob rejects the message M if all participants are honest. In this case, the likely bit strings $\{K_i^{XB}\}$ and $\{K_j^{YB}\}$ generated by Bob must contain X^A and Y^A, respectively, i.e., $X^A \in \{K_i^{XB}\}$ and $Y^A \in \{K_j^{YB}\}$. According to our comparison function $\mathbb{C}_B$ (11), Bob must accept the message M. Therefore, the probability that Bob rejects the message when all participants are honest is

$$p_{ro} = 0. \tag{25}$$

Based on the above analysis, we can define ε as the security level [5] of our QDS protocol:

Definition 1 (ε-secure).—If ε satisfies

$$\varepsilon = \{p_f, p_{re}, p_{ro}\}_{\max}, \tag{26}$$

then we say that the security level of the QDS protocol is ε, or that the QDS protocol is ε-secure.

IV. IMPROVED METHOD

In our protocol, we have given an example of the way for receivers to verify that the expected hash values received from Alice have been generated from one likely bit string, by listing all possible hash values from likely strings. In order to save computing resources, here we propose an improved method where Alice is involved in the verification process. The improved method does not require large complex computing relating to the huge number of likely strings while keeping the good performance of QDS in terms of the signature rate and distance without consuming large computing resources.

A. Improved protocol

In the signing stage of the protocol, Alice follows the steps described in Sec. II to generate the signature S of message M. At the end of the signing stage, Alice sends (M, S) to a receiver for verification.

In the verification stage, Bob forwards (M, S) received from Alice to Charlie and also sends the bit strings (X^B, Y^B) to Charlie. Then Bob informs Alice that he has received (M, S). Next, Charlie sends the bit strings (X^C, Y^C) to Bob and informs Alice that he has received (M, S). Alice separately verifies whether both Bob and Charlie have received (M, S). If both Bob and Charlie have received (M, S), then Alice publishes (X^A, Y^A); otherwise, the protocol is terminated.

Bob verifies the message M based on the local bit strings and the information published by Alice. Bob obtains the expected digest D_B^E using Y^A and S,

$$D_B^E = (h^B, p^B) = S \oplus Y^A. \tag{27}$$

Next, based on X^A and p^B, Bob generates the LFSR-based Toeplitz matrix H^B and acts it on the message M to obtain the actual hash value

$$H^B \cdot M = \mathbb{h}^B. \tag{28}$$

If $\mathbb{h}^B = h^B$, Bob accepts the message M; otherwise, Bob rejects the message M.

Charlie uses similar steps for verification. Charlie obtains the expected digest D_C^E using Y^A and S,

$$D_C^E = (h^C, p^C) = S \oplus Y^A. \tag{29}$$

Next, based on X^A and p^C, Charlie generates the LFSR-based Toeplitz matrix H^C and acts it on the message M to obtain the actual hash value

$$H^C \cdot M = \mathbb{h}^C. \tag{30}$$

If $\mathbb{h}^C = h^C$, Charlie accepts the message M; otherwise, Charlie rejects the message M.

B. Security analysis of improved protocol

We can perform a similar security analysis of the improved protocol in terms of unforgeability, nonrepudiation, and robustness using the methods in Sec. III. In our improved protocol, Alice will not publish (X^A, Y^A) until she confirms that both Bob and Charlie have received (M, S). That is, Bob does not know the bit string (X^A, Y^A) before he sends the message to Charlie. Therefore, for Bob, the probability that he makes Charlie accept the forged message is still p_f (23), i.e., either due to successfully guessing Charlie's local bit string X^C, Y^C or due to the collision probability of LFSR-based Toeplitz matrices. In terms of nonrepudiation, since both Bob and Charlie in the improved protocol use (X^A, Y^A) published by Alice in their verification stage, their results of verification are consistent. Therefore, the probability of successful repudiation is still p_{re} (24). In terms of robustness, when all participants are honest, Alice will publish bit (X^A, Y^A) after she has confirmed that both Bob and Charlie have received (M, S). Then, Bob will obtain $\mathbb{h}^B = h^B$ using the bit string published by Alice. Thus Bob will accept the message M when all participants are honest, i.e., the probability associated with robustness is still p_{ro} (25). In general, the expression for the security level ε of the improved protocol is the same as that (26) of the protocol introduced in Sec. II.

Compared with our protocol in Sec. II, our improved protocol can significantly reduce the computing resources without compromising security and efficiency. In our improved protocol, instead of using the old verification method, which may need large computing resources, such as listing all possible hash values from likely strings, Bob and Charlie utilize the bit strings published by Alice for verification, which completely eliminates the need for large computing resources.

V. DISCUSSION

Our protocol can be applied to any QKD-based QDS protocol. Using existing mature QKD devices, based on the idea of likely bit strings, we can implement efficient QDS over longer distances. We use the signature rate [23, 24] R to describe the efficiency of the QDS protocol,

$$R = \frac{m}{2N}, \tag{31}$$

where m is the length of the message M and N is the total number of pulses in the KGP between Alice and Bob, i.e.,

TABLE I. System parameters used in our numerical simulation. Notation: α is the loss coefficient of the fiber; e_d is the misalignment error rate; p_d is the dark count rate of the detectors; η_d is the efficiency of the detectors; ξ and ε_p are failure probabilities in parameter estimation, which are described in detail in the Appendix; and ε is the security level of our QDS protocol.

α	e_d	p_d	η_d	ξ	ε_p	ε
0.2 dB/km	2%	10^{-8}	50%	10^{-12}	10^{-10}	10^{-10}

the whole process of generating X_A^B and Y_A^B. The total number of pulses in the KGP between Alice and Charlie, i.e., the whole process of generating X_A^C and Y_A^C, in the symmetric case is also N. Therefore, Eq. (31) has a factor of 2 in the denominator.

We demonstrate the advantages of our method and improved method using QDS with SNS KGP [19] as an example. Alice performs the quantum communication of SNS QKD, i.e., SNS KGP, with Bob and Charlie, respectively, to generate the associated bit strings. After the associated bit strings are generated among the participants, the signing and verification steps can be performed as described in Secs. II and IV. In practical applications of QDS, the total number of pulses is finite. We need to estimate the parameters while considering the effects of finite data size. In our QDS protocol, as well as security analysis, we need to obtain the bit-flip error rate E, the proportion of effective bits of a single-photon state Δ_1, and the phase-flip error rate e^{ph} of the associated bit strings. In the Appendix, we describe in detail how to obtain these parameters.

In Fig. 2, considering the effects of finite data size, we compare the signature rate with our method, our improved method, and the method using final keys. The solid line indicates the signature rate obtained by our method and our improved method, and the broken line indicates the signature rate obtained by the method of final keys. As can be seen in Fig. 2, both our method and our improved method can significantly improve the signature rate and dramatically increase the secure distance of QDS. The signature rate can be improved by more than 100 times, and the secure distance can be increased by about 150 km using our method or our improved method under the system parameters in Table I and the length of the message $m = 10^{20}$. It should be emphasized that, as introduced in Sec. IV, the improved method can improve the performance of QDS in terms of signature rate and distance as well as save computing resources.

During the implementation of our protocol, the actual number of comparisons n_0 that the receiver needs to perform satisfies $n_0 \le n_x n_y$. This is because, according to the characteristics of our comparison functions $\mathbb{C}^B$ (11) and $\mathbb{C}^C$ (18), as soon as an equal set of expected and actual digests is found, the receiver can stop the subsequent comparisons and accept the message M.
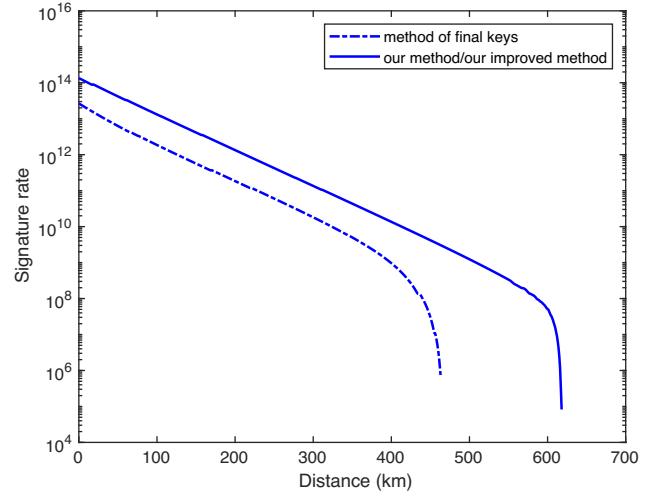

FIG. 2. Comparison of the signature rate with our method, our improved method, and the method of final keys. The solid line represents the signature rate of our method and our improved method, and the broken line represents the signature rate using the method of final keys. Under the system parameters shown in Table I and the length of the message $m = 10^{20}$, both our method and our improved method can improve the signature rate by more than 100 times, and increase the secure distance of QDS by about 150 km.

In this work, we use LFSR Toeplitz hash functions [26] in our QDS protocol. Of course, our method of using likely bit strings for QDS can be extended to the case of using other types of hash functions. For protocols that use other types of hash functions, the corresponding security analysis can be performed using similar ideas based on the collision probability of hash functions as in this work.

In order to reduce the consumption of computing resources, we propose an improved protocol based on our protocol in Sec. II. In our improved protocol, Alice is involved in the verification process of Bob and Charlie. On the one hand, Alice publishes the bit strings (X^A, Y^A) after confirming that both Bob and Charlie have received (M, S). Therefore, this does not pose a threat to security. On the other hand, Bob and Charlie can directly use (X^A, Y^A) published by Alice for verification, which no longer need to consume large computing resources.

In conclusion, based on the idea of likely bit strings, we propose an efficient and practicable QDS protocol over long distances. Our method only requires mature QKD devices, while improving the performance of QDS in terms of signature rate and distance. With typical parameters, our method of likely bit strings can improve the signature rate by more than 100 times and increase the distance of QDS by about 150 km compared with a hash function–based QDS protocol using final keys. To save computing resources, we propose an improved method, which can significantly improve the signature rate and distance of QDS without large computing resources.

APPENDIX: SENDING-OR-NOT-SENDING KEY GENERATION PROTOCOL

We take the QDS with SNS KGP as an example to demonstrate the advantages of our method of likely bit strings. We generate the associated bit strings by the quantum communication of SNS QKD [19,32,33].

In each time window, Alice (Bob) chooses the window as a signal window or decoy window with probability p_z or $(1 - p_z)$. In the signal window, Alice (Bob) writes down the bit value 1 (0) when sending the phase-randomized coherent state of intensity μ to Eve with probability q and writes down the bit value 0 (1) when not sending the coherent state, i.e., sending the vacuum state, to Eve with probability $(1 - q)$. In the decoy window, Alice (Bob) randomly sends phase-randomized coherent states of intensities $\mu_0 = 0$, μ_1, and μ_2, with probability p_0, p_1, and $(1 - p_0 - p_1)$, to Eve. Eve performs measurements using beam splitters and announces the outcome of measurements. We can record the event where one and only one detector clicks as an effective event. The corresponding time window of an effective event is an effective window and the corresponding bit is an effective bit. The effective bits used for parameter estimation are discarded. And then, Alice and Bob obtain the associated bit strings of raw keys.

The receivers can construct likely bit strings based on the bit strings of raw keys and the bit-flip error rates and use these likely bit strings for verification. In the following, we show how to obtain the bit-flip error rate E, the proportion of effective bits of a single-photon state Δ_1, and the phase-flip error rate e^{ph} of the associated bit strings taking into account the effects of finite data size.

Effective events can be divided into three categories with counting rates S_C, S_D, and S_V, respectively, with

$$
\begin{aligned}
S_C &= 2[(1 - p_d)e^{-\eta\mu/2} - (1 - p_d)^2 e^{-\eta\mu}], \\
S_D &= 2[(1 - p_d)e^{-\eta\mu}I_0(\eta\mu) - (1 - p_d)^2 e^{-2\eta\mu}], \quad \text{(A1)} \\
S_V &= 2p_d(1 - p_d),
\end{aligned}
$$

where S_C corresponds to the case where one and only one participant chooses to send and the other participant chooses not to send, and S_D and S_V represent the case where both participants choose to send and both participants choose not to send, respectively. The notation p_d represents the dark count rate of the detectors, $\eta = 10^{-\alpha l/10}\eta_d$ is the total system efficiency, α is the loss coefficient of the fiber, l is half of the distance between the two participants

(we assume that the node used for measurement in QDS with SNS KGP is midway between the two participants), η_d represents the detection efficiency, μ is the intensity of the coherent state, and $I_0(x)$ is the zeroth-order hyperbolic Bessel function of the first kind.

Based on the above three counting rates S_C, S_D, and S_V, we can obtain their effective event numbers n_C, n_D, and n_V as follows:

$$
\begin{aligned}
n_C &= 2Np_z^2 q(1 - q)S_C, \\
n_D &= Np_z^2 q^2 S_D, \quad \text{(A2)} \\
n_V &= Np_z^2 (1 - q)^2 S_V.
\end{aligned}
$$

Here N is the total number of pluses in SNS KGP between two participants, p_z is the probability that the participant chooses the time window as a signal window (i.e., the participant chooses the time window as a decoy window with probability $(1 - p_z)$), and q represents the probability that the participant chooses to send.

Therefore, the bit-flip error rate E_T of the bit strings generated by SNS KGP is

$$
E_T = \frac{n_D + n_V}{N_t}, \quad \text{(A3)}
$$

where N_t is the total number of three effective events

$$
N_t = n_C + n_D + n_V. \quad \text{(A4)}
$$

It should be noted that, in experiments, the bit-flip error rate E_T can be directly observed. In our numerical simulations, we use the expression (A3) of E_T. We randomly select T (in our simulation, we set $T = \gamma N_t$ and $\gamma = 10\%$) bits to obtain their bit-flip error rate, which is used to estimate the bit-flip error rate E of the remaining bits [5,8,34],

$$
E \leq E_T + \mu(n, T, \varepsilon_p), \quad \text{(A5)}
$$

with

$$
\mu(n, T, \varepsilon_p) = \sqrt{\frac{(n - T + 1)\ln(1/\varepsilon_p)}{2nT}}, \quad \text{(A6)}
$$

where ε_p is the failure probability of this estimation, and n is the length of X^A,

$$
n = \left\lfloor \frac{N_t - T}{3} \right\rfloor. \quad \text{(A7)}
$$

Next, we analyze how to obtain the proportion Δ_1 of effective bits of a single-photon state and phase-flip error rate e^{ph} of the bit string X^C, which are used in the calculation of p_e (21).

We use the Chernoff bound [35] to estimate the expected values ϕ from the observed values, which can be obtained directly in experiments, and then we use the Chernoff bound to estimate the real values φ of a specific experiment from the expected values. The lower and upper bounds of the expected value are denoted by ϕ^L and ϕ^U, respectively,

$$
\phi^L(X) = \frac{X}{1 + \delta_1(X)},
$$
$$
\phi^U(X) = \frac{X}{1 - \delta_2(X)},
$$
(A8)

and $\delta_1(X)$ and $\delta_2(X)$ satisfy

$$
\left(\frac{e^{\delta_1}}{(1 + \delta_1)^{1+\delta_1}} \right)^{X/(1+\delta_1)} = \frac{\xi}{2},
$$
$$
\left(\frac{e^{-\delta_2}}{(1 - \delta_2)^{1-\delta_2}} \right)^{X/(1-\delta_2)} = \frac{\xi}{2},
$$
(A9)

where ξ denotes the failure probability of estimation.

The lower and upper bounds of the real value are denoted by φ^L and φ^U, respectively,

$$
\varphi^L(Y) = [1 + \delta_1'(Y)]Y,
$$
$$
\varphi^U(Y) = [1 - \delta_2'(Y)]Y,
$$
(A10)

and $\delta_1'(X)$ and $\delta_2'(X)$ satisfy

$$
\left(\frac{e^{\delta_1'}}{(1 + \delta_1')^{1+\delta_1'}} \right)^{Y} = \frac{\xi}{2},
$$
$$
\left(\frac{e^{-\delta_2'}}{(1 - \delta_2')^{1-\delta_2'}} \right)^{Y} = \frac{\xi}{2},
$$
(A11)

where ξ represents the failure probability of estimation.

Based on the analysis in SNS QKD [19,32,33], we can obtain the lower bound of the expected value of the counting rate of single-photon state $\langle s_1^L \rangle$ and the upper bound of the expected value of the phase-flip error rate $\langle e^{\mathrm{ph},U} \rangle$,

$$
\langle s_1 \rangle \geq \langle s_1^L \rangle = \frac{1}{2}(\langle s_{01}^L \rangle + \langle s_{10}^L \rangle),
$$
(A12)

where $\langle s_{01}^L \rangle$ and $\langle s_{10}^L \rangle$ satisfy

$$
\langle s_{01}^L \rangle = \frac{\mu_2^2 e^{\mu_1} \langle S_{01}^L \rangle - \mu_1^2 e^{\mu_2} \langle S_{02}^U \rangle - (\mu_2^2 - \mu_1^2) \langle S_{00}^U \rangle}{\mu_1 \mu_2 (\mu_2 - \mu_1)},
$$
$$
\langle s_{10}^L \rangle = \frac{\mu_2^2 e^{\mu_1} \langle S_{10}^L \rangle - \mu_1^2 e^{\mu_2} \langle S_{20}^U \rangle - (\mu_2^2 - \mu_1^2) \langle S_{00}^U \rangle}{\mu_1 \mu_2 (\mu_2 - \mu_1)},
$$
(A13)

and

$$
\langle e^{\mathrm{ph}} \rangle \leq \langle e^{\mathrm{ph},U} \rangle = \frac{\langle T_\Delta^U \rangle - \frac{1}{2} e^{-2\mu_1} \langle S_{00}^L \rangle}{2\mu_1 e^{-2\mu_1} \langle s_1^L \rangle}.
$$
(A14)

The notation S_{ij}, $i,j \in \{0,1,2\}$, in the above equations denotes the counting rate of source $\mu_i \mu_j$ when participants choose coherent state μ_i and μ_j to send. In the decoy window, participants randomly select three different intensities $\mu_0 = 0$, μ_1, and μ_2 of phase-randomized coherent states with probability p_0, p_1, and $(1 - p_0 - p_1)$, respectively. The notation $\langle T_\Delta^U \rangle$ can be calculated by

$$
\langle T_\Delta \rangle \leq \langle T_\Delta^U \rangle = \frac{\phi^U(n_{\Delta+}^l + n_{\Delta-}^r)}{2N_{\Delta\pm}},
$$
(A15)

where $n_{\Delta+}^l$ and $n_{\Delta-}^r$ represent effective events heralded by the left and right detector, respectively,

$$
n_{\Delta+}^l = n_{\Delta-}^r = [T_X(1 - 2e_d) + e_d S_X]N_{\Delta\pm},
$$
(A16)

$$
N_{\Delta\pm} = \frac{\Delta}{2\pi}(1 - p_z)^2 p_1^2 N,
$$
(A17)

and

$$
T_X = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} (1 - p_d) e^{-2\eta\mu_1 \cos^2(\delta/2)} \, d\delta
$$
$$
- (1 - p_d)^2 e^{-2\eta\mu_1},
$$
(A18)

$$
S_X = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} (1 - p_d) e^{-2\eta\mu_1 \sin^2(\delta/2)} \, d\delta
$$
$$
- (1 - p_d)^2 e^{-2\eta\mu_1} + T_X.
$$
(A19)

We set $\Delta = \pi/15$ in our simulation.

Based on $\langle s_1^L \rangle$ (A12) and $\langle e^{\mathrm{ph},U} \rangle$ (A14), we can use the Chernoff bound to estimate the real values Δ_1 and e^{ph} from the expected values,

$$
\Delta_1 = \frac{\varphi^L(n\langle \Delta_1^L \rangle)}{n},
$$
(A20)

$$
e^{\mathrm{ph}} = \frac{\varphi^U(n\Delta_1 \langle e^{\mathrm{ph},U} \rangle)}{n\Delta_1},
$$
(A21)

where

$$
\langle \Delta_1^L \rangle = \frac{2Np_z^2 q(1 - q)\mu e^{-\mu} \langle s_1^L \rangle}{N_t}.
$$
(A22)

According to the bit-flip error rate E (A5), the proportion of effective bits of a single-photon state Δ_1 (A20),

and the phase-flip error rate e^{ph} (A21), we can optimize the signature rate R by selecting the appropriate $(N, \mu, \mu_1, \mu_2, q, p_z, p_0, p_1)$ given security level ε and other system parameters.

---

[1] W. Diffie and M. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory **22**, 644 (1976).

[2] D. Gottesman and I. Chuang, Quantum digital signatures, arXiv:0105032 (2001).

[3] V. Dunjko, P. Wallden, and E. Andersson, Quantum Digital Signatures Without Quantum Memory, Phys. Rev. Lett. **112**, 040502 (2014).

[4] H.-L. Yin, Y. Fu, and Z.-B. Chen, Practical quantum digital signature, Phys. Rev. A **93**, 032316 (2016).

[5] R. Amiri, P. Wallden, A. Kent, and E. Andersson, Secure quantum signatures using insecure quantum channels, Phys. Rev. A **93**, 032325 (2016).

[6] R. J. Donaldson, R. J. Collins, K. Kleczkowska, R. Amiri, P. Wallden, V. Dunjko, J. Jeffers, E. Andersson, and G. S. Buller, Experimental demonstration of kilometer-range quantum digital signatures, Phys. Rev. A **93**, 012329 (2016).

[7] R. J. Collins, R. Amiri, M. Fujiwara, T. Honjo, K. Shimizu, K. Tamaki, M. Takeoka, E. Andersson, G. S. Buller, and M. Sasaki, Experimental transmission of quantum digital signatures over 90 km of installed optical fiber using a differential phase shift quantum key distribution system, Opt. Lett. **41**, 4883 (2016).

[8] I. V. Puthoor, R. Amiri, P. Wallden, M. Curty, and E. Andersson, Measurement-device-independent quantum digital signatures, Phys. Rev. A **94**, 022328 (2016).

[9] G. Roberts, M. Lucamarini, Z. Yuan, J. Dynes, L. Comandar, A. Sharpe, A. Shields, M. Curty, I. Puthoor, and E. Andersson, Experimental measurement-device-independent quantum digital signatures, Nat. Commun. **8**, 1 (2017).

[10] H.-L. Yin, Y. Fu, H. Liu, Q.-J. Tang, J. Wang, L.-X. You, W.-J. Zhang, S.-J. Chen, Z. Wang, Q. Zhang, et al., Experimental quantum digital signature over 102 km, Phys. Rev. A **95**, 032334 (2017).

[11] H.-L. Yin, W.-L. Wang, Y.-L. Tang, Q. Zhao, H. Liu, X.-X. Sun, W.-J. Zhang, H. Li, I. V. Puthoor, L.-X. You, et al., Experimental measurement-device-independent quantum digital signatures over a metropolitan network, Phys. Rev. A **95**, 042338 (2017).

[12] C.-H. Zhang, X.-Y. Zhou, H.-J. Ding, C.-M. Zhang, G.-C. Guo, and Q. Wang, Proof-of-principle demonstration of passive decoy-state quantum digital signatures over 200 km, Phys. Rev. Appl. **10**, 034033 (2018).

[13] X.-B. An, H. Zhang, C.-M. Zhang, W. Chen, S. Wang, Z.-Q. Yin, Q. Wang, D.-Y. He, P.-L. Hao, S.-F. Liu, et al., Practical quantum digital signature with a gigahertz BB84 quantum key distribution system, Opt. Lett. **44**, 139 (2019).

[14] S. L. Braunstein and S. Pirandola, Side-Channel-Free Quantum Key Distribution, Phys. Rev. Lett. **108**, 130502 (2012).

[15] H.-K. Lo, M. Curty, and B. Qi, Measurement-Device-Independent Quantum Key Distribution, Phys. Rev. Lett. **108**, 130503 (2012).

[16] Y.-H. Zhou, Z.-W. Yu, and X.-B. Wang, Making the decoy-state measurement-device-independent quantum key distribution practically useful, Phys. Rev. A **93**, 042324 (2016).

[17] Y.-H. Li, S.-L. Li, X.-L. Hu, C. Jiang, Z.-W. Yu, W. Li, W.-Y. Liu, S.-K. Liao, J.-G. Ren, H. Li, L. You, Z. Wang, J. Yin, F. Xu, Q. Zhang, X.-B. Wang, Y. Cao, C.-Z. Peng, and J.-W. Pan, Free-Space and Fiber-Integrated Measurement-Device-Independent Quantum Key Distribution under High Background Noise, Phys. Rev. Lett. **131**, 100802 (2023).

[18] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, Overcoming the rate-distance limit of quantum key distribution without quantum repeaters, Nature **557**, 400 (2018).

[19] X.-B. Wang, Z.-W. Yu, and X.-L. Hu, Twin-field quantum key distribution with large misalignment error, Phys. Rev. A **98**, 062323 (2018).

[20] S. Wang, Z.-Q. Yin, D.-Y. He, W. Chen, R.-Q. Wang, P. Ye, Y. Zhou, G.-J. Fan-Yuan, F.-X. Wang, W. Chen, et al., Twin-field quantum key distribution over 830-km fibre, Nat. Photonics **16**, 154 (2022).

[21] Y. Liu, W.-J. Zhang, C. Jiang, J.-P. Chen, C. Zhang, W.-X. Pan, D. Ma, H. Dong, J.-M. Xiong, C.-J. Zhang, et al., Experimental Twin-Field Quantum Key Distribution over 1000 km Fiber Distance, Phys. Rev. Lett. **130**, 210801 (2023).

[22] G.-J. Fan-Yuan, F.-Y. Lu, S. Wang, Z.-Q. Yin, D.-Y. He, W. Chen, Z. Zhou, Z.-H. Wang, J. Teng, G.-C. Guo, et al., Robust and adaptable quantum key distribution network without trusted nodes, Optica **9**, 812 (2022).

[23] C.-H. Zhang, X. Zhou, C.-M. Zhang, J. Li, and Q. Wang, Twin-field quantum digital signatures, Opt. Lett. **46**, 3757 (2021).

[24] J.-Q. Qin, C. Jiang, Y.-L. Yu, and X.-B. Wang, Quantum digital signatures with random pairing, Phys. Rev. Appl. **17**, 044047 (2022).

[25] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography* (CRC Press, Boca Raton, 1997).

[26] H. Krawczyk, in *Annual International Cryptology Conference* (Springer, Berlin, 1994), p. 129.

[27] R. Amiri, A. Abidin, P. Wallden, and E. Andersson, in *International Conference on Applied Cryptography and Network Security* (Springer, Leuven, 2018), p. 143.

[28] H.-L. Yin, Y. Fu, C.-L. Li, C.-X. Weng, B.-H. Li, J. Gu, Y.-S. Lu, S. Huang, and Z.-B. Chen, Experimental quantum secure network with digital signatures and encryption, Natl. Sci. Rev. **10**, nwac228 (2023).

[29] E. O. Kiktenko, A. Zelenetsky, and A. K. Fedorov, Practical quantum multiparty signatures using quantum-key-distribution networks, Phys. Rev. A **105**, 012408 (2022).

[30] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, Theor. Comput. Sci. **560**, 7 (2014). theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84.

[31] C. Jiang, Z.-W. Yu, X.-L. Hu, and X.-B. Wang, Higher key rate of measurement-device-independent quantum key

distribution through joint data processing, Phys. Rev. A **103**, 012402 (2021).

[32] Z.-W. Yu, X.-L. Hu, C. Jiang, H. Xu, and X.-B. Wang, Sending-or-not-sending twin-field quantum key distribution in practice, Sci. Rep. **9**, 3080 (2019).

[33] C. Jiang, Z.-W. Yu, X.-L. Hu, and X.-B. Wang, Unconditional security of sending or not sending twin-field quantum

key distribution with finite pulses, Phys. Rev. Appl. **12**, 024061 (2019).

[34] R. J. Serfling, Probability inequalities for the sum in sampling without replacement, Ann. Stat. **2**, 39 (1974).

[35] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Ann. Math. Stat. **23**, 493 (1952).