

One-Time Universal Hashing Quantum Digital Signatures without Perfect Keys

Bing-Hong Li,¹ Yuan-Mei Xie,¹ Xiao-Yu Cao,¹ Chen-Long Li,¹ Yao Fu,^{2,*} Hua-Lei Yin,^{1,†} and Zeng-Bing Chen^{1,‡}

¹*National Laboratory of Solid State Microstructures and School of Physics,
Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China*

²*Beijing National Laboratory for Condensed Matter Physics and Institute of Physics,
Chinese Academy of Sciences, Beijing 100190, China*

(Dated: October 6, 2023)

Quantum digital signatures (QDS), generating correlated bit strings among three remote parties for signatures through quantum law, can guarantee non-repudiation, authenticity, and integrity of messages. Recently, one-time universal hashing QDS framework, exploiting the quantum asymmetric encryption and universal hash functions, has been proposed to significantly improve the signature rate and ensure unconditional security by directly signing the hash value of long messages. However, similar to quantum key distribution, this framework utilizes keys with perfect secrecy by performing privacy amplification that introduces cumbersome matrix operations, thereby consuming large computational resources, causing delays, and increasing failure probability. Here, we prove that, different from private communication, imperfect quantum keys with partial information leakage can be used for digital signatures and authentication without compromising the security while having eight orders of magnitude improvement on signature rate for signing a megabit message compared with conventional single-bit schemes. This study significantly reduces the delay for data postprocessing and is compatible with any quantum key generation protocols. In our simulation, taking two-photon twin-field key generation protocol as an example, QDS can be practically implemented over a fiber distance of 650 km between the signer and receiver. For the first time, this study offers a cryptographic application of quantum keys with imperfect secrecy and paves a way for the practical and agile implementation of digital signatures in a future quantum network.

I. INTRODUCTION

Digital signatures are cryptographic primitives that offer data authenticity and integrity [1], especially for the non-repudiation of sensitive information. It has become an indispensable and essential technique in the global internet owing to its wide application especially in digital financial transactions, email, and digital currency. However, the security of classical digital signatures, guaranteed by public-key infrastructure [2–4], is threatened by rapidly developing algorithms [5, 6] and quantum computing [7]. Different from classical digital signatures, quantum digital signatures (QDSs) can provide a higher level of security, information-theoretic security, by employing the fundamental principles of quantum mechanics. That is, QDS can protect data integrity, authenticity, and non-repudiation even if the attacker utilizes unlimited computational power. The rudiment of the single-bit QDS scheme was introduced in 2001 [8], but it could not be implemented due to some impractical requirements such as high-dimensional single-photon states and quantum memories. Subsequently, there have been many developments to remove these impractical requirements [9–11], making QDS closer to real implementation. Furthermore, based on non-orthogonal encoding [12] and orthogonal encoding [13], respectively, two independent single-bit QDS protocols without secure quantum channels were

proposed and proved to be secure for the first time. These two protocols have triggered numerous achievements of single-bit QDS theoretically [14–25] and experimentally [26–36].

Nonetheless, all these schemes still have several limitations. Protocols utilizing orthogonal encoding require additional symmetrization steps which results in extra secure channels [13]. Therefore, to guarantee information-theoretic security, quantum key distribution (QKD) and one-time pad encryption are required between two receivers in orthogonal-type protocols [28, 29]. Single-bit QDS schemes based on non-orthogonal encoding [12, 20, 23] are independent of additional QKD channels, but the signature rate is sensitive to the misalignment error of the quantum channel. In addition, all these schemes can sign only a one-bit message in each round. If one wants to sign a multi-bit message using single-bit QDS schemes, he needs to encode it into a new message string and sign the new string bit by bit [22, 37–41]. However, these solutions have not been completely proved as information-theoretically secure with the quantified failure probability, and the signature rate is very low and far from implementation for long messages with a lot of bits.

Recently, an efficient QDS scheme has been proposed based on secret sharing, one-time pad, and one-time universal hashing (OTUH) [42]. Different from single-bit QDS protocols that require a long key string to sign a one-bit message, this OUTH-QDS protocol offers a method to directly sign the hash value of multi-bit messages through one key string with information-theoretic security, and thus drastically improves the QDS effi-

* yfu@iphy.ac.cn

† hlyin@nju.edu.cn

‡ zbchen@nju.edu.cn

ciency. However, this framework requires perfect keys with complete secrecy, which is an expensive resource guaranteed by the complete procedure of QKD or quantum secret sharing (QSS). Accordingly, privacy amplification steps are required, thereby adding to the complexity of the algorithm and causing unendurable delays.

Here, we point out that quantum keys with imperfect secrecy are adequate for protecting the authenticity and integrity of messages in such a digital signature scheme. Accordingly, we propose a new OTUH-QDS protocol with imperfectly secret keys, utilizing only asymmetric quantum keys without perfect secrecy to sign multi-bit messages. We demonstrate that our proposed scheme provides information-theoretic security for digital signature tasks and simulate the performance of our protocol. The result reveals that our protocol outperforms other QDS schemes in terms of signature rate and transmission distance. In a practical case of signing a megabit message, the proposed scheme has a higher signature rate of nearly eight orders of magnitude, compared with single-bit QDS schemes due to its robustness against message size. Moreover, we show that our scheme can significantly reduce the computational costs and delays of postprocessing owing to the removal of privacy amplification. Furthermore, the proposed scheme is a general framework that can be applied to all existing QKD protocols. When utilizing the idea of two-photon twin-field QKD [43], one of the most efficient QKD protocols, to execute our work, a transmission distance of 650 km can be achieved with a signature rate of 0.01 times per second.

To date, almost all quantum communication protocols such as QKD [44–54], QSS [55–57], and quantum conference key agreement [55, 58] aim at generating quantum states among the parties and extract keys with perfect secrecy through complex postprocessing steps. Thereafter, these keys are then used to finish the corresponding cryptographic tasks such as private communication, secret sharing, and group encryption. In contrast, the proposed protocol offers a new approach to digital signature tasks that only require keys with imperfect secrecy through quantum optical communication. The troublesome postprocessing steps are thus moved out without relaxing the security assumption. This is the first instance of applying this kind of keys to cryptographic tasks with information-theoretic security. We believe that our proposed solution can provide a feasible approach to the practical application of QDS and enlighten other applications of quantum keys with imperfect secrecy in a future quantum communication network.

The remainder of this paper is organized as follows. In Sec. II we review OTUH-QDS scheme and introduce the motivation of this work. In Sec. III we propose our protocol with two approaches of universal hashing. In Sec. IV we give the security proof of authentication based on quantum keys with imperfect secrecy and then, the security analysis of the proposed QDS protocol. In Sec. V we discuss the performance of the proposed scheme and

compare it with both single-bit QDS and OTUH-QDS schemes. Finally, we conclude the paper in Sec. VI.

II. PRELIMINARIES

A. OTUH-QDS protocol

The schematic of OTUH-QDS [42] is reviewed herein. The protocol can be segmented into the distribution stage and messaging stage, consistent with single-bit QDS introduced in Appendix A 1. The length of message is denoted as m . The schematic of OTUH-QDS is shown in Fig. 1(a).

1. distribution stage

Alice, Bob, and Charlie each have two key bit strings $\{X_a, X_b, X_c\}$ with n bits and $\{Y_a, Y_b, Y_c\}$ with $2n$ bits, satisfying the perfect correlation $X_a = X_b \oplus X_c$ and $Y_a = Y_b \oplus Y_c$, respectively. The distribution stage can be realized using quantum communication protocols, such as QKD and QSS. It need to be mentioned that single-bit QDS requires only the quantum part of QKD protocols, also referred as key generation protocol (KGP). In OTUH-QDS, the users need to perform additional error correction and privacy amplification steps after KGP.

2. messaging stage

(i) *Signing of Alice*—First, Alice uses a local quantum random number, characterized by an n -bit string p_a , to randomly generate an irreducible polynomial $p(x)$ of degree n [59]. Second, she uses the initial vector (key bit string X_a) and irreducible polynomial (quantum random number p_a) to generate a random linear feedback shift register-based (LFSR-based) Toeplitz matrix [60] H_{nm} , with n rows and m columns. Third, she uses a hash operation with $Hash = H_{nm} \cdot Doc$ to acquire an n -bit hash value of the m -bit document. Fourth, she exploits the hash value and the irreducible polynomial to constitute the $2n$ -bit digest $Dig = (Hash || p_a)$. Fifth, she encrypts the digest with her key bit string Y_a to obtain the $2n$ -bit signature $Sig = Dig \oplus Y_a$ using OTP. Finally, she uses the public channel to send the signature and document $\{Sig, Doc\}$ to Bob. Note that Sig includes the information of the irreducible polynomial chosen for the hashing.

(ii) *Verification of Bob*—Bob uses the authentication classical channel to transmit the received $\{Sig, Doc\}$, as well as his key bit strings $\{X_b, Y_b\}$, to Charlie. Thereafter, Charlie uses the same authentication channel to forward his key bit strings $\{X_c, Y_c\}$ to Bob. Bob obtains two new key bit strings $\{K_{X_b} = X_b \oplus X_c, K_{Y_b} = Y_b \oplus Y_c\}$ by the XOR operation. Bob exploits K_{Y_b} to obtain an expected digest and bit string p_b via XOR decryption.

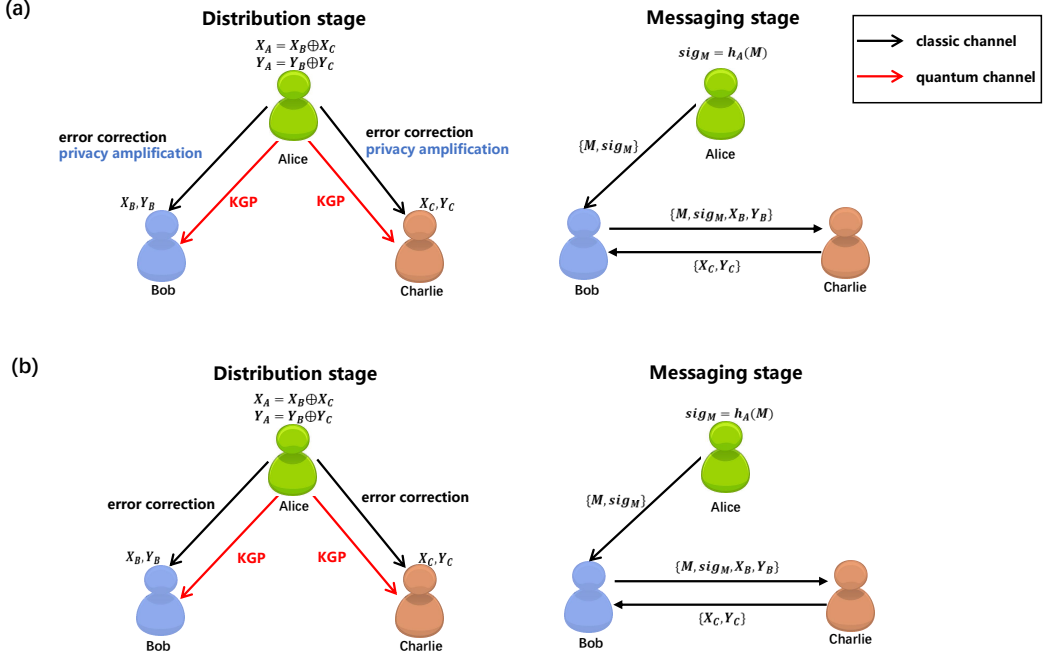


FIG. 1. (a) OTUH-QDS [42]. In the distribution stage, Alice, Bob and Charlie share key bit strings with perfect secret sharing relationship through key generation protocol (KGP), error correction and privacy amplification. In the messaging stage Alice generates the signature through AXU hashing, and sends the message and signature to Bob. Bob then sends his keys and received information to Charlie, who will later send his keys to Bob. Ultimately, Bob and Charlie use their own and received keys to infer Alice keys and then perform AXU hashing to verify the signature. (b) Schematic of the proposed protocol. In the distribution stage, the users only perform KGP and error correction to share keys with full correctness but some secrecy leakage. Their keys still hold secret sharing relationship. In the messaging stage the manipulation of classic information is analogous to that in OUTH-QDS.

Bob utilizes the initial vector K_{X_b} and irreducible polynomial p_b to establish an LFSR-based Toeplitz matrix. He uses a hash operation to acquire an n -bit hash value and then constitutes a $2n$ -bit actual digest. Bob will accept the signature if the actual digest is equal to the expected digest. Then, he informs Charlie of the result. Otherwise, Bob rejects the signature and announces to abort the protocol.

(iii) *Verification of Charlie*—If Bob announces that he accepts the signature, Charlie then uses his original key along with the key sent to Bob to create two new key bit strings $\{K_{X_c} = X_b \oplus X_c, K_{Y_c} = Y_b \oplus Y_c\}$. Charlie employs K_{Y_c} to acquire an expected digest and bit string p_c via XOR decryption. Charlie uses a hash operation to obtain an n -bit hash value and then constitutes a $2n$ -bit actual digest, where the hash function is an LFSR-based Toeplitz matrix generated by initial vector K_{X_c} and irreducible polynomial p_c . Charlie accepts the signature only if the two digests are identical; otherwise, Charlie rejects the signature.

The core point of this protocol is to realize the perfect bits correlation of the three parties, construct a completely asymmetric key relationship for them, and perform one-time almost XOR universal₂ (AXU) hashing, specifically, LFSR-based Toeplitz hashing, to generate

the signature. AXU hash functions is a special class of hash functions that can map an input value of arbitrary length into an almost random hash value with a preset length [61]. The signature generated in OTUH-QDS is simply the AXU hash value of the long message to be signed, where the AXU hash function is determined by using only one string of Alice keys. After the distribution stage, Alice's, Bob's and Charlie's keys are completely secret and correct with the relationship of secret sharing. Bob (Charlie) can only obtain Alice's keys after he receives keys of Charlie (Bob). Thus Bob can obtain no information of Alice's keys which decides the AXU hash function before transferring the message and signature to Charlie. Accordingly, Bob's forging attack under this protocol is equivalent to that against an authentication scenario where Alice sends an authenticated message to Charlie. It has been proved that such a message authentication scheme based on AXU hashing is information-theoretically secure [60]. Consequently, forging attack is protected by one-time AXU hash functions and key relationship among three parties. From the perspective of Alice, Bob and Charlie's keys are totally symmetric when they verify the signature. Thus, Alice's repudiation attack is prevented as well.

B. Motivation of this work

Different from all single-bit QDS protocols that require a long key string to sign a one-bit message, OUTH-QDS offers a method to sign multi-bit messages through one key string with information-theoretic security, and thereby drastically improves the QDS efficiency. Essentially, this advantage is introduced by AXU hash functions, which has been proved to be information-theoretically secure only under perfectly secret keys in previous studies. Thus, compared with single-bit QDS, OTUH-QDS requires extra error correction and privacy amplification steps to realize the perfect bits correlation in the distribution stage. These postprocessing steps especially privacy amplification involves multiplication calculations on matrices with comparable length of data size, which introduces heavy computational costs and unpleasant delays in practical scenarios. For large-size data, the delays will become unendurable and constrain the practicality.

The process of AXU hashing is equivalent to the scenario where the input value decides the function, mapping the initial input keys into almost random output hash values. We notice that partial secrecy leakage of input value (keys) will be concealed in AXU hash value because of its randomness. Thus, these imperfect keys with partial secrecy leakage will not undermine the authenticity of messages in a QDS scheme like OTUH-QDS. Moreover, the integrity of messages is also not compromised. Based on this concept, in this paper we propose a solution for OTUH-QDS protocols with imperfectly secret keys. In other words, we implement QDS with quantum keys without privacy amplification. As the additional computational cost and delays of OTUH-QDS are primarily introduced by privacy amplification, this concept can effectively reduce the weaknesses of OTUH-QDS and lay a ground for the future implementation of QDS in a quantum network.

The schematic of the proposed protocol is illustrated in Fig. 1(b). In the distribution stage users only perform the error correction step after KGP, ensuring that their keys have no mismatches, and build a secret sharing relationship through Alice's XOR operation. The final keys will be randomly divided into several n -bit groups for AXU hashing. Each of these groups of keys contains full correctness and some secrecy leakage with an upper bound which can be estimated through finite-size analysis using experimental data in KGP. In the messaging stage, the rules of information exchange are consistent with that in OTUH-QDS. We will prove that the bit strings generated in our distribution stage are sufficient for AXU hashing and quantify the security bound in Sec. IV. In addition, we give two solutions based on different types of AXU hash functions.

III. QDS PROTOCOL

A schematic of setups of the proposed QDS protocol is illustrated herein and illustrated in Fig. 2.

A. Distribution stage

Our proposal is a general framework in which KGP can be derived from any type of QKD protocol. As an example, the proposed scheme is demonstrated based on two-photon twin-field (TP-TF) QKD [43]. In the distribution stage, Alice—Bob and Alice—Charlie independently implement TP-TF KGP (TP-TFKGP for simplicity) to share key bit strings. We remark that in this three-party protocol the process of Alice—Bob and Alice—Charlie are independent, and can be performed separately. The difficulty of the experiment is the same as two-party QKD protocols. Specifically, TP-TFKGP utilizes the idea of two-photon interference to distribute quantum states. Consequently, the performance is independent of probability and intensity for each user, meanwhile having high misalignment error tolerance. The protocol is thus unaffected by the addition or deletion of users (as long as the number of users is on less than three), highly versatile and suitable for future quantum metropolitan networks.

1. *Preparation.* At each time bin $i \in \{1, 2, \dots, N\}$, Alice and Bob (Alice and Charlie) each independently prepare a weak coherent pulse $|e^{i(\theta_x^i + r_x^i \pi)} \sqrt{k_x^i}\rangle$ with probability p_{k_x} , where the subscript $x \in \{a, b, c\}$ represents the user Alice, Bob or Charlie, the phase $\theta_x^i \in [0, 2\pi)$, classical bit $r_x^i \in \{0, 1\}$, intensity $k_x^i \in \{\mu_x, \nu_x, \mathbf{o}_x, \hat{\mathbf{o}}_x\}$ (represent signal, decoy, preserve-vacuum and declare-vacuum intensity, $\mu_x > \nu_x > \mathbf{o}_x = \hat{\mathbf{o}}_x = 0$) are chosen randomly. Then Alice and Bob (Alice and Charlie) transmit the corresponding pulses to the untrusted relay Eve through insecure quantum channels, respectively. In addition, they send a bright reference light to Eve to measure the phase noise difference ϕ_{ab}^i (ϕ_{ac}^i).

2. *Measurement.* Eve performs interference measurements on every received pulse pair with a beam splitter and two detectors. If one and only one detector clicks, Eve announces that she obtained a successful detection event and which detector clicked. In the following we use the brace with the information of users' intensity selection in it to distinguish these events. For example, $\{\mu_a, \mathbf{o}_b\}$ represents the events that Alice selects signal intensity and Bob selects vacuum intensity.

3. *Sifting.* Here we only list the sifting process between Alice and Bob for simplicity since Alice—Bob and Alice—Charlie are symmetric. Alice and Charlie will sift their successful detection events following the same approach.

All successful events are segmented into two parts. The first part is those when neither Alice nor Bob selects the decoy or declare-vacuum intensity, i.e., $\{\mu_a, \mathbf{o}_b\}$, $\{\mu_a, \mu_b\}$, $\{\mathbf{o}_a, \mu_b\}$, and $\{\mathbf{o}_a, \mathbf{o}_b\}$, which will be used for generating data in the Z basis to form the key. The other successful events, i.e., the second part, are used for estimating

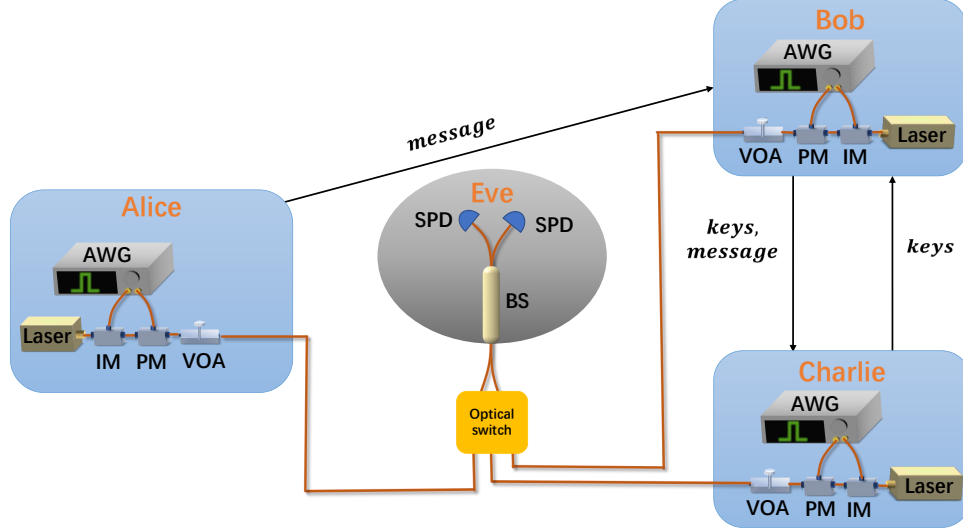


FIG. 2. Schematic of the setup of the proposed QDS protocol. The red line represents quantum optical channel in the distribution stage, and the black arrow line represents the information exchange through classic authenticated channel in the messaging stage. (i) In the distribution stage, Alice, Bob and Charlie utilize a narrow-linewidth continuous-wave laser, intensity modulator (IM), phase modulator (PM), arbitrary wave generator (AWG) and variable optical attenuator (VOA) to prepare a phase-randomized weak coherent source with different intensities and phases. The signals from Bob and Charlie will both go through an optical switch. An untrusted relay Eve performs interference measurement on the signals from Alice and an optical switch with a beam splitter (BS) and a single-photon detector (SPD). After sifting, parameter estimation and error correction, Alice can share bit strings with Bob and Charlie, respectively. (ii) In the messaging stage, Alice transmits the desired message to Bob. Bob sends the message along with his keys to Charlie. Charlie will then send his keys to Bob. Then Bob verifies the signature by his own and received keys. If he accepts the signature, he will inform Charlie who will also verify the signature by his own and received keys. The signature is successfully validated if both Bob and Charlie accept it.

parameters. For the first part of events, Alice randomly matches a time bin i of intensity μ_a with another time bin j of intensity \mathbf{o}_a . Thereafter she sets her bit value as 0 (1) if $i < j$ ($i > j$), and informs the serial numbers i and j to Bob. In the corresponding time bins, if Bob chooses intensities $k_b^{\min\{i,j\}} = \mu_b$ (\mathbf{o}_b), and $k_b^{\max\{i,j\}} = \mathbf{o}_b$ (μ_b), he sets his bit value as 0 (1). Bob announces to abort the event where $k_b^i = k_b^j = \mathbf{o}_b$ or μ_b . To conclude, the preserved events in the Z basis are sifted as $\{\mu_a \mathbf{o}_a, \mathbf{o}_b \mu_b\}$, $\{\mu_a \mathbf{o}_a, \mu_b \mathbf{o}_b\}$, $\{\mathbf{o}_a \mu_a, \mathbf{o}_b \mu_b\}$, and $\{\mathbf{o}_a \mu_a, \mu_b \mathbf{o}_b\}$.

For the second part of events, Alice and Bob communicate their intensities and phase information with each other via an authenticated channel. Define the global phase difference at time bin i as $\theta^i := \theta_a^i - \theta_b^i + \phi_{ab}^i$. Alice and Bob keep detection events $\{\nu_a^i, \nu_b^i\}$ only if $\theta^i \in [-\delta, \delta] \cup [\pi - \delta, \pi + \delta]$. They randomly select two retained detection events that satisfy $|\theta^i - \theta^j| = 0$ or π , and then match these two events, denoted as $\{\nu_a^i \nu_a^j, \nu_b^i \nu_b^j\}$. By calculating classical bits $r_a^i \oplus r_a^j$ and $r_b^i \oplus r_b^j$, Alice and Bob extract a bit value in the X basis, respectively. Subsequently, Bob always flips his bit in the Z basis. In the X basis, Bob flips part of his bits to correctly correlate them with those of Alice. To be specific, when the global phase difference between two matching time bins is 0 (π) and the two clicking detectors announced by Eve are different (same), Bob will flip

his bits. Otherwise, Bob will directly save his bits for later use. The other events in the second part is used for decoy analysis.

4. *Parameter estimation.* Alice and Bob (Alice and Charlie) form the n_Z -length raw key bit from the random bits under the Z basis. The remaining bits in the Z basis are used to estimate the bit error rate E^Z . Further, they communicate all bit values in the X basis to obtain the total number of errors. The decoy-state method [62, 63] is used to estimate the number of vacuum events in the Z basis $s_{\mu_b}^Z$, the count of single-photon pairs s_{11}^Z , and the phase error rate of single-photon pairs ϕ_{11}^Z on the Z basis.

5. *Error correction and examination.* Alice and Bob (Alice and Charlie) distill final keys by utilizing an error correction algorithm with ε_{cor} -correctness. [64, 65] The size of the distilled key remains n_Z , and the unknown information of a possible attacker can be expressed as \mathcal{H} . Alice then randomly disturbs the orders of the distilled key and publicizes the new order to Bob (Charlie) through the authenticated channel. Subsequently, Alice and Bob (Alice and Charlie) divide the final keys into several n -bit strings, each of which is used to perform a task in the messaging stage. The grouping process can be considered as a random sampling. More details are shown in Sec. IV A and Appendix C1.

B. Messaging stage

Various AXU hash functions can be employed in the messaging stage of the proposed protocol by following the framework presented in Fig. 1(b). To demonstrate the detailed procedure, we here present two specific approaches to the messaging stage utilizing LFSR based Toeplitz hashing and generalized division hashing, respectively. LFSR-based Toeplitz hashing is highly compatible with the hardware systems whereas generalized division hashing is more suitable for realizing software systems. In a practical case we select either of methods of hashing depending on the different application environments of users. The message to be signed is denoted as M . For each M if using LFSR-based Toeplitz hashing Alice generates six bit strings $\mathbb{X}_B, \mathbb{X}_C, \mathbb{Y}_B, \mathbb{Y}_C, \mathbb{Z}_B, \mathbb{Z}_C$, each of length n . If choosing generalized division hashing in the messaging stage, Alice will only generate four bit strings $\mathbb{X}_B, \mathbb{X}_C, \mathbb{Y}_B, \mathbb{Y}_C$. The subscripts represent the participants performing KGP with Alice, where B represents Bob and C represents Charlie. Thereafter, Alice will generate $\mathbb{X}_a = \mathbb{X}_b \oplus \mathbb{X}_c, \mathbb{Y}_a = \mathbb{Y}_b \oplus \mathbb{Y}_c$, and $\mathbb{Z}_a = \mathbb{Z}_b \oplus \mathbb{Z}_c$ as her own key strings. For the scheme with LFSR-based Toeplitz hashing the signature rate is

$$R_{\text{LFSR}} = n_Z/3n, \quad (1)$$

whereas for generalized division hashing there is

$$R_{\text{GDH}} = n_Z/2n. \quad (2)$$

1. Utilizing LFSR-based Toeplitz hashing

Definition 1. LFSR-based Toeplitz hash functions: LFSR-based Toeplitz hash functions can be expressed as $h_{p,s}(M) = H_{nm}M$, where p, s determines the function and $M = (M_0, M_1, \dots, M_{m-1})^T$ is the message in the form of an m -bit vector. The process of generating LFSR-based Toeplitz hash function is detailed as follows.

A randomly selected irreducible polynomial of order n in the field $\text{GF}(2)$, $p(x)$, determines the construction of LFSR. $p(x) = x^n + p_{n-1}x^{n-1} + \dots + p_1x + p_0$ can be characterized by its coefficients of order from 0 to $n-1$, i.e., $p = (p_{n-1}, p_{n-2}, \dots, p_1, p_0)$. For the initial state s which is also represented as an n -bit vector $s = (a_n, a_{n-1}, \dots, a_2, a_1)^T$, the LFSR will be performed n times to generate n vectors. Specifically, it will shift down every element in the previous column, and add a new element to the top of the column. For instance, the LFSR transforms s into $s_1 = (a_{n+1}, a_n, \dots, a_3, a_2)^T$, where $a_{n+1} = p \cdot s$, and likewise, transforms s_1 to s_2 . Then the m vectors s, s_1, \dots, s_{m-1} will together construct the Toeplitz matrix $H_{nm} = (s, s_1, \dots, s_{m-1})$, and the hash value of the message is $H_{nm}M$.

(i) Alice obtains a string of random numbers through a quantum random number generator and uses it to randomly generate a monic irreducible polynomial in $\text{GF}(2)$

of order n , denoted as $p(x)$. $p(x)$ can be characterized by its coefficients of order from 0 to $n-1$, i.e., an n -bit string, denoted by p_a . Details of generating $p(x)$ can be found in Appendix B 1.

(ii) Alice uses her key bit string \mathbb{Y}_a and $p(x)$ to perform LFSR-based Toeplitz hashing and generates an n -bit hash value $Dig = H_{\mathbb{Y}_a, p_a}(M)$, and encrypts it by \mathbb{Z}_a to obtain the final signature $Sig = Dig \oplus \mathbb{Z}_a$. In addition, Alice encrypts p_a by the key set \mathbb{X}_a to obtain the encrypted string $p = p_a \oplus \mathbb{X}_a$. Here we adopt a different expression from that in OUTH-QDS that we independently list the hash value as Dig and the coefficients of the irreducible polynomial as p_a , i.e., Sig does not include the information of the irreducible polynomial to avoid misunderstanding. She then transmits $\{Sig, p, M\}$ to Bob through an authenticated classical channel.

(iii) Bob transmits $\{Sig, p, M\}$ as well as his key bit strings $\{\mathbb{X}_b, \mathbb{Y}_b, \mathbb{Z}_b\}$ to Charlie so as to inform Charlie that he has received the signature. Thereafter, Charlie forwards his key bit strings $\{\mathbb{X}_c, \mathbb{Y}_c, \mathbb{Z}_c\}$ to Bob. These data are all transmitted through an authenticated channel. Bob obtains three new key bit strings $K_{\mathbb{X}_b} = \mathbb{X}_b \oplus \mathbb{X}_c, K_{\mathbb{Y}_b} = \mathbb{Y}_b \oplus \mathbb{Y}_c$, and $K_{\mathbb{Z}_b} = \mathbb{Z}_b \oplus \mathbb{Z}_c$ using the XOR operation. He exploits $K_{\mathbb{X}_b}$ and $K_{\mathbb{Z}_b}$ to obtain the expected digest and string p_b via XOR decryption. He utilizes $K_{\mathbb{Y}_b}$ and p_b to establish an LFSR-based Toeplitz matrix and derive an actual digest via a hash operation. Bob will accept the signature if the actual digest is equal to the expected digest. Then he informs Charlie of the result.

(iv) If Bob announces that he accepts the signature, Charlie creates three new key bit strings $K_{\mathbb{X}_c} = \mathbb{X}_b \oplus \mathbb{X}_c, K_{\mathbb{Y}_c} = \mathbb{Y}_b \oplus \mathbb{Y}_c$ and $K_{\mathbb{Z}_c} = \mathbb{Z}_b \oplus \mathbb{Z}_c$ using his original key and that received from Bob. He employs $K_{\mathbb{X}_c}$ and $K_{\mathbb{Z}_c}$ to acquire the expected digest and variable p_c via XOR decryption. Charlie obtains an actual digest via hash operation, where the hash function is an LFSR-based Toeplitz matrix generated by $K_{\mathbb{Y}_c}$ and p_c . Charlie accepts the signature if the two digests are identical.

2. Utilizing generalized division hashing

Definition 2. Generalized division hash functions: The generalized division hash functions can be expressed as $h_P(M) = M(x) \cdot x^{n/k} \bmod P(x)$, where $P(x)$ is a monic irreducible polynomial of order n/k in the field $\text{GF}(2^k)$, M is the message and $M(x)$ is the polynomial of order m/k in $\text{GF}(2^k)$ with every coefficient corresponding to k bits of M . The calculation is also performed in $\text{GF}(2^k)$. The final result is a polynomial of order n/k in field $\text{GF}(2^k)$, and can be transformed into an n -bit strings. [66]

Commonly, k is set as $k = 2^x$ for simplicity, where x is a positive integer. In the current scheme, we select $k = 2^3 = 8$.

(i) In this case, Alice selects $\mathbb{X}_a = \mathbb{X}_b \oplus \mathbb{X}_c$ and $\mathbb{Y}_a = \mathbb{Y}_b \oplus \mathbb{Y}_c$ as her own key sets. Alice first obtains

a string of random numbers through a quantum random number generator and uses it to randomly generate a monic irreducible polynomial in $\text{GF}(2^8)$ of order $n/8$, denoted by $P(x)$. The generation process of $p(x)$ are detailed in Appendix B1. $P(x)$ can be characterized by its coefficients of order from 0 to $n/8 - 1$. By encoding each coefficient into an 8-bit string, we can use an n -bit string to express $P(x)$, denoted as P_a . Subsequently, Alice encrypts P_a by the key set \mathbb{X}_a to obtain the encrypted string $P = P_a \oplus \mathbb{X}_a$.

(ii) Alice uses $P(x)$ to perform the generalized division hashing [66] to obtain an n -bit hash value $Dig = h_{P_a}(M)$. She encrypts Dig by \mathbb{Y}_a to derive the signature $Sig = Dig \oplus \mathbb{Y}_a$ and transmits the message along with the obtained signature $\{Sig, p, M\}$ to Bob.

step (iii) and (iv) are similar to those utilizing LFSR-based Toeplitz hashing. Bob and Charlie will exchange their key strings in turn through an authenticated channel and examine their expected and received digests.

This summarizes the entire procedure of the proposed protocol. Note that the TP-TFKGP can be replaced by any other KGP such as BB84-KGP or sending-or-not-sending (SNS)-KGP. Actually, in the distribution stage Alice shares bit strings with Bob and Charlie in the relationship of secret sharing. Thus, the distribution stage can also be performed based on QSS without employing the privacy amplification step.

IV. SECURITY ANALYSIS

Similar to OTUH-QDS, the core point of the proposed protocol is the security of the authentication based on AXU hashing, which directly protects the security of QDS against fogery [42]. However, the security of our protocol differs because of the information leakage during the distribution stage. In this section we first analyze the success probability of an attacker guessing a key string generated in the distribution stage, and thereafter provide a more detailed security analysis of AXU hashing under imperfect keys with partial secrecy leakage, and finally demonstrate the security of our protocol.

A. Guessing probability of the attacker

Unlike QKD that generates keys with perfect secrecy, in our protocol the keys are imperfectly secret. Any possible attackers may obtain partial information on the keys. After the distribution stage, users share keys in the form of several n -bit strings. We need to quantify the information leakage and bound the maximum probability of the attacker guessing such a string of keys. Suppose an n -bit key string as \mathbb{X} and the attacker's system is \mathbb{B} . We consider a general attack scenario where attackers can execute any entangling operations on the system of any or all states, obtain a system ρ_B^x and perform any positive

operator-valued measure $\{E_B^x\}_x$ on it. The probability that the attacker correctly guesses \mathbb{X} using an optimal strategy is denoted as $P_{\text{guess}}(\mathbb{X}|\mathbb{B})$. According to the definition of min-entropy in Ref. [67],

$$P_{\text{guess}}(\mathbb{X}|\mathbb{B}) = \max_{\{E_B^x\}_x} \sum_x P_x \text{tr}(E_B^x \rho_B^x) = 2^{-H_{\min}(\mathbb{X}|\mathbb{B})_\rho}, \quad (3)$$

where $H_{\min}(\mathbb{X}|\mathbb{B})_\rho$ is the min-entropy of \mathbb{X} and \mathbb{B} . If \mathbb{X} is generated in the distribution stage of our protocol, $H_{\min}(\mathbb{X}|\mathbb{B})_\rho$ can be estimated by

$$H_{\min}(\mathbb{X}|\mathbb{B})_\rho = \mathcal{H}_n. \quad (4)$$

Thus, we have the relationship

$$P_{\text{guess}}(\mathbb{X}|\mathbb{B}) = 2^{-\mathcal{H}_n}, \quad (5)$$

which means that the attacker can correctly guess \mathbb{X} with a probability no more than $2^{-\mathcal{H}_n}$. \mathcal{H}_n is the total unknown information of the n -bit string and can be upper bounded by parameters estimated in the distribution stage

$$\mathcal{H}_n \leq \underline{s}_{0\mu_b}^{zn} + \underline{s}_{11}^{zn} \left[1 - H(\phi_{11}^{zn}) \right] - nfH(E^z), \quad (6)$$

where f is the error correction efficiency, $\underline{s}_{0\mu_b}^{zn}$ and \underline{s}_{11}^{zn} are the lower bounds of vacuum events and single-photon pairs events in the n -bit string, respectively, and ϕ_{11}^{zn} is the upper bound of the phase error rate of single-photon pairs in the n -bit string. More details of calculation are shown in Appendix C1.

B. Security of authentication based on hashing

In our QDS schemes, hashing is used to perform the authentication task. Thus we first consider the authentication scenario where the sender generates a signature $Sig = h(M) \oplus r$ as message authentication code, and sends $\{M, Sig\}$ to the recipient. The attacker can intercept and capture $\{M, Sig\}$, tamper a new message and signature $\{M', Sig'\}$, and send it to the recipient, who will examine whether $Sig' = h(M') \oplus r$ before accepting it. The attacker succeed if and only if (iff) a combination $\{m, t\}$ is select with the relationship $h(m) = t$, and $\{M' = M \oplus m, Sig' = Sig \oplus t\}$ is sent to the recipient. In this case, the recipient will accept the message because of the relationship $h(M \oplus m) = h(M) \oplus h(m) = Sig \oplus t$. It should be mentioned that $m \neq 0$ due to the requirement for a valid forge.

Suppose keys generated in the distribution stage of our protocol, i.e., keys with partial information leakage, are used to perform this authentication task, and define ϵ as the success probability of the attacker under this scenario. We should consider three types of possible attacks. The first one is to randomly generate m, t . It is a trivial strategy whose success probability is only

$$\epsilon_1 = 2^{-n}. \quad (7)$$

The other two types of attacks are guessing keys that decide the hash function and recovering the function from signatures.

1. Attack of guessing keys

The LFSR-based Toeplitz hash function is represented as $h_{p,s}(M) = H_{nm} \cdot M$, where H_{nm} is determined by the two bit strings p and s [60]. Herein we follow the terminology in the messaging stage of the proposed protocol where p is actually p_a encrypted by \mathbb{X}_a , s is \mathbb{Y}_a , and the hash value Dig is encrypted by \mathbb{Z}_a . We show that guessing only \mathbb{X}_a or in other words, guessing only p_a is enough to execute an optimal attack by a proposition.

Proposition 1. *For the LFSR-based Toeplitz hash function $h_{p,s}(M) = H_{nm} \cdot M$, if $p(x)|M(x) = M_{m-1}x^{m-1} + \dots + M_1x + M_0$, then $h_{p,s}(M) = 0$.*

The proof of this proposition is shown in Appendix B2. It means that the attacker can easily generate a message m satisfying the relationship $h(m) = 0$ if he knows p . In the scenario described above, suppose the attacker obtains a string \mathbb{X}_g as his estimation of \mathbb{X}_a . He can decrypt it to obtain p_g as his guessing of p_a and transform p_g into a polynomial $p_g(x)$. Thereafter the attacker can easily generate a bit string m satisfying $p_g(x)|m(x)$, and there is the relationship $h(m) = 0$ if $p_g = p_a$ (or equivalently $\mathbb{X}_g = \mathbb{X}_a$) according to Proposition 1. Then he can tamper the message into $M \oplus m$ without changing the signature. $\{M + m, Sig\}$ will pass the authentication test if $\mathbb{X}_g = \mathbb{X}_a$. As $m(x)$ is m -order and the polynomial is n -order, the attacker can select no more than m/n polynomials and multiply them to consist his choice of $m(x)$. In other words, he can guess the string \mathbb{X}_a for no more than m/n times. It must be considered that the attacker knows p_a is irreducible, so he will only choose those guesses that satisfy p_a is irreducible. The success probability of this optimized strategy can be expressed as

$$P_1 = \frac{m}{n} \cdot P(\mathbb{X}_a = \mathbb{X}_g | p_g \in \mathcal{I}), \quad (8)$$

where $P(A|B)$ represents the probability of event A under the condition that event B occurs, and \mathcal{I} denotes the set of all irreducible polynomials of order n in $\text{GF}(2)$. The cardinal number of \mathcal{I} , i.e., the number of all n -order irreducible polynomials in $\text{GF}(2)$, is more than $2^{n-1}/n$. Thus $P(p_g \in \mathcal{I}) \leq (2^{n-1}/n)/2^n = 1/2n$. It is obvious that $P(\mathbb{X}_a = \mathbb{X}_g, p_g \in \mathcal{I}) = P(\mathbb{X}_a = \mathbb{X}_g)$ because if $\mathbb{X}_a = \mathbb{X}_g$ then $p_g = p_a \in \mathcal{I}$. Then we can obtain the upper bound of the success probability of this type of

attack, denoted as ϵ_{LFSR} ,

$$\begin{aligned} P_1 &= \frac{m}{n} \cdot \frac{P(\mathbb{X}_a = \mathbb{X}_g)}{P(p_g \in \mathcal{I})} \\ &\leq \frac{m}{n} \cdot \frac{2^{-\mathcal{H}_n}}{\frac{1}{2n}} \\ &= m \cdot 2^{1-\mathcal{H}_n} = \epsilon_{\text{LFSR}}. \end{aligned}$$

The attacker can also guess the strings \mathbb{X}_a and \mathbb{Y}_a to obtain p and s so that he can guess the hash function and make a successful attack for certainty. Under this circumstance his success probability is no more than ϵ_{LFSR} ,

$$\begin{aligned} P_2 &= P(\mathbb{X}_a = \mathbb{X}_g, \mathbb{Y}_a = \mathbb{Y}_g | p_g \in \mathcal{I}) \\ &\leq P(\mathbb{X}_a = \mathbb{X}_g | p_g \in \mathcal{I}) \\ &\leq \epsilon_{\text{LFSR}}. \end{aligned}$$

The generalized division hash function $h_P(M) = m(x) \cdot x^{n/8} \bmod P(x)$ is determined only by P . As earlier, we also follow the terminology in the proposed protocol that P is P_a encrypted by \mathbb{X}_a and the hash value Dig is encrypted by \mathbb{Y}_a . The attacker's strategy is to guess a string \mathbb{X}_g such that he can obtain P_g and then forge a message. Analogous to the analysis discussed above, the upper bound of the success probability is defined as

$$\epsilon_{\text{GDH}} = \frac{m}{n} \cdot \frac{2^{-\mathcal{H}_n}}{\frac{4}{n}} = m \cdot 2^{-2-\mathcal{H}_n}. \quad (9)$$

The only difference in the calculation is that there are at least $2^{n-1}/(n/8)$ irreducible polynomials of order $n/8$ in $\text{GF}(2^8)$, so $P(P_g \in \mathcal{I}) \geq (2^{n-1}/(n/8))/2^n = 4/n$.

2. Attack of recovering keys from signature

The attacker can attempt to recover the desired keys from the captured signature. In both kinds of hashing the hash value is encrypted to generate the signature. Thus the attacker must first guess the corresponding key strings (\mathbb{Z}_a in LFSR-based Toeplitz hashing or \mathbb{Y}_a in generalized division hashing) and then perform the recovering algorithm. The success probability of this strategy is no more than that only guessing the bit string ($P(\mathbb{Z}_a = \mathbb{Z}_g)$ or $P(\mathbb{Y}_a = \mathbb{Y}_g)$) and is obviously no more than ϵ_{LFSR} or ϵ_{GDH} .

In conclusion, the optimal strategy on LFSR-based Toeplitz hashing and generalized division hashing (GDH) is to guess the key string that encrypts the polynomial. We can quantify the upper bound of failure probability of authentication based on both types of hashing with imperfect keys of secrecy leakage:

$$\epsilon_{\text{LFSR}} = m \cdot 2^{1-\mathcal{H}_n}, \quad (10)$$

$$\epsilon_{\text{GDH}} = m \cdot 2^{-2-\mathcal{H}_n}. \quad (11)$$

C. Security of the QDS scheme

Finally, we analyze the security in the QDS scheme which contains three parts, robustness, repudiation, and forgery.

1. Robustness.

The honest run abortion means the protocol is aborted when all parties are honest. It occurs only when Alice and Bob (or Charlie) share different key bits after the distribution stage. In the proposed protocol Alice and Bob (Charlie) perform error correction in the distribution stage. Thus, they share the identical final key, and the honest run occurs only at the case where errors occur. The robustness bound is $\epsilon_{\text{rob}} = 2\epsilon_{\text{cor}} + 2\epsilon'$, where ϵ_{cor} is the failure probability of the error correction protocol in the distribution stage, and ϵ' is the probability that error occurs in classical message transmission. Remark that we assume $\epsilon' = 10^{-11}$ for simplicity since it is a parameter of classical communication.

2. Repudiation.

Alice successfully repudiates when Bob accepts the message while Charlie rejects it. For Alice's repudiation attacks, Bob and Charlie are both honest and symmetric and possess the same new key strings. They will converge on the same decision for the same message and signature. In other words, when Bob rejects (accepts) the message, Charlie also rejects (accepts) it. Repudiation attacks succeed only when errors occur in one of the key exchange steps. Thus, the repudiation bound is $\epsilon_{\text{rep}} = 2\epsilon'$.

3. Forgery.

Bob forges successfully when Charlie accepts the tampered message forwarded by Bob. According to the proposed protocol, Charlie accepts the message iff Charlie obtains the same result through one-time pad decryption and one-time AXU hash functions. In principle, this is the same as an authentication scenario in Sec. IV B where Bob is the attacker attempting to forge the information sent from Alice to Charlie. Therefore, the probability of a successful forgery ϵ_{for} can be determined by the failure probability of hashing, i.e., one chooses two distinct messages with identical hash values. For the scheme utilizing LFSR-based Toeplitz hash $\epsilon_{\text{for}} = m \cdot 2^{1-\mathcal{H}_n}$, and for generalized division hashing $\epsilon_{\text{for}} = m \cdot 2^{-2-\mathcal{H}_n}$.

The total security bound of QDS, i.e., the maximum failure probability of the protocol, is $\epsilon = \max\{\epsilon_{\text{rob}}, \epsilon_{\text{rep}}, \epsilon_{\text{for}}\}$.

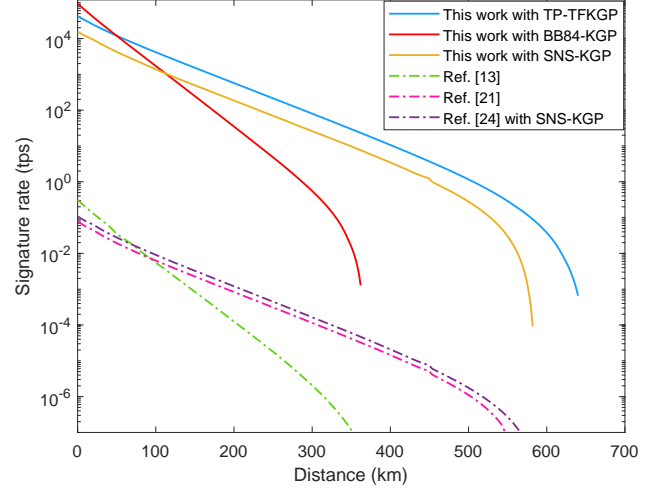


FIG. 3. Signature rates of the proposed protocol with TP-TFKGP, BB84-KGP, SNS-KGP, and decoy state BB84-QDS [13], SNS-QDS [21], SNS-QDS with random pairing [24] with the message size of 1 Kb. In the proposed protocol we use generalized division hashing in messaging stage. The repetition rate of the laser is 1 GHz. The distances between Alice-Bob and Alice-Charlie are assumed to be the same. The data size N is 10^{13} and the security bound is 10^{-10} .

V. DISCUSSION

From Eqs. (1), (2), (10), and (11), there are just differences in a constant $2/3$ between two signature rates and a constant 8 between two security parameters. The difference between the two approaches is trivial. For simplicity, we only discuss the protocol with generalized division hashing in this section.

To demonstrate the advantage of the current proposal, we first build our protocol based on BB84-KGP, SNS-KGP and TP-TFKGP, and compare them with decoy-state BB84-QDS [13] and SNS-QDS [21] which are single-bit QDS protocols based on BB84-KGP and SNS-KGP. We also compare SNS-QDS with random pairing [24], which improves the signature rate of SNS-QDS and can be applied to other QDS. More details of the calculation are shown in Appendix C. In the simulations, we consider two common cases where each message to be signed is 10^3 bits (1 Kb) and 10^6 bits (1 Mb), respectively. The repetition rate of the laser is 1 GHz, and the distances between Alice-Bob and Alice-Charlie are assumed to be

TABLE I. Simulation parameters. η_d and p_d denote the detector efficiency and dark count rate, respectively. e_d represents the misalignment error rate. N is the data size. α is the attenuation coefficient of the fiber. f is the error correction efficiency. ϵ is the failure probability of QDS schemes.

η_d	p_d	e_d	N	α	f	ϵ
70%	10^{-8}	0.02	10^{13}	0.165	1.1	10^{-10}

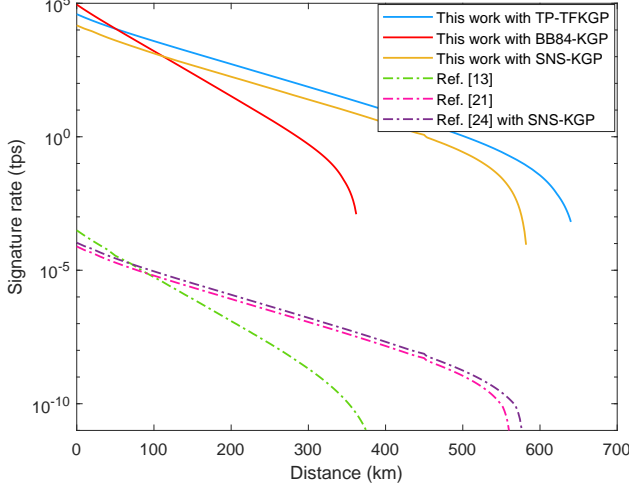


FIG. 4. Signature rates of the proposed protocol with TP-TFKGP, BB84-KGP, SNS-KGP, and decoy state BB84-QDS [13], SNS-QDS [21], SNS-QDS with random pairing [24] with the message size of 1 Mb. In the protocol, we use generalized division hashing in messaging stage. The repetition rate of the laser is 1 GHz. The distances between Alice–Bob and Alice–Charlie are assumed to be the same. The data size N is 10^{13} and the security bound is 10^{-10} .

the same. The unit of signature rate is set as time per second (tps). Detailed analysis is shown in Appendix A 2. Other simulation parameters are listed in Table I.

It should be mentioned that all conventional single-bit QDS protocols sign only a one-bit message every round. In the case of signing the multi-bit message, an m -bit message must be encoded into a new sequence with length h by inserting ‘0’ and adding ‘1’ to the original sequence. The signing efficiency, i.e., $\hat{\eta} = m/h$, is obviously less than 1. For simplicity, we use the upper bound $\hat{\eta} = 1$, i.e., $h = m$, in our simulation. It is obvious that key consumption of single-bit QDS increases linearly with message size m . In other words, the signature rate is proportional to $1/m$. In our proposed scheme, the signature is generated by hash functions operating on the message, so that the signature rate is robust against the length of the message. From Eqs. (10) and (11), ϵ increases linearly as m increase, but decrease exponentially as \mathcal{H}_n increases. Thus, to guarantee the same epsilon, \mathcal{H}_n , which is proportional to group size n , increases logarithmically with m . Consequently, the signature rate of the proposed scheme is proportional to $-\log_2 m$.

The simulation results of all the protocols mentioned are presented in Figs. 3 and 4. For the message size of 1 Kb, our protocols show an advantage on signature rate of over five orders of magnitude compared with conventional QDS schemes, which is a quite larger improvement than SNS-QDS with random pairing. If the message size becomes 1 Mb, the signature rate of conventional BB84-QDS, SNS-QDS, and SNS-QDS with random pairing will decrease by three orders of magnitude, whereas that of

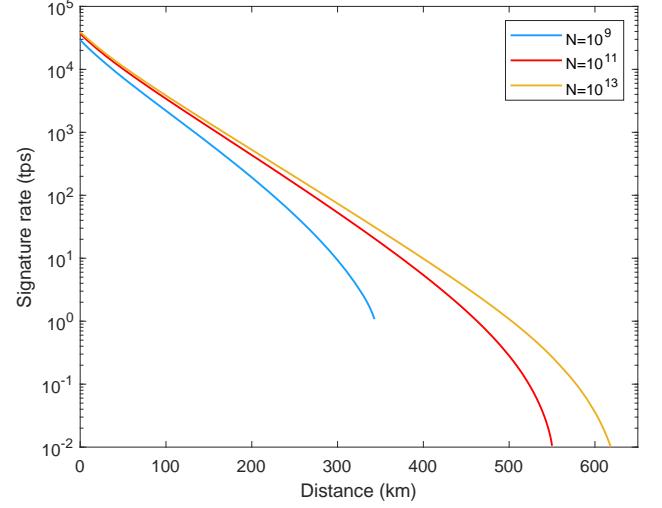


FIG. 5. Signature rates of the proposed protocols with TP-TFKGP under different data sizes $N = 10^9$, 10^{11} and 10^{13} . The message size is assumed to be 1 Mb, and the repetition rate of the laser is 1 GHz. The security bound is 10^{-10} .

our protocols decreases only slightly. Thus the proposed QDS scheme delivers a signature rate with eight orders of magnitude higher than previous schemes. As demonstrated, the proposed protocol shows great robustness to message size. Furthermore, based on TP-TFKGP the proposed scheme can reach a transmission distance of 650 km as well as a signature rate of approximately 0.01 times per second (tps). It is an immense breakthrough in terms of both distance and signature rate, indicating the considerable potential of the proposed protocol in the practical implementation of QDS. The performance of the proposed protocol under different data sizes 10^9 , 10^{11} and 10^{13} is depicted in Fig. 5. The curve of $N = 10^9$ stops at 1 tps, i.e., one time for all data, because signing less than 1 time (1 message) for all data is nonsense. The result shows that even with a data size as small as 10^9 , the proposed protocol can reach a transmission distance of 350 km, and performance of data size $N = 10^{11}$ is close to that of $N = 10^{13}$. The influence of finite-size effects caused by small data size on our protocol is in an acceptable level.

Compared with OTUH-QDS, the proposed protocol does not require perfectly secret keys, and thus involves no privacy amplification step. Therefore, the proposed protocol only consumes keys with partial information leakage, which is an affordable and practical resource compared with perfect quantum keys generated by quantum secure communication. Error correction of quantum keys can be easily performed by classical Cascade protocol [64, 65] where the bit string is first blocked and then manipulated by blocks. Thus the complexity of error correction increases linearly with the data size N and can be performed via stream computing. Privacy amplification, however, requires a hash matrix multiplication step

TABLE II. Time consumption of error correction T_{EC} and privacy amplification T_{PA} under different data sizes $N = 10^{13}$ and $N = 10^{11}$ when the distance is 400 km. $T_1 = T_{EC}$ and $T_2 = T_{EC} + T_{PA}$ represent the postprocessing time of the proposed scheme and OTUH-QDS, respectively. n_Z is the number of raw bits generated in TP-TFKGP; l is the length of keys after privacy amplification. In case $N = 10^{13}$, postprocessing time of OTUH-QDS is 5.85 h, and that of the proposed protocol is only 8.07 min.

N	n_Z	errors	l	T_{EC}	T_{PA}	T_1	T_2
10^{13}	1.695×10^8	300	4.87×10^7	8.07 min	5.71 h	8.07 min	5.85 h
10^{11}	1.267×10^6	39830	2.51×10^5	3.62 s	2.98 s	3.62 s	6.6 s

where the numbers of columns and rows are proportional to N . Thus the computational complexity of privacy amplification is $O(N^2)$. The fast Fourier transform algorithm can reduce the complexity to $O(N \log N)$ [68], and one can also block the keys before performing privacy amplification. However, as the minimum blocks should be adequately large to minimize the finite-size effect, the actual computational cost and delay of privacy amplification are still very large.

The time consumed in conducting consumption of error correction, privacy amplification, and data transmission are listed in Table II, including the total postprocessing time of both protocols, at a distance of 400 km with data sizes 10^{13} and 10^{11} . Details of simulation are introduced in Appendix D. If $N = 10^{11}$, time consumption of postprocessing in OTUH-QDS is 6.6 s, while that of the proposed protocol is 3.62 s. Moreover, when $N = 10^{13}$, time for privacy amplification is 5.71 h, which will introduce a quite long delay in experiment. Accordingly, time for error correction is only 8.07 min. The proposed scheme, free of privacy amplification, can significantly save computational resources and minimize postprocessing delays.

We further compare the signature rates of the proposed protocol and that of OTUH-QDS [42]. Theoretically, the two signature rates should be equal under ideal conditions. In practical cases, there are two effects that influence the performance of the proposed protocol compared with OTUH-QDS. The first effect is that in our protocol the parameter n is optimized, which will improve the signature rate compared with OUTH-QDS. This effect will decrease as distance increases. The second effect is that in our protocol we consider the statistical fluctuation of the error rate in the grouping process. This effect will damage the signature rate compared with OUTH-QDS. At both long and short distances, this effect is slight because the size of groups is small and the error rate is small, respectively.

In Fig. 6, we draw the ratio of the signature rate of the proposed protocol based on TP-TFKGP and that of OTUH-QDS [42] combined with TP-TFQKD, if not considering the postprocessing time, with data sizes of 10^{13} and 10^{11} , and message size of 1Kb. The result shows that the ratio is more than 80% for transmission distances less than 500 km. Overall, the signature rates of the two protocols are comparable. In addition, in case of assuming the repetition rate of the laser as 1 GHz, time consump-

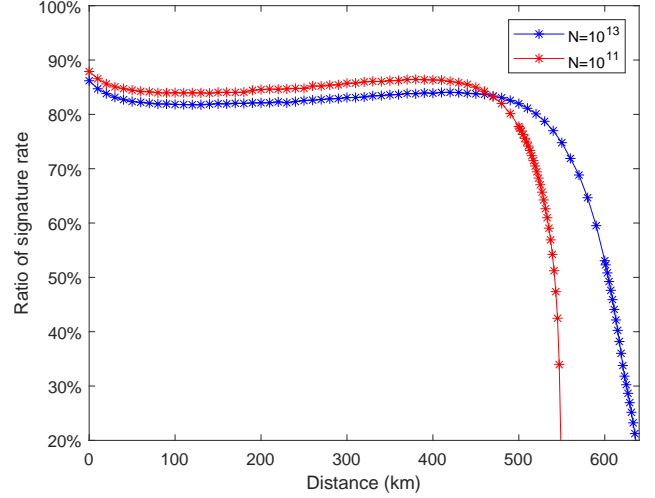


FIG. 6. Ratio of the signature rate of the proposed protocol with TP-TFKGP and that of OTUH-QDS [42] combined with TP-TFQKD, if not considering the postprocessing time, with data sizes 10^{13} and 10^{11} . The message size is 1 Kb and the repetition rate of the laser is 1 GHz. The ratio is more than 0.8 with a transmission distance lower than 500 km.

tion for postprocessing (2.057×10^4 s) is even longer than time for data transmission (10^4 s) for $N = 10^{13}$. The signature rate of OTUH-QDS will be constrained by the efficiency of privacy amplification. That is, in practice the signature rate of OTUH-QDS is lower than the simulation result, while the proposed scheme can overcome this shortcoming. Considering the fact that the proposed protocol can save postprocessing time by even one hundred times, our proposal shows significant improvement in the practical scenario especially when the digital signature tasks are performed at high frequency and the data size is large.

VI. CONCLUSION

In summary, in this paper we prove that keys with partial secrecy leakage can protect the authenticity and integrity of messages if combined with AXU hash functions. Furthermore, we theoretically propose an efficient QDS protocol utilizing imperfect quantum keys without privacy amplification based on the framework of OTUH-

QDS, reducing computational resources and delays of postprocessing without compromising the security. The simulation results demonstrate that the proposed protocol outperforms previous single-bit QDS protocols in terms of both signing efficiency and distance. For instance, for a 1-MB-size message to be signed, the signature rate of the proposed protocol is higher than that of single-bit QDS protocols by over eight orders of magnitude. Specifically, for the protocol based on TP-TFKGP, the transmission distance can reach up to 650 km and still holds a signature rate of 0.01 tps. Moreover, compared with OUTH-QDS, the proposed protocol notably saves the postprocessing time into an endurable range and therefore, significantly improves the practicality. Our scheme is a general framework that can be applied to any existing QKD or QSS protocol, and is highly compatible with future quantum networks and feasible in numerous applications. Additionally, this work, only requiring keys with imperfect secrecy, is a new approach of quantum communication that is different from other quantum secret communication protocols. We suggest that raw quantum keys can be directly used to finish cryptographic tasks including message authentication and digital signatures, indicating the enormous potential of this resource and the possibility of removing the classical postprocessing step in a future quantum world. We believe that the proposed scheme and the idea of utilizing imperfect quantum keys provide a solution for the real implementation of practical and commercial QDS as well as other quantum cryptography tasks in future quantum networks.

ACKNOWLEDGMENTS

This study was supported by the National Natural Science Foundation of China (No. 12274223), the Natural Science Foundation of Jiangsu Province (No. BK20211145), the Fundamental Research Funds for the Central Universities (No. 020414380182), the Key Research and Development Program of Nanjing Jiangbei New Area (No. ZDYD20210101), the Program for Innovative Talents and Entrepreneurs in Jiangsu (No. JSS-CRC2021484), and the Program of Song Shan Laboratory (Included in the management of Major Science and Technology Program of Henan Province) (No. 221100210800-02).

Appendix A: single-bit QDS

1. schematic of single-bit QDS

Here, we first introduce orthogonal encoding QDS [13] as an example of single-bit QDS schemes. Commonly, all single-bit QDS protocols can be segmented into two stages: the distribution stage and messaging stage. The schematic of orthogonal encoding QDS is shown in Fig. 7. distribution stage:

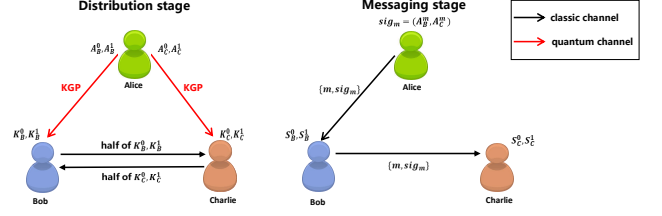


FIG. 7. Orthogonal encoding QDS. In the distribution stage, Alice–Bob and Alice–Charlie independently perform KGP to generate correlated bit strings with limited mismatches. Then Bob and Charlie symmetrize their keys by exchanging half of their keys. In the messaging stage Alice generates the signature depending on the message bit, and sends the message and signature to Bob, who will transfer it to Charlie. Bob and Charlie examine their mismatch and compare it with the threshold to verify the signed message..

(i) For each possible future message $m = 0$ or 1 , Alice uses the KGP to generate four different length L keys, $A_B^0, A_B^1, A_C^0, A_C^1$, where the subscript denotes the participant with whom she performed the KGP and the superscript denotes the future message, to be decided later by Alice. Bob holds the length L strings K_B^0, K_B^1 and Charlie holds the length L strings K_C^0, K_C^1 .

(ii) For each future message, Bob and Charlie symmetrize their keys by choosing half of the bit values in their K_B^m, K_C^m and sending them (as well as the corresponding positions) to the other participant using the Bob–Charlie secret classical channel. They will only use the bits they did not forward and those received from the other participant. Their final symmetrized keys are denoted as S_B^m and S_C^m . Bob (and Charlie) will keep a record of whether an element in S_B^m (S_C^m) came directly from Alice or whether it was forwarded to him by Charlie (or Bob).

messaging stage:

(i) To send a signed one-bit message m , Alice sends (m, sig_m) to the desired recipient (say Bob), where $sig_m = (A_B^m, A_C^m)$.

(ii) Bob checks whether (m, sig_m) matches his S_B^m and records the number of mismatches he finds. He separately checks the part of his key received directly from Alice and the part of the key received from Charlie. If there are fewer than $s_a(L/2)$ mismatches in both halves of the key, where $s_a < 1/2$ is a small threshold determined by the parameters and the desired security level of the protocol, then Bob accepts the message.

(iii) To forward the message to Charlie, Bob forwards the pair (m, sig_m) that he received from Alice.

(iv) Charlie tests for mismatches in the same way, but in order to protect against repudiation by Alice he uses a different threshold. Charlie accepts the forwarded message if the number of mismatches in both halves of his

key is below $s_v(L/2)$ where s_v is another threshold, with $0 < s_a < s_v < 1/2$.

KGP is actually part of QKD protocol except for the error correction and privacy amplification steps. In distribution stage, A_X^m and K_X^m generated through KGP are correlated with limited mismatch, and A_X^m contains fewer mismatches with K_X^m than does any string produced by an eavesdropper, where $X \in \{B, C\}$ represents Bob and Charlie, m is the message. After Bob and Charlie's symmetrization step, Bob holds S_B^m and Charlie holds S_C^m , each containing half of K_B^m and K_C^m . From the perspective of Alice, S_B^m and S_C^m are symmetric. Alice has no information on whether it is Bob's S_B^m or Charlie's S_C^m that contains a particular element of the string (K_B^m, K_C^m) . This protects against repudiation. From the perspective of Bob, S_B^m and S_C^m are asymmetric. Bob has access to all of K_B^m and only half of K_C^m , but, even if he is dishonest, he does not know the half of K_C^m that Charlie chose to keep. This protects against forging.

The framework of non-orthogonal encoding QDS is analogous to that of orthogonal encoding. The difference is that it does not require the symmetrization step. However the signer needs to send the same quantum states to two receivers and only detection events where the two receivers both have clicks are valid.

2. Signing a multi-bit message using single-bit QDS

The framework above only offer a way for signing a one-bit message. To sign multi-bit messages with these protocols, it is not sufficient to directly iterate the protocol on each bits of the message, which will give a chance for an outside or inside attacker to perform forgery attacks [37]. In order to offer information-theoretic security, one must reconstruct the multi-bit message and then sign it bit by bit. This step will make the new message become longer and thus damage the efficiency. To date, the most efficient coding rule is given in Ref. [39], which can be summarized as follows.

Suppose the signer Alice needs to sign an n -bit message $M = m_1||m_2||\dots||m_n$, $m_i \in \{0, 1\}$, $i = 1, 2, \dots, n$. She will encode M into

$$\begin{aligned} \hat{M} = & 1_1||1_2||\dots||1_{x+1}||0||m_1||m_2||\dots||m_x||0|| \\ & ||m_{x+1}||m_{x+2}||\dots||m_{2x}||0|| \\ & \dots \\ & ||m_{\lfloor \frac{n}{x} \rfloor x+1}||m_{\lfloor \frac{n}{x} \rfloor x+2}||\dots||m_n||0||1_1||1_2||\dots||1_{x+1}, \end{aligned} \quad (A1)$$

where x refers to the coding interval, $\lfloor x \rfloor$ is the round down function. To conclude, the coding rule is that the encoder replenishes a '0' in the head of M , and another in the tail. Then, the encoder inserts a '0' every x bits and adds '1' with a number of $x+1$ to both the start and the end.

Denote the length of \hat{M} as h . An iteration of conventional QDS protocols with h rounds on \hat{M} is an information-theoretically secure protocol to sign the

multi-bit message M . According to the encoding rule, $h = n + \lfloor \frac{n}{x} \rfloor + 2x + 4$. For a given n we can optimize x to obtain the minimal h and the maximum efficiency $\eta = \frac{n}{h}$. It is clear that if n is large, the maximum efficiency will be close to 1, but will definitely be less than 1. Thus in our simulation in Sec. V we use the upper bound of deficiency, i.e., assume $h = n$.

Appendix B: Mathematical details

1. Generating an irreducible polynomial

In this section we introduce ways to generate an irreducible polynomial over Gloise fields $\text{GF}(2)$ in random, which is the first step in the messaging stage of our protocol.

Suppose $p(x)$ is a polynomial of order n in $\text{GF}(2)$. $p(x)$ is irreducible means that no polynomials can divide it except the identity element '1' and $p(x)$ itself. The necessary and sufficient condition for $p(x)$ being irreducible can be expressed as:

$$\begin{cases} x^{2^n} \equiv x \pmod{p(x)} \\ \text{gcf}(x^{2^{\frac{n}{d}}} - x, p(x)) = 1 \end{cases} \quad (B1)$$

where d is any prime factor of n , $\text{gcf}(f(x), g(x))$ represents the greatest common factor (GCF) of $f(x)$ and $g(x)$.

In order to randomly generate an irreducible polynomial, one way is to generate polynomials at random and test for irreducibility through the condition above. However, this is quite time consuming and requires a lot of random bits.

A better solution is proposed in Ref. [66]. We can first have an irreducible polynomial of order n , defining the extension field $\text{GF}(2^n)$. Given this, we generate a random element in $\text{GF}(2^n)$ and then compute the minimal polynomial of this element, which will be irreducible. This procedure only needs n random bits and consumes less time. The concrete procedure is as follows.

Denote the initial irreducible polynomial as $f(x)$ and the polynomial generated by random element as $g(x)$. We will calculate the sequence $a_0 = g_0(0)$, $a_1 = g_1(0)$, ..., $a_{2n-1} = g_{2n-1}(0)$, where $g_i(x) = g^i(x) \pmod{f(x)}$. This sequence of $2n$ elements can fully determine the minimal polynomial of $g(x)$, which can be efficiently computed by Berlekamp-Massey algorithm [69]. The result, i.e., the minimal polynomial of $g(x)$, will be the irreducible polynomial we generate.

If choosing generalized division hashing, we need to generate an irreducible polynomial over $\text{GF}(2^k)$. The procedure is the same as that described above. The only difference is that all the calculations need to be done under $\text{GF}(2^k)$.

2. Proof of proposition 1

An LFSR-based Toeplitz hash function can be expressed as $h_{p,s}(M) = H_{nm}M$. The construction of H_{nm} is introduced in Def. 1. Here we follow the expression in Def. 1 and define an $n \times n$ matrix W which is only decided by p .

$$W = \begin{pmatrix} p_{n-1} & p_{n-2} & \dots & p_1 & p_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \quad (\text{B2})$$

then we can express s_i through s and W

$$s_i = W^i s. \quad (\text{B3})$$

Thereafter we rewrite $h_{p,s}(M)$

$$\begin{aligned} h_{p,s}(M) &= H_{nm}M \\ &= \begin{pmatrix} s & s_1 & \dots & s_{m-1} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \dots \\ M_{m-1} \end{pmatrix} \\ &= \sum_{i=0}^{m-1} M_i W^i s \\ &= M(W)s, \end{aligned} \quad (\text{B4})$$

where $M(W) = M_{m-1}W^{m-1} + \dots + m_1W + m_0I$ is an $n \times n$ matrix.

Define $f(x)$ as the characteristic polynomial of the matrix W , and we can calculate it as follows.

$$\begin{aligned} f(x) &= |xI - W| \\ &= \begin{vmatrix} x + p_{n-1} & p_{n-2} & \dots & p_1 & p_0 \\ 1 & x & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & x \end{vmatrix} \\ &= x^n + p_{n-1}x^{n-1} + \dots + p_1x + p_0. \end{aligned} \quad (\text{B5})$$

It is obvious that $f(x) = p(x)$, in other words, $p(x)$ is the characteristic polynomial of the matrix W . Then according to Hamilton-Cayley theorem, $p(W) = 0$. Thereafter, it is trivial that if $p(x)|M(x)$, $M(W) = 0$, and thus $h_{p,s}(M) = M(W)s = 0$.

Appendix C: Calculation details

1. TP-TFQKD and TP-TFKGP in this work

The calculation of TP-TFKGP in this work is analogous to that in TP-TFQKD [43].

When Alice and Bob send intensities k_a and k_b with phase difference θ , the gain corresponding to only one detector (**L** or **R**) clicking is

$$\begin{aligned} Q_{k_a k_b}^{L\theta} &= y_{k_a k_b} (e^{\omega_{k_a k_b} \cos \theta} - y_{k_a k_b}), \\ Q_{k_a k_b}^{R\theta} &= y_{k_a k_b} (e^{-\omega_{k_a k_b} \cos \theta} - y_{k_a k_b}). \end{aligned} \quad (\text{C1})$$

where $y_{k_a k_b} = e^{-\frac{(\eta_a k_a + \eta_b k_b)}{2}}(1 - p_d)$, $\omega_{k_a k_b} = \sqrt{\eta_a k_a \eta_b k_b}$. The overall gain can be expressed as $Q_{k_a k_b} = 1/2\pi \int_0^{2\pi} (Q_{k_a k_b}^{L\theta} + Q_{k_a k_b}^{R\theta}) d\theta = 2y_{k_a k_b} [I_0(\omega_{k_a k_b}) - y_{k_a k_b}]$, where $I_0(x)$ refers to the zero-order modified Bessel functions of the first kind.

The total number for $\{k_a, k_b\}$ is

$$x_{k_a k_b} = N p_{k_a} p_{k_b} Q_{k_a k_b}. \quad (\text{C2})$$

The valid post-matching events on the basis of Z can be divided into two types: correct events $\{\mu_a \mathbf{o}_a, \mathbf{o}_b \mu_b\}$, $\{\mathbf{o}_a \mu_a, \mu_b \mathbf{o}_b\}$, and incorrect events $\{\mu_a \mathbf{o}_a, \mu_b \mathbf{o}_b\}$, $\{\mathbf{o}_a \mu_a, \mathbf{o}_b \mu_b\}$. The corresponding numbers are denoted as n_C^z and n_E^z , respectively, which can be written as

$$n_C^z = x_{\min} \frac{x_{\mathbf{o}_a \mu_b}}{x_0} \frac{x_{\mu_a \mathbf{o}_b}}{x_1} = \frac{x_{\mathbf{o}_a \mu_b} x_{\mu_a \mathbf{o}_b}}{x_{\max}},$$

and

$$n_E^z = x_{\min} \frac{x_{\mathbf{o}_a \mathbf{o}_b}}{x_0} \frac{x_{\mu_a \mu_b}}{x_1} = \frac{x_{\mathbf{o}_a \mathbf{o}_b} x_{\mu_a \mu_b}}{x_{\max}},$$

where $x_0 = x_{\mathbf{o}_a \mu_b} + x_{\mathbf{o}_a \mathbf{o}_b}$, $x_1 = x_{\mu_a \mathbf{o}_b} + x_{\mu_a \mu_b}$, $x_{\min} = \min\{x_0, x_1\}$, and $x_{\max} = \max\{x_0, x_1\}$. s_{11}^z corresponds to the number of successful detection events, where Alice and Bob emit a single photon in different time bins in the Z basis. The overall number of events in the Z basis is

$$n^z = n_C^z + n_E^z. \quad (\text{C3})$$

Considering the misalignment error e_d^z , the number of bit errors in the Z basis is $m^z = (1 - e_d^z)n_E^z + e_d^z n_C^z$. Thus, the bit error rate in the Z basis is

$$E^z = \frac{m^z}{n^z}. \quad (\text{C4})$$

The overall number of “effective” events in the X basis is

$$\begin{aligned} n^x &= \frac{1}{\pi} \int_0^\delta x_{\nu_a \nu_b}^\theta d\theta \\ &= \frac{N p_{\nu_a} p_{\nu_b}}{\pi} \int_\sigma^{\sigma+\delta} y_{\nu_a \nu_b} (e^{\omega_{\nu_a \nu_b} \cos \theta} \\ &\quad + e^{-\omega_{\nu_a \nu_b} \cos \theta} - 2y_{\nu_a \nu_b}) d\theta. \end{aligned} \quad (\text{C5})$$

For simplicity, we only consider the case in which all matched events satisfy $\theta^i - \theta^j = 0$. In this case, when $r_a^i \oplus r_a^j \oplus r_b^i \oplus r_b^j = 0$ (1), the $\{\nu_a^i \nu_a^j, \nu_b^i \nu_b^j\}$ event is considered to be an error event when different detectors (the same detector) click at time bins i and j .

The overall error count in the X basis can be given as

$$m^x = \frac{1}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_a \nu_b}^{\theta} p_E d\theta$$

$$= \frac{2Np_{\nu_a} p_{\nu_b}}{\pi} \int_{\sigma}^{\sigma+\delta} y_{\nu_a \nu_b} \times$$

$$\left[\frac{(1 - y_{\nu_a \nu_b})^2}{e^{\omega_{\nu_a \nu_b} \cos \theta} + e^{-\omega_{\nu_a \nu_b} \cos \theta} - 2y_{\nu_a \nu_b}} - 1 \right] d\theta, \quad (C6)$$

where $p_E = \frac{2q_{\nu_a \nu_b}^{L\theta} q_{\nu_a \nu_b}^{R\theta}}{q_{\nu_a \nu_b}^{\theta} q_{\nu_a \nu_b}^{\theta}}$.

We can then calculate the parameters in Eq. (6) to estimate the key rate and the information leaked after the distribution stage. In the following description, let x^* denote the expected value of x . We denote the number of $\{k_a, k_b\}$ as $x_{k_a k_b}$. We denote the number and error number of events $\{k_a^i k_a^j, k_b^i k_b^j\}$ after post-matching as $n_{k_a^i k_a^j, k_b^i k_b^j}$ and $m_{k_a^i k_a^j, k_b^i k_b^j}$, respectively. For simplicity, we abbreviate $k_a^i k_a^j, k_b^i k_b^j$ as $2k_a, 2k_b$ when $k_a^i = k_a^j$ and $k_b^i = k_b^j$.

(1) \underline{s}_{11}^z .

\underline{s}_{11}^z corresponds to the number of successful detection events, where Alice and Bob emit a single photon in different time bins in the Z basis. Define z_{10} (z_{01}) as the number of events in which Alice (Bob) emits a single photon and Bob (Alice) emits a vacuum state in an $\{\mu_a, \mathbf{o}_b\}$ ($\{\mathbf{o}_a, \mu_b\}$) event. The lower bounds of their expected values are $\underline{z}_{10}^* = Np_{\mu_a} p_{\mathbf{o}_b} \mu_a e^{-\mu_a} \underline{y}_{10}^*$ and $\underline{z}_{01}^* = Np_{\mathbf{o}_a} p_{\mu_b} \mu_b e^{-\mu_b} \underline{y}_{01}^*$, respectively, where \underline{y}_{10}^* and \underline{y}_{01}^* are the corresponding yields. These can be estimated using the decoy-state method

$$\underline{y}_{01}^* \geq \frac{\mu_b}{N(\mu_b \nu_b - \nu_b^2)} \left(\frac{e^{\nu_b} \underline{x}_{\mathbf{o}_a \nu_b}^*}{p_{\mathbf{o}_a} p_{\nu_b}} - \frac{\nu_b^2 e^{\mu_b} \bar{x}_{\mathbf{o}_a \mu_b}^*}{\mu_b^2 p_{\mathbf{o}_a} p_{\mu_b}} - \frac{\mu_b^2 - \nu_b^2}{\mu_b^2} \frac{\bar{x}_{\mathbf{o}_o}^{d*}}{p_{\mathbf{o}_a \mathbf{o}_b}^d} \right), \quad (C7)$$

$$\underline{y}_{10}^* \geq \frac{\mu_a}{N(\mu_a \nu_a - \nu_a^2)} \left(\frac{e^{\nu_a} \underline{x}_{\nu_a \mathbf{o}_b}^*}{p_{\nu_a} p_{\mathbf{o}_b}} - \frac{\nu_a^2 e^{\mu_a} \bar{x}_{\mu_a \mathbf{o}_b}^*}{\mu_a^2 p_{\mu_a} p_{\mathbf{o}_b}} - \frac{\mu_a^2 - \nu_a^2}{\mu_a^2} \frac{\bar{x}_{\mathbf{o}_o}^{d*}}{p_{\mathbf{o}_a \mathbf{o}_b}^d} \right), \quad (C8)$$

where $x_{\mathbf{o}_o}^d = x_{\mathbf{o}_a \mathbf{o}_b} + x_{\mathbf{o}_a \mathbf{o}_b} + x_{\mathbf{o}_a \mathbf{o}_b}$ represents the number of events where at least one user chooses the declare-vacuum state and $p_{\mathbf{o}_o}^d = p_{\mathbf{o}_a} p_{\mathbf{o}_b} + p_{\mathbf{o}_a} p_{\mathbf{o}_b} + p_{\mathbf{o}_a} p_{\mathbf{o}_b}$ refers to the corresponding probability. Thus, the lower bound of \underline{s}_{11}^z is given by

$$\underline{s}_{11}^{z*} = \frac{\underline{z}_{10}^* \underline{z}_{01}^*}{x_{\max}}. \quad (C9)$$

(2) $\underline{s}_{0\mu_b}^z$. $\underline{s}_{0\mu_b}^z$ represents the number of events in the Z basis, Alice emits a zero-photon state in the two matched time bins, and the total intensity of Bob's pulses is μ_b . We define z_{00} ($z_{0\mu_b}$) as the number of detection events where the state sent by Alice collapses to the vacuum state in the $\{\mu_a, \mathbf{o}_b\}$ ($\{\mu_a, \mu_b\}$) event. The lower bound

of the expected values is $\underline{z}_{00}^* = p_{\mu_a} p_{\mathbf{o}_b} e^{-\mu_a} \underline{x}_{\mathbf{o}_o}^{d*} / p_{\mathbf{o}_a \mathbf{o}_b}^d$ and $\underline{z}_{0\mu_b}^* = p_{\mu_a} p_{\mu_b} e^{-\mu_a} \underline{x}_{\mathbf{o}_a \mu_b}^* / p_{\mathbf{o}_a} p_{\mu_b}$, respectively. Here, we employ the relationship between the expected value $\underline{x}_{\mathbf{o}_a \mu_b}^* = p_{\mathbf{o}_a} \underline{x}_{\mathbf{o}_a \mu_b}^* / p_{\mathbf{o}_a}$, and $\underline{x}_{\mathbf{o}_a \mathbf{o}_b}^* = p_{\mathbf{o}_a} p_{\mathbf{o}_b} \underline{x}_{\mathbf{o}_o}^{d*} / p_{\mathbf{o}_o}^d$. The lower bound of $\underline{s}_{0\mu_b}^z$ can be written as

$$\underline{s}_{0\mu_b}^{z*} = \frac{\underline{x}_{\mathbf{o}_a \mu_b}^* \underline{z}_{00}^*}{x_{\max}} + \frac{\underline{x}_{\mathbf{o}_a \mathbf{o}_b}^* \underline{z}_{0\mu_b}^*}{x_{\max}} \quad (C10)$$

(3) \underline{s}_{11}^x . We define the phase difference between Alice and Bob as $\theta = \theta_a - \theta_b + \phi_{ab}$. All valid events in the X basis can be grouped according to the phase difference θ ($\in \{-\delta, \delta\} \cup \{\pi - \delta, \pi + \delta\}$), and the corresponding number in the $\{k_a, k_b\}$ event is denoted as $x_{k_a k_b}^{\theta}$. In the post-matching step, two time bins are matched if they have the same phase difference θ . Suppose the global phase difference θ is a randomly and uniformly distributed value, and considering the angle of misalignment in the X basis σ , the expected number of single-photon pairs can be given by

$$\underline{s}_{11}^{x*} = \frac{1}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_a \nu_b}^{\theta} \times 2 \frac{\nu_b e^{-(\nu_a + \nu_b)} \underline{y}_{01}^*}{q_{\nu_a \nu_b}^{\theta}} \frac{\nu_a e^{-(\nu_a + \nu_b)} \underline{y}_{10}^*}{q_{\nu_a \nu_b}^{\theta}} d\theta$$

$$= \frac{Np_{\nu_a} p_{\nu_b}}{\pi} \int_{\sigma}^{\sigma+\delta} \frac{2\nu_a \nu_b e^{-2(\nu_a + \nu_b)} \underline{y}_{01}^* \underline{y}_{10}^*}{q_{\nu_a \nu_b}^{\theta}} d\theta, \quad (C11)$$

where $q_{\nu_a \nu_b}^{\theta}$ is the gain when Alice chooses intensity ν_a , and Bob chooses the intensity ν_b with phase difference θ and $x_{\nu_a \nu_b}^{\theta} = Np_{\nu_a} p_{\nu_b} q_{\nu_a \nu_b}^{\theta}$.

(4) \bar{t}_{11}^x . For single-photon pairs, the expected value of the phase error rate in the Z basis is equal to the expected value of the bit error rate in the X basis. Therefore, we first calculate the number of errors of the single-photon pairs in the X basis t_{11}^x . The upper bound of t_{11}^x can be expressed as

$$\bar{t}_{11}^x \leq m^x - \underline{(m_{\nu_a 0, \nu_b 0} + m_{0\nu_a, 0\nu_b})} + \bar{m}_{00,00}, \quad (C12)$$

where $(m_{\nu_a 0, \nu_b 0} + m_{0\nu_a, 0\nu_b})$ is the error count when the states sent by Alice and Bob in time bin i (j) both collapse to the vacuum state in events $\{2\nu_a, 2\nu_b\}$, and $m_{00,00}$ corresponds to the event where the states sent by Alice and Bob both collapse to vacuum states in events $\{2\nu_a, 2\nu_b\}$. The expected counts $(n_{\nu_a 0, \nu_b 0} + n_{0\nu_a, 0\nu_b})^*$ and $\bar{n}_{00,00}$ can be expressed as

$$(n_{\nu_a 0, \nu_b 0} + n_{0\nu_a, 0\nu_b})^* = \frac{2}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_a \nu_b}^{\theta} \frac{e^{-(\nu_a + \nu_b)} \underline{q}_{00}^*}{q_{\nu_a \nu_b}^{\theta}} d\theta$$

$$= \frac{\delta Np_{\nu_a} p_{\nu_b} e^{-(\nu_a + \nu_b)} \underline{q}_{00}^*}{\pi} \quad (C13)$$

and

$$\bar{n}_{00,00}^* = \frac{1}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_a \nu_b}^{\theta} \left(\frac{e^{-(\nu_a + \nu_b)} \bar{q}_{00}^*}{q_{\nu_a \nu_b}^{\theta}} \right)^2 d\theta$$

$$= \frac{Np_{\nu_a} p_{\nu_b}}{\pi} \int_{\sigma}^{\sigma+\delta} \frac{e^{-2(\nu_a + \nu_b)} (\bar{q}_{00}^*)^2}{q_{\nu_a \nu_b}^{\theta}} d\theta \quad (C14)$$

respectively. Here $q_{00}^* = x_{o_a o_b}^{d*}/(Np_{oo}^d)$. Using the fact that the error rate of the vacuum state is always $1/2$, we have $(m_{\nu_a 0, \nu_b 0} + m_{0\nu_a, 0\nu_b})^* = \frac{1}{2}(n_{\nu_a 0, \nu_b 0} + n_{0\nu_a, 0\nu_b})^*$ and $\bar{m}_{00,00}^* = \frac{1}{2}\bar{n}_{00,00}^*$. Hence the upper bound of the bit error rate in the X basis can be given by

$$\bar{e}_{11}^x = \bar{t}_{11}^x / \bar{s}_{11}^x. \quad (C15)$$

(5) $\bar{\phi}_{11}^z$. For a failure probability ε , the upper bound of the phase error rate ϕ_{11}^z can be obtained by using the random sampling without replacement [70]

$$\bar{\phi}_{11}^z \leq \bar{e}_{11}^x + \gamma^U(\bar{s}_{11}^z, \bar{s}_{11}^x, \bar{e}_{11}^x, \epsilon), \quad (C16)$$

where

$$\gamma^U(n, k, \lambda, \epsilon) = \frac{\frac{(1-2\lambda)AG}{n+k} + \sqrt{\frac{A^2G^2}{(n+k)^2} + 4\lambda(1-\lambda)G}}{2 + 2\frac{A^2G}{(n+k)^2}}, \quad (C17)$$

with $A = \max\{n, k\}$ and $G = \frac{n+k}{nk} \ln \frac{n+k}{2\pi nk\lambda(1-\lambda)\epsilon^2}$.

(6) \bar{s}_{11}^z , $\bar{s}_{0\mu_b}^z$ and $\bar{\phi}_{11}^z$. Finally we can estimate the parameters in Eq. (6), i.e., the lower bound of vacuum events and single-photon pairs in a selected key group \bar{s}_{11L}^z and $\bar{s}_{0\mu_bL}^z$, and the upper bound of the phase error rate of the n -bit group $\bar{\phi}_{11U}^z$. They can be obtained from the parameters above by using the random sampling without replacement.

$$\begin{aligned} \bar{s}_{11}^z &\geq n [\bar{s}_{11}^z/n^z - \gamma^U(n, n^z - n, \bar{s}_{11}^z/n^z, \epsilon)], \\ \bar{s}_{0\mu_b}^z &\geq n [\bar{s}_{0\mu_b}^z/n^z - \gamma^U(n, n^z - n, \bar{s}_{0\mu_b}^z/n^z, \epsilon)], \\ \bar{\phi}_{11}^z &\leq \bar{\phi}_{11}^z + \gamma^U(\bar{s}_{11}^z, \bar{s}_{11}^z - \bar{s}_{11}^z, \bar{\phi}_{11}^z, \epsilon). \end{aligned} \quad (C18)$$

(7) l_{key} . We can also obtain the length of final keys of TP-TFQKD, which can be used to simulate the performance of OTUH-QDS in Fig. 4.

$$\begin{aligned} l_{key} &= \bar{s}_{0\mu_b}^z + \bar{s}_{11}^z \left[1 - H(\bar{\phi}_{11}^z) \right] - n^z fH(E^z) \\ &\quad - \log_2 \frac{2}{\epsilon_{cor}} - 2 \log_2 \frac{1}{2\epsilon_{PA}}, \end{aligned} \quad (C19)$$

where ϵ_{PA} is the failure probability of privacy amplification.

2. BB84-KGP in BB84-QDS and this work

Both BB84-QDS and this work utilize decoy-state BB84-KGP to generate correlated bit strings. According to Ref. [71] we can estimate the number of vacuum events and single-photon events under X basis,

$$s_{X,0} \geq \tau_0 \frac{\mu_2 n_{X,\mu_3}^- - \mu_3 n_{X,\mu_2}^+}{\mu_2 - \mu_3}, \quad (C20)$$

$$s_{X,1} \geq \frac{\tau_1 \mu_1 \left[n_{X,\mu_2}^- - n_{X,\mu_3}^+ - \frac{\mu_2^2 - \mu_3^2}{\mu_1^2} (n_{X,\mu_1}^+ - \frac{s_{X,0}}{\tau_0}) \right]}{\mu_1(\mu_2 - \mu_3) - \mu_2^2 + \mu_3^2}. \quad (C21)$$

where $\tau_n := \sum_{k \in \mathcal{K}} e^{-k} k^n p_k / n!$ is the probability that Alice sends a n -photon state, and

$$n_{X,k}^\pm := \frac{e^k}{p_k} \left(n_{X,k} \pm \sqrt{\frac{n_X}{2} \log \frac{21}{\varepsilon_{\text{sec}}}} \right), \quad \forall k \in \mathcal{K}.$$

We can also calculate the number of vacuum events, $s_{Z,0}$, and the number of single-photon events, $s_{Z,1}$, for $\mathcal{Z} = \cup_{k \in \mathcal{K}} \mathcal{Z}_k$, i.e., by using Eqs. (C20) and (C21) with statistics from the basis Z . Then we can obtain the phase error rate of the single-photon events in the X basis by

$$\phi_{X,1} := \frac{c_{X,1}}{s_{X,1}} \leq \frac{v_{Z,1}}{s_{Z,1}} + \gamma^U \left(s_{Z,1}, s_{X,1}, \frac{v_{Z,1}}{s_{Z,1}}, \varepsilon_{\text{sec}} \right), \quad (C22)$$

where

$$v_{Z,1} \leq \tau_1 \frac{m_{Z,\mu_2}^+ - m_{Z,\mu_3}^-}{\mu_2 - \mu_3},$$

$$m_{Z,k}^\pm := \frac{e^k}{p_k} \left(m_{Z,k} \pm \sqrt{\frac{m_Z}{2} \log \frac{21}{\varepsilon_{\text{sec}}}} \right), \quad \forall k \in \mathcal{K},$$

$$\gamma^U(n, k, \lambda, \epsilon) = \frac{\frac{(1-2\lambda)AG}{n+k} + \sqrt{\frac{A^2G^2}{(n+k)^2} + 4\lambda(1-\lambda)G}}{2 + 2\frac{A^2G}{(n+k)^2}}$$

The total number of events under X basis is $n_X = \sum_{k \in \mathcal{K}} n_{X,k}$ and the number of error events is $m_X = \sum_{k \in \mathcal{K}} m_{X,k}$.

In BB84-QDS, the unknown information to the attacker is given by

$$\mathcal{H} = \bar{s}_{X,0} + \bar{s}_{X,1}(1 - h(\bar{\phi}_{X,1})). \quad (C23)$$

In our protocol based on BB84-KGP, we need to estimate parameters in a selected n -bit group, i.e., the lower bound of number of vacuum events and single-photon events under X basis $\bar{s}_{X,0}^n$ and $\bar{s}_{X,1}^n$, and the upper bound of the phase error rate of the single-photon events in the X basis $\bar{\phi}_{X,1}^n$.

$$s_{X,0}^n \geq n [\bar{s}_{X,0}^n/n^z - \gamma^U(n, n^z - n, \bar{s}_{X,0}^n/n^z, \epsilon)], \quad (C24)$$

$$s_{X,1}^n \geq n [\bar{s}_{X,1}^n/n^z - \gamma^U(n, n^z - n, \bar{s}_{X,1}^n/n^z, \epsilon)], \quad (C25)$$

$$\phi_{X,1}^n \leq \phi_{X,1} + \gamma^U(\bar{s}_{X,1}^n, \bar{s}_{X,1} - \bar{s}_{X,1}^n, \bar{\phi}_{X,1}^n, \epsilon). \quad (C26)$$

Finally we can obtain

$$\mathcal{H} = \bar{s}_{X,0}^n + \bar{s}_{X,1}^n \left[1 - h(\bar{\phi}_{X,1}^n) \right] - \lambda_{EC}, \quad (C27)$$

where $\lambda_{EC} = nh(m_X/n_X)$.

3. SNS-KGP and SNS-QDS with random pairing

We first follow the calculation in Ref. [72]. Alice and Bob obtain N_{jk} ($jk = \{00, 01, 02, 10, 20\}$) instances when Alice sends intensity j and Bob sends state k . Here '1' and '2' represent the two intensities used in the KGP. After the sifted step, Alice and Bob obtain n_{jk} one-detector heralded events. We denote the counting rate of source jk as $S_{jk} = n_{jk}/N_{jk}$. With all these definitions, we have

$$\begin{aligned} N_{00} &= [(1-p_z)^2 p_0^2 + 2(1-p_z)p_z p_0 p_{z0}] N, \\ N_{01} &= N_{10} = [(1-p_z)^2 p_0 p_1 + (1-p_z)p_z p_{z0} p_1] N, \\ N_{02} &= N_{20} = [(1-p_z)^2 (1-p_0-p_1)p_0 \\ &\quad + (1-p_z)p_z p_{z0} (1-p_0-p_1)] N. \end{aligned} \quad (C28)$$

In addition, we need to define two new subsets of X_1 windows, $C_{\Delta+}$ and $C_{\Delta-}$, to estimate the upper bound of e_1^{ph} . The number of instances in $C_{\Delta\pm}$ is

$$N_{\Delta\pm} = \frac{\Delta}{2\pi} (1-p_z)^2 p_1^2 N. \quad (C29)$$

We denote the number of effective events of right detectors responding from $C_{\Delta+}$ as $n_{\Delta+}^R$, and the number of effective events of left detectors responding from $C_{\Delta-}$ as $n_{\Delta-}^L$. And we obtain the counting error rate of $C_{\Delta\pm}$, $T_{\Delta} = \frac{n_{\Delta+}^R + n_{\Delta-}^L}{2N_{\Delta\pm}}$.

If we denote the expected value of the counting rate of **untagged** photons as s_1^{Z*} , the lower bound of s_1^{Z*} is

$$\begin{aligned} s_1^{Z*} \geq \underline{s}_1^{Z*} &= \frac{1}{2\mu_1\mu_2(\mu_2 - \mu_1)} [\mu_2^2 e^{\mu_1} (\underline{S}_{01}^* + \underline{S}_{10}^*) \\ &\quad - \mu_1^2 e^{\mu_2} (\overline{S}_{02}^* + \overline{S}_{20}^*) - 2(\mu_2^2 - \mu_1^2) \overline{S}_{00}^*], \end{aligned} \quad (C30)$$

where S_{jk}^* is the expected value of S_{jk} , and \overline{S}_{jk}^* and \underline{S}_{jk}^* are the upper bound and lower bound of S_{jk}^* when we estimate the expected value from its observed value.

The expected value of the phase-flip error rate of the **untagged** photons satisfies

$$e_1^{ph*} \leq \overline{e}_1^{ph*} = \frac{\overline{T}_{\Delta}^* - \frac{1}{2} e^{-2\mu_1} \underline{S}_{00}^*}{2\mu_1 e^{-2\mu_1} \underline{s}_1^{Z*}}. \quad (C31)$$

Here we use the fact that the error rate of vacuum state is always $\frac{1}{2}$.

If the total transmittance of the experimental setups is η , then we have

$$\begin{aligned} n_{00} &= 2p_d(1-p_d)N_{00}, \\ n_{01} &= n_{10} = 2 \left[(1-p_d)e^{\eta\mu_1/2} - (1-p_d)^2 e^{-\eta\mu_1} \right] N_{01}, \\ n_{02} &= n_{20} = 2 \left[(1-p_d)e^{\eta\mu_2/2} - (1-p_d)^2 e^{-\eta\mu_2} \right] N_{02}, \\ n_t &= n_{signal} + n_{error}, \\ E_z &= \frac{n_{error}}{n_t}, \\ n_{\Delta+}^R &= n_{\Delta-}^L = [T_X(1-2e_d) + e_d S_X] N_{\Delta\pm}, \end{aligned}$$

where $N_{00}, N_{01}, N_{10}, N_{02}, N_{20}, N_{\Delta\pm}$ are defined in Eqs. (C28) and (C29), and

$$\begin{aligned} n_{signal} &= 4Np_z^2 p_{z0}(1-p_{z0}) \left[(1-p_d)e^{-\eta\mu_z/2} \right. \\ &\quad \left. - (1-p_d)^2 e^{-2\eta\mu_z} \right], \\ n_{error} &= 2Np_z^2(1-p_{z0})^2 \left[(1-p_d)e^{-\eta\mu_z} I_0(\eta\mu_z) \right. \\ &\quad \left. - (1-p_d)^2 e^{-2\eta\mu_z} \right] + 2Np_z^2 p_{z0}^2 p_d(1-p_d), \\ T_X &= \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} (1-p_d)e^{-2\eta\mu_1 \cos^2 \frac{\delta}{2}} d\delta \\ &\quad - (1-p_d)^2 e^{-2\eta\mu_1}, \\ S_X &= \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} (1-p_d)e^{-2\eta\mu_1 \sin^2 \frac{\delta}{2}} d\delta \\ &\quad - (1-p_d)^2 e^{-2\eta\mu_1} + T_X, \end{aligned}$$

where $I_0(x)$ is the 0-order hyperbolic Bessel functions of the first kind.

In SNS-QDS, the unknown information to the attacker is given by

$$\mathcal{H} = \underline{s}_1^{Z*} (1 - h(\overline{e}_1^{ph*})). \quad (C32)$$

In our protocol based on SNS-KGP, there is

$$\begin{aligned} \underline{s}_{1n}^{Z*} &= n \left[\underline{s}_1^{Z*} - \gamma^U(n, n_Z - n, \underline{s}_1^{Z*}/n_Z, \epsilon) \right], \\ \overline{e}_{1n}^{ph*} &= n \left[\overline{e}_1^{ph*} + \gamma^U \left(\underline{s}_{1n}^{Z*}, \underline{s}_1^{Z*} - \underline{s}_{1n}^{Z*} \overline{e}_1^{ph*}, \epsilon \right) \right]. \end{aligned} \quad (C33)$$

and

$$\mathcal{H} = \underline{s}_{1n}^{Z*} \left[1 - h(\overline{e}_{1n}^{ph*}) \right] - \lambda_{EC}, \quad (C34)$$

where $\lambda_{EC} = nh(E_z)$.

In SNS-QDS with random pairing, we follow the calculation in Ref. [24]. After random pairing there are two different phase error rates

$$\tilde{e}_1^{ph} = \frac{(e_1^{ph})^2}{(e_1^{ph})^2 + (1 - e_1^{ph})^2}, \quad (C35)$$

$$\tilde{e}_2^{ph} = \frac{1}{2}, \quad (C36)$$

and a new bit flip error rate

$$E' = 2E_z(1 - E_z). \quad (C37)$$

The proportion of untagged bits after random pairing is

$$\Delta'_{un} = \Delta_{un}^2 + 2\Delta_{un}(1 - \Delta_{un}), \quad (C38)$$

where $\Delta_{un} = Np_z^2 p_{z0}(1-p_{z0})s_1^Z/n_t$ is the proportion of untagged bits before random pairing. The unknown information to the attacker is given by

$$\begin{aligned} \mathcal{H} &= \Delta'_{un} - \Delta_{un}^2 \left[p_1 H(\tilde{e}_1^{ph}) + (1-p_1) H(\tilde{e}_2^{ph}) \right] \\ &\quad - 2\Delta_{un}(1 - \Delta_{un}) H(e_1^{ph}), \end{aligned} \quad (C39)$$

where $p_1 = (e_1^{ph})^2 + (1 - e_1^{ph})^2$.

Appendix D: Error correction and privacy amplification

In this section we introduce our simulation of error correction and privacy amplification in Table II. We use the simulated data of TP-TFKGP at the distance of 400 km, which can be calculated by Eqs. C3, C4, C19 in Appendix C. We implement our simulation on a desktop computer with an Intel i5-10400 CPU (with RAM of 8 GB).

We use improved Cascade protocol to perform error correction to correct 300 (or 39830) errors among 1.267×10^6 (or 1.695×10^8) bits. The detailed procedure of improved Cascade protocol can be seen in Ref. [65], where users first block their keys, and then perform binary process on each block to correct the errors and start trace-back section to check the error, until there are no

errors in each block. The results show that time consumption is 3.62 and 930.98 seconds with data sizes 10^{11} and 10^{13} , respectively.

In privacy amplification step, Alice chooses a random universal₂ hash function and performs it on the n_Z -bit keys after error correction to obtain l -bit final keys. The choice of function is communicated to Bob, who also uses it to obtain his l -bit final keys. In the simulation we utilize a random Toeplitz matrix as the universal₂ hash function. When the data size is 10^{13} , the matrix is too large that it exceeds the storage of our computer. Thus in the algorithm we block the matrix into 10×10 (100) submatrices with the same size to accomplish the calculation. For hash manipulations of every submatrix we follow the procedure in Ref. [73], where fast Fourier transform is used to speed up calculation time. In the simulation it takes 2.98 and 2.057×10^4 seconds with data sizes 10^{11} and 10^{13} , respectively.

-
- [1] W. Diffie and M. Hellman, New directions in cryptography, IEEE trans. Inf. Theory **22**, 644 (1976).
 - [2] R. A. DeMillo, *Foundations of secure computation*, Tech. Rep. (Georgia Institute of Technology, 1978).
 - [3] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM **21** (1978).
 - [4] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE trans. Inf. Theory **31**, 469 (1985).
 - [5] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, The first collision for full sha-1, in *Advances in Cryptology – CRYPTO 2017*, LNCS, Vol. 10401 (Springer, 2017) pp. 570–596.
 - [6] F. Boudot, P. Gaudry, A. Guillevis, N. Heninger, E. Thomé, and P. Zimmermann, Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment, in *Annual International Cryptology Conference* (Springer, 2020) pp. 62–91.
 - [7] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in *Proceedings 35th annual symposium on foundations of computer science* (IEEE, 1994) pp. 124–134.
 - [8] D. Gottesman and I. Chuang, Quantum digital signatures, arXiv preprint quant-ph/0105032 (2001).
 - [9] P. J. Clarke, R. J. Collins, V. Dunjko, E. Andersson, J. Jeffers, and G. S. Buller, Experimental demonstration of quantum digital signatures using phase-encoded coherent states of light, Nat. Commun. **3**, 1174 (2012).
 - [10] R. J. Collins, R. J. Donaldson, V. Dunjko, P. Wallden, P. J. Clarke, E. Andersson, J. Jeffers, and G. S. Buller, Realization of quantum digital signatures without the requirement of quantum memory, Phys. Rev. Lett. **113**, 040502 (2014).
 - [11] V. Dunjko, P. Wallden, and E. Andersson, Quantum digital signatures without quantum memory, Phys. Rev. Lett. **112**, 040502 (2014).
 - [12] H.-L. Yin, Y. Fu, and Z.-B. Chen, Practical quantum digital signature, Phys. Rev. A **93**, 032316 (2016).
 - [13] R. Amiri, P. Wallden, A. Kent, and E. Andersson, Secure quantum signatures using insecure quantum channels, Phys. Rev. A **93**, 032325 (2016).
 - [14] I. V. Puthoor, R. Amiri, P. Wallden, M. Curty, and E. Andersson, Measurement-device-independent quantum digital signatures, Phys. Rev. A **94**, 022328 (2016).
 - [15] T. Shang, Q. Lei, and J. Liu, Quantum random oracle model for quantum digital signature, Phys. Rev. A **94**, 042314 (2016).
 - [16] Y.-G. Yang, Z.-C. Liu, J. Li, X.-B. Chen, H.-J. Zuo, Y.-H. Zhou, and W.-M. Shi, Theoretically extensible quantum digital signature with starlike cluster states, Quantum Inf. Process. **16**, 12 (2017).
 - [17] M. Thornton, H. Scott, C. Croal, and N. Korolkova, Continuous-variable quantum digital signatures over insecure channels, Phys. Rev. A **99**, 032341 (2019).
 - [18] W. Qu, Y. Zhang, H. Liu, T. Dou, J. Wang, Z. Li, S. Yang, and H. Ma, Multi-party ring quantum digital signatures, J. Opt. Soc. Am. B **36**, 1335 (2019).
 - [19] C.-M. Zhang, Y. Zhu, J.-J. Chen, and Q. Wang, Practical quantum digital signature with configurable decoy states, Quantum Inf. Process. **19**, 151 (2020).
 - [20] Y.-S. Lu, X.-Y. Cao, C.-X. Weng, J. Gu, Y.-M. Xie, M.-G. Zhou, H.-L. Yin, and Z.-B. Chen, Efficient quantum digital signatures without symmetrization step, Opt. Express **29**, 10162 (2021).
 - [21] C.-H. Zhang, X. Zhou, C.-M. Zhang, J. Li, and Q. Wang, Twin-field quantum digital signatures, Opt. Lett. **46**, 3757 (2021).
 - [22] W. Zhao, R. Shi, J. Shi, P. Huang, Y. Guo, and D. Huang, Multibit quantum digital signature with continuous variables using basis encoding over insecure channels, Phys. Rev. A **103**, 012410 (2021).
 - [23] C.-X. Weng, Y.-S. Lu, R.-Q. Gao, Y.-M. Xie, J. Gu, C.-L. Li, B.-H. Li, H.-L. Yin, and Z.-B. Chen, Secure and practical multiparty quantum digital signatures, Opt. Express **29**, 27661 (2021).
 - [24] J.-Q. Qin, C. Jiang, Y.-L. Yu, and X.-B. Wang, Quantum digital signatures with random pairing, Phys. Rev. Applied **17**, 044047 (2022).
 - [25] M.-H. Zhang, J.-H. Xie, J.-Y. Wu, L.-Y. Yue, C. He,

- Z.-W. Cao, and J.-Y. Peng, Practical long-distance twin-field quantum digital signatures, *Quantum Inf. Process.* **21**, 150 (2022).
- [26] 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, T.-Y. Chen, Z.-B. Chen, and J.-W. Pan, Experimental quantum digital signature over 102 km, *Phys. Rev. A* **95**, 032334 (2017).
- [27] R. J. Collins, R. Amiri, M. Fujiwara, T. Honjo, K. Shimizu, K. Tamaki, M. Takeoka, M. Sasaki, E. Andersson, and G. S. Buller, Experimental demonstration of quantum digital signatures over 43 db channel loss using differential phase shift quantum key distribution, *Sci. Rep.* **7**, 3235 (2017).
- [28] 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, E. Andersson, Z. Wang, Y. Liu, X. Jiang, X. Ma, Q. Zhang, M. Curty, T.-Y. Chen, and J.-W. Pan, Experimental measurement-device-independent quantum digital signatures over a metropolitan network, *Phys. Rev. A* **95**, 042338 (2017).
- [29] 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**, 1098 (2017).
- [30] 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. Applied* **10**, 034033 (2018).
- [31] 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, X.-Y. Zhou, G.-C. Guo, and Z.-F. Han, Practical quantum digital signature with a gigahertz bb84 quantum key distribution system, *Opt. Lett.* **44**, 139 (2019).
- [32] H.-J. Ding, J.-J. Chen, L. Ji, X.-Y. Zhou, C.-H. Zhang, C.-M. Zhang, and Q. Wang, 280-km experimental demonstration of a quantum digital signature with one decoy state, *Opt. Lett.* **45**, 1711 (2020).
- [33] S. Richter, M. Thornton, I. Khan, H. Scott, K. Jaksch, U. Vogl, B. Stiller, G. Leuchs, C. Marquardt, and N. Krolkova, Agile and versatile quantum communication: Signatures and secrets, *Phys. Rev. X* **11**, 011038 (2021).
- [34] M.-C. Roehsner, J. A. Kettlewell, J. Fitzsimons, and P. Walther, Probabilistic one-time programs using quantum entanglement, *npj Quantum Inf.* **7**, 98 (2021).
- [35] M.-C. Roehsner, J. A. Kettlewell, T. B. Batalhão, J. F. Fitzsimons, and P. Walther, Quantum advantage for probabilistic one-time programs, *Nat. Commun.* **9**, 5225 (2018).
- [36] Y. Pelet, I. V. Puthoor, N. Venkatachalam, S. Wengerovsky, M. Loncaric, S. P. Neumann, B. Liu, Z. Samec, M. Stipčević, R. Ursin, *et al.*, Unconditionally secure digital signatures implemented in an 8-user quantum network, *New J. Phys.* (2022).
- [37] T.-Y. Wang, X.-Q. Cai, Y.-L. Ren, and R.-L. Zhang, Security of quantum digital signatures for classical messages, *Sci. Rep.* **5**, 9321 (2015).
- [38] T.-Y. Wang, J.-F. Ma, and X.-Q. Cai, The postprocessing of quantum digital signatures, *Quantum Inf. Process.* **16**, 19 (2017).
- [39] H. Zhang, X.-B. An, C.-H. Zhang, C.-M. Zhang, and Q. Wang, High-efficiency quantum digital signature scheme for signing long messages, *Quantum Inf. Process.* **18**, 3 (2019).
- [40] X.-Q. Cai, T.-Y. Wang, C.-Y. Wei, and F. Gao, Cryptanalysis of multiparty quantum digital signatures, *Quantum Inf. Process.* **18**, 252 (2019).
- [41] X. Yao, X. Liu, R. Xue, H. Wang, H. Li, Z. Wang, L. You, Y. Huang, and W. Zhang, Multi-bit quantum digital signature based on quantum temporal ghost imaging, *arXiv preprint arXiv:1901.03004* (2019).
- [42] 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).
- [43] Y.-M. Xie, C.-X. Weng, Y.-S. Lu, Y. Fu, Y. Wang, H.-L. Yin, and Z.-B. Chen, Scalable high-rate twin-field quantum key distribution networks without constraint of probability and intensity, *Phys. Rev. A* **107**, 042603 (2023).
- [44] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, *Theor. Comput. Sci.* **560**, 7 (2014).
- [45] H.-K. Lo, M. Curty, and B. Qi, Measurement-device-independent quantum key distribution, *Phys. Rev. Lett.* **108**, 130503 (2012).
- [46] S. L. Braunstein and S. Pirandola, Side-channel-free quantum key distribution, *Phys. Rev. Lett.* **108**, 130502 (2012).
- [47] 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).
- [48] 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).
- [49] X. Ma, P. Zeng, and H. Zhou, Phase-matching quantum key distribution, *Phys. Rev. X* **8**, 031043 (2018).
- [50] 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).
- [51] W.-B. Liu, C.-L. Li, Y.-M. Xie, C.-X. Weng, J. Gu, X.-Y. Cao, Y.-S. Lu, B.-H. Li, H.-L. Yin, and Z.-B. Chen, Homodyne detection quadrature phase shift keying continuous-variable quantum key distribution with high excess noise tolerance, *PRX Quantum* **2**, 040334 (2021).
- [52] Y.-M. Xie, Y.-S. Lu, C.-X. Weng, X.-Y. Cao, Z.-Y. Jia, Y. Bao, Y. Wang, Y. Fu, H.-L. Yin, and Z.-B. Chen, Breaking the rate-loss bound of quantum key distribution with asynchronous two-photon interference, *PRX Quantum* **3**, 020315 (2022).
- [53] P. Zeng, H. Zhou, W. Wu, and X. Ma, Mode-pairing quantum key distribution, *Nat. Commun.* **13**, 3903 (2022).
- [54] J. Gu, X.-Y. Cao, Y. Fu, Z.-W. He, Z.-J. Yin, H.-L. Yin, and Z.-B. Chen, Experimental measurement-device-independent type quantum key distribution with flawed and correlated sources, *Sci. Bull.* **67**, 2167 (2022).
- [55] Y. Fu, H.-L. Yin, T.-Y. Chen, and Z.-B. Chen, Long-distance measurement-device-independent multiparty quantum communication, *Phys. Rev. Lett.* **114**, 090501 (2015).
- [56] J. Gu, X.-Y. Cao, H.-L. Yin, and Z.-B. Chen, Differential phase shift quantum secret sharing using a twin field, *Opt. Express* **29**, 9165 (2021).
- [57] A. Shen, X.-Y. Cao, Y. Wang, Y. Fu, J. Gu, W.-B. Liu,

- C.-X. Weng, H.-L. Yin, and Z.-B. Chen, Experimental quantum secret sharing based on phase encoding of coherent states, *Sci. China-Phys. Mech. Astron.* **66**, 260311 (2023).
- [58] Z. Li, X.-Y. Cao, C.-L. Li, C.-X. Weng, J. Gu, H.-L. Yin, and Z.-B. Chen, Finite-key analysis for quantum conference key agreement with asymmetric channels, *Quantum Sci. Technol.* **6**, 045019 (2021).
- [59] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography* (CRC press, 2018).
- [60] H. Krawczyk, Lfsr-based hashing and authentication, in *Annual International Cryptology Conference* (1994) pp. 129–139.
- [61] J. L. Carter and M. N. Wegman, Universal classes of hash functions, in *Proceedings of the ninth annual ACM symposium on Theory of computing* (1977) pp. 106–112.
- [62] X.-B. Wang, Beating the photon-number-splitting attack in practical quantum cryptography, *Phys. Rev. Lett.* **94**, 230503 (2005).
- [63] H.-K. Lo, X. Ma, and K. Chen, Decoy state quantum key distribution, *Phys. Rev. Lett.* **94**, 230504 (2005).
- [64] G. Brassard and L. Salvail, Secret-key reconciliation by public discussion, in *Workshop on the Theory and Application of Cryptographic Techniques* (Springer, 1993) pp. 410–423.
- [65] H. Yan, T. Ren, X. Peng, X. Lin, W. Jiang, T. Liu, and H. Guo, Information reconciliation protocol in quantum key distribution system, in *2008 Fourth International Conference on Natural Computation*, Vol. 3 (IEEE, 2008) pp. 637–641.
- [66] V. Shoup, On fast and provably secure message authentication based on universal hashing, in *Annual International Cryptology Conference* (Springer, 1996) pp. 313–328.
- [67] R. König, R. Renner, and C. Schaffner, The operational meaning of min-and max-entropy, *IEEE trans. Inform. Theory* **55**, 4337 (2009).
- [68] M. Hayashi, Exponential decreasing rate of leaked information in universal random privacy amplification, *IEEE trans. Inform. Theory* **57**, 3989 (2011).
- [69] J. Massey, Shift-register synthesis and bch decoding, *IEEE trans. Inform. Theory* **15**, 122 (1969).
- [70] H.-L. Yin, M.-G. Zhou, J. Gu, Y.-M. Xie, Y.-S. Lu, and Z.-B. Chen, Tight security bounds for decoy-state quantum key distribution, *Sci. Rep.* **10**, 14312 (2020).
- [71] C. C. W. Lim, M. Curty, N. Walenta, F. Xu, and H. Zbinden, Concise security bounds for practical decoy-state quantum key distribution, *Phys. Rev. A* **89**, 022307 (2014).
- [72] 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. Applied* **12**, 024061 (2019).
- [73] B.-Y. Tang, B. Liu, Y.-P. Zhai, C.-Q. Wu, and W.-R. Yu, High-speed and large-scale privacy amplification scheme for quantum key distribution, *Sci. Rep.* **9**, 15733 (2019).