

Information-theoretically secure quantum timestamping with one-time universal hashing

Ming-Yang Li,^{1,*} Chen-Xun Weng,^{1,*} Wen-Bo Liu,^{2,*} Mengya Zhu,³ and Zeng-Bing Chen^{1,3,†}

¹*National Laboratory of Solid State Microstructures and School of Physics,*

Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China

²*College of Information and Communication, National University of Defense Technology, Wuhan 430010, China*

³*MatricTime Digital Technology Co. Ltd., Nanjing 211899, China*

(Dated: June 13, 2025)

Accurate and tamper-resistant timestamps are essential for applications demanding verifiable chronological ordering, such as legal documentation and digital intellectual property protection. Classical timestamp protocols rely on computational assumptions for security, rendering them vulnerable to quantum attacks, which is a critical limitation given the rapid progress in quantum computing. To address this, we propose an information-theoretically secure quantum timestamping protocol based on one-time universal hashing with quantum keys. Our protocol simultaneously achieves information-theoretic security and high efficiency, enabling secure timestamping for arbitrarily long documents. Simulations demonstrate a generation rate exceeding 100 timestamps per second over intercity distances. In addition, our protocol only requires weak coherent states and tolerates imperfect quantum keys, making it practical for large-scale deployment. This work advances the field of quantum timestamping and contributes to the broader development of quantum cryptography and the future quantum internet.

I. INTRODUCTION

Accurate and reliable timestamps are crucial for determining the chronological order of events in various applications. They provide legally and technically precise time values, ensuring that issuance, query, and verification processes comply with the non-repudiation and unforgeability requirements of network applications. Secure and reliable timestamp protocols are crucial for legal documentation, digital intellectual property protection, and data integrity verification. For example, in electronic financial systems, such as electronic banking or Alipay, a precise completion time of the ownership transferring for a significant asset is legally crucial, which consists of a reliable proof of when the buyer's ownership of the asset starts. To support these requirements, various classical timestamp protocols utilizing hash algorithms and digital signatures have been proposed and implemented in practice [1–5]. However, these existing classical protocols face a fundamental security challenge: the vulnerability of classical digital signature schemes and computational assumptions to quantum attacks given the ongoing advancements in quantum computing and algorithms [6–12]. This weakness creates opportunities for timestamp manipulation of malicious attackers during issuance and transmission, compromising the reliability of the received timestamp and undermining its credibility.

On the other hand, based on the quantum no-cloning theorem, quantum cryptographic protocols offering information-theoretic security and superior performance have been developed, such as quantum key distri-

bution (QKD) [13–20], quantum secure direct communication [21–23], quantum digital signatures (QDS) [24–30], quantum Byzantine agreement [31–35], quantum e-commerce [36] and quantum digital payments [37–42]. However, to date, no existing information-theoretically secure quantum timestamp protocols can guarantee transferability, unforgeability, and non-repudiation of timestamps against quantum attackers without relying on computational assumptions. Developing such a quantum solution to protect timestamps against attacks from malicious quantum attackers is crucial for secure timestamping in the quantum era.

Here, for the first time, we present a quantum timestamp protocol offering information-theoretic security against quantum attacks and high efficiency for documents of arbitrary length. By using the one-time universal hashing (OTUH) introduced in Ref. [29] (see Appendix A), the protocol employs different quantum keys for each hash computation, making it infeasible for the hash values to be attacked. Our protocol employs quantum key generation (QKG) to produce correlated quantum secret keys for one-time universal hash function construction and one-time pad encryption, ensuring security against quantum attacks without relying on computational assumptions. Moreover, we prove the composable security of our protocol, demonstrating the resilience against both internal and external threats. The security of the timestamps is ensured by the non-repudiation and unforgeability properties derived from the inherent asymmetric correlations of quantum keys in our protocol [29, 30]. Numerical simulations demonstrate that our protocol can achieve a rate exceeding 100 secure timestamps per second over intercity distances. The protocol requires only the preparation of weak coherent quantum states, not entangled states or high-dimensional qudits, and is compatible with QKG stages adapted from various

* These authors contributed equally to this work.

† zbchen@nju.edu.cn

QKD protocols, simplifying its large-scale experimental deployment.

The remaining part of this paper is organized as follows. Section II describes the scheme of our timestamp protocol. Section III presents details of security analysis. Numerical simulations are presented in Sec. IV to show the efficiency of our scheme. Finally, we conclude in Sec. V.

II. PROTOCOL DESCRIPTION

In our quantum timestamp scheme, there are four legal parties: a subscriber, a verifier, a trusted certificate authority (CA), and a timestamp authority (TSA). Our scheme considers the scenario where the subscriber possesses a document, *doc*, and seeks a timestamp to prove its existence prior to a specific point in time. The verifier checks the validity of the file and its timestamp. CA records the file to prevent unauthorized alteration by the subscriber. Finally, TSA provides the timestamp and its associated signature. Through our protocol, a reliable timestamp can be created for a valid file. After successful verification, the file and its timestamp can not be tampered with or denied.

The whole protocol includes two stages. First, the related participants share correlated quantum keys through a QKG stage. Then, a timestamp is generated for a certified file; the file's integrity and the timestamp's validity are verified using three-party verification with the messaging stage of the OTUH-QDS [29, 30].

A. Quantum key generation stage

The QKG stage is employed to generate correlated quantum keys for three participants, and any QKD and quantum secret sharing [43–52] protocol can be adapted for this QKG stage [29, 30]. We illustrate this stage using the two-photon twin-field (TPTF)-QKG as an example, adapted from the two-party TPTF-QKD protocol in Ref. [53]. Simulation results for sending-or-not-sending (SNS)-QKG (based on SNS-QKD in Ref. [54]) and BB84-QKG (based on BB84-QKD in Ref. [55]) are also provided. Detailed calculations for these three QKG methods are given in Appendix D.

The steps for TPTF-QKG are detailed below. The three participants are named U_1 , U_2 and U_3 respectively.

(a) Preparation. For each of the N_{tot} time bins, U_1 and U_2 independently generate a weak coherent pulse. These pulses are encoded with randomly selected phases ($\theta_x^l \in [0, 2\pi)$) and bits ($r_x^l \in \{0, 1\}$), where $x \in \{1, 2\}$ indexes the users and $l \in \{1, 2, \dots, N_{\text{tot}}\}$ denotes the sequence of time bin. The pulses are described by the state $|\psi_x^l\rangle = |e^{i(\theta_x^l + r_x^l \pi)} \sqrt{k_x^l}\rangle$, with intensity k_x^l drawn probabilistically (p_{k_x}) from the set $\{\mu_x, \nu_x, \mathbf{o}_x, \hat{\mathbf{o}}_x\}$. This set represents signal, decoy, and two types of vacuum states (preserved and declared), satisfying $\mu_x > \nu_x > \mathbf{o}_x =$

$\hat{\mathbf{o}}_x = 0$. Following encoding, the pulses are transmitted to an untrusted relay (UR) over insecure channels. Concurrently, a bright reference pulse is also sent to the UR to enable measurement of the phase noise difference ϕ_{12}^l . This establishes the initial state for the subsequent stages of the protocol.

(b) Measurement. For UR, interference measurements are performed on each arriving pulse pair. A beam splitter and two detectors are employed; a successful detection event is defined as only one detector registering a click. The UR publicly announces each successful detection event, identifying the specific detector that registered the click. For clarity, we use a notation that includes the intensity selections of both users. For example, $\{\mu_1, \nu_2\}$ indicates that user U_1 selected the signal intensity (μ_1), and user U_2 chose a decoy intensity (ν_2).

(c) Sifting. Successful detection events are categorized into two distinct sets. The first set comprises events where neither user employs decoy or declared-vacuum intensities; these are $\{\mu_1, \mathbf{o}_2\}$, $\{\mu_1, \mu_2\}$, $\{\mathbf{o}_1, \mu_2\}$, and $\{\mathbf{o}_1, \mathbf{o}_2\}$, and are used to generate the Z-basis key. The second set encompasses all remaining successful events and is dedicated to parameter estimation for the X-basis. For Z-basis key generation, U_1 randomly pairs a time bin (l_1) with intensity μ_1 and another time bin (l_2) with intensity \mathbf{o}_1 . A bit value of 0 (1) is assigned if $l_1 < l_2$ ($l_1 > l_2$), with l_1 and l_2 communicated to U_2 . If U_2 's intensities are $k_2^{\min(l_1, l_2)} = \mu_2$ (\mathbf{o}_2) and $k_2^{\max(l_1, l_2)} = \mathbf{o}_2$ (μ_2), U_2 assigns a bit value of 0 (1). Events with $k_2^{l_1} = k_2^{l_2} = \mathbf{o}_2$ or μ_2 are discarded. The resulting Z-basis events are $\{\mu_1 \mathbf{o}_1, \mathbf{o}_2 \mu_2\}$, $\{\mu_1 \mathbf{o}_1, \mu_2 \mathbf{o}_2\}$, $\{\mathbf{o}_1 \mu_1, \mathbf{o}_2 \mu_2\}$, and $\{\mathbf{o}_1 \mu_1, \mu_2 \mathbf{o}_2\}$. X-basis parameter estimation involves exchanging intensity and phase information via an authenticated channel. The global phase difference at time bin l is $\theta^l = \theta_1^l - \theta_2^l + \phi_{12}^l$. Events $\{\nu_1^l, \nu_2^l\}$ are retained only if $\theta^l \in [-\delta, \delta] \cup [\pi - \delta, \pi + \delta]$, where δ defines the phase slice. Pairs of retained events with $|\theta^{l_1} - \theta^{l_2}| = 0$ or π are formed: $\{\nu_1^{l_1} \nu_1^{l_2}, \nu_2^{l_1} \nu_2^{l_2}\}$. X-basis bit values are derived from $r_1^{l_1} \oplus r_1^{l_2}$ and $r_2^{l_1} \oplus r_2^{l_2}$. U_2 always flips its Z-basis bit. In the X-basis, U_2 conditionally flips bits to ensure correlation with U_1 : If the global phase difference is 0 (π) and the UR detectors are different (same), U_2 flips; otherwise, it keeps its bit. The remaining events are used for decoy-state analysis.

(d) Parameter estimation. From the randomly selected Z-basis bits, U_1 and U_2 generate an n^Z -bit raw key. The remaining Z-basis bits are used to estimate the bit error rate, E^Z . Following this, all X-basis bit values are exchanged to count the total number of errors. Employing a decoy-state method [56, 57], the number of vacuum events ($s_{0\mu_2}^Z$), single-photon pairs (s_{11}^Z), and the single-photon pair phase error rate (ϕ_{11}^Z) in the Z-basis are estimated.

(e) Post-processing. The key post-processing involves two steps.

1. *Error Correction and Verification:* An error correction algorithm with ε_{cor} -correctness [58, 59] is

applied by U_1 and U_2 to distill the final key. The resulting key remains n^Z bits long. To further enhance security, U_1 permutes the key and transmits this permutation to U_2 securely.

2. *Privacy Amplification:* A randomly chosen universal₂ hash function is applied by U_1 to the n^Z -bit error-corrected key, producing an $n^{Z'}$ -bit final key. This function is then communicated to U_2 , who applies the same function to their n^Z -bit key to obtain the final $n^{Z'}$ -bit key. This step significantly reduces the information potentially accessible to an eavesdropper.

(f) **Correlating.** U_1 and U_3 conduct steps (a)–(e) again to generate another quantum key with equal length. Then, U_1 performs a bitwise XOR operation on the two resulting keys.

For the timestamp stage, the following key generations are required: (1) The subscriber and verifier share a $3n$ -bit key, divided into an n -bit segment (s_1) and a $2n$ -bit segment (r_1). The subscriber and CA also share a $3n$ -bit key, partitioned into an n -bit segment (s_2) and a $2n$ -bit segment (r_2). The subscriber then calculates $s = s_1 \oplus s_2$ and $r = r_1 \oplus r_2$ to form related quantum keys, where \oplus denotes the bit-wise XOR of two strings. (2) TSA and subscriber share a $3m$ -bit key, divided into an m -bit segment (u_1) and a $2m$ -bit segment (v_1). TSA and verifier also share a $3m$ -bit key, partitioned into an m -bit segment (u_2) and a $2m$ -bit segment (v_2). TSA then calculate $u = u_1 \oplus u_2$ and $v = v_1 \oplus v_2$ to form related quantum keys.

B. Timestamp stage

The timestamp stage includes three steps. In the first step, The subscriber, the verifier and CA check the correctness of the digest of doc with CA recording the file. In the second step, TSA produces the timestamp file according to the digest and the time. TSA, the subscriber, and the verifier then verify the validity of the timestamp in the third step. The details of our proposed timestamp stage are presented below and illustrated in Figure 1. The length of the file doc is denoted as L .

Step 1. File verification.

(a) The subscriber generates a binary random bit string str_1 with length n with the local quantum random number generator (QRNG) [60–64]. str_1 is used to generate an irreducible polynomial $p(x)$ of order n . The specific method of generating $p(x)$ is provided in Appendix B.

(b) The subscriber generates the LFSR-based Toeplitz hash function $h_{p,s}$ by $p(x)$ and s as input random number, as shown in Appendix C. Then the hash function $h_{p,s}$ is used to hash the file doc to be timestamped, and the digest $h_{p,s}(doc)$ is obtained. The subscriber encrypts $h_{p,s}(doc)$ and str_1 with the shared key r to obtain $r \oplus \{h_{p,s}(doc) \mid str_1\}$ and sends it to the verifier. The file

doc is also sent to the verifier without encryption. The verifier then forwards doc and $r \oplus \{h_{p,s}(doc) \mid str_1\}$ to CA.

(c) The verifier and CA exchange their quantum keys s_1, s_2, r_1, r_2 with each other via classical authenticated channels. Each of them can combine s_1 and s_2 to get s' , r_1 and r_2 to get r' . Subsequently, the verifier and the CA utilize r' to decrypt the ciphertext, thereby retrieving $h_{p,s}(doc)$ and str_1 . The polynomial $p'(x)$ is generated using str_1 and a hash function $h'_{p,s}$ is obtained using $p'(x)$ and s' . Apply $h'_{p,s}$ to doc and verify the equivalence of the hash value $h'_{p,s}(doc)$ with $h_{p,s}(doc)$. If they are the same, the authentication succeeds. If they are different, the authentication fails.

(d) The file is accepted only when both the verifier and CA successfully verify it.

Step 2. Timestamp production.

(a) After successful authentication, the subscriber sends a timestamp request to TSA and attaches the digest $h_{p,s}(doc)$ verified by the verifier.

(b) TSA gets the time value T at the requested time from the National Time Centre.

(c) Given that there may be more than one subscriber request timestamp within a standard time precision, multiple requests are numbered by a Request identifier: Req-ID. The assignment of Req-ID is random, not sequential.

(d) With the organization identifier Auth-ID of TSA and organization identifier Ver-ID of the verifier, TSA produces the timestamp document $TS(doc)$:

$$TS(doc) = [h_{p,s}(doc), T, \text{Req-ID}, \text{Auth-ID}, \text{Ver-ID}], \quad (1)$$

whose length is denoted as k .

The inclusion of Auth-ID and Ver-ID simulates a practical, multi-authority timestamp system. This reflects real-world scenarios where multiple authorities and verifiers might provide timestamp services, necessitating verification of each party's organisation information and legitimacy to prevent malicious actors from generating fraudulent timestamps.

Step 3. Timestamp verification.

(a) TSA generates a random bit string str_2 with length m using local QNRG, and an irreducible polynomial $q(x)$ using str_2 . The hash function $h_{q,u}$ is obtained by $u = u_1 \oplus u_2$ and $q(x)$. $TS(doc)$ is hashed by $h_{q,u}$ to get the hash value $h_{q,u}(TS(doc))$. TSA encrypts $h_{q,u}(TS(doc))$ and str_2 with the key $v = v_1 \oplus v_2$ and sends the ciphertext $v \oplus \{h_{q,u}(TS(doc)) \mid str_2\}$ and the timestamp document $TS(doc)$ to the subscriber. Then the subscriber forwards the ciphertext and the timestamp document to the verifier.

(b) After they both receive the ciphertext and the timestamp document, the subscriber and the verifier exchange their keys u_1, u_2, v_1, v_2 through authenticated classical channels. Each of them can combine u_1 and u_2 to get u' , v_1 and v_2 to get v' . The subscriber and the verifier then decrypt the ciphertext by v' to obtain

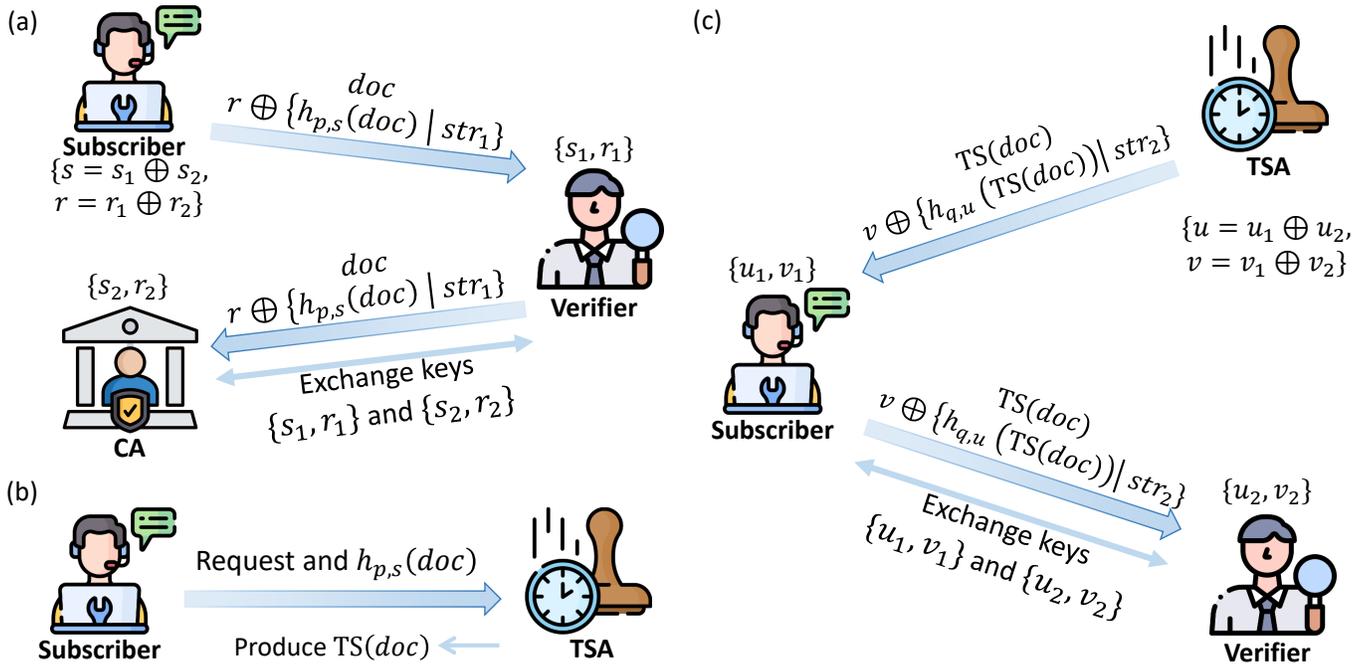


FIG. 1. Illustration of our quantum timestamp scheme. (a) File verification: The subscriber generates a hash digest of the document, encrypts it along with the irreducible polynomial coefficients, and sends it to the verifier for authentication. The verifier forwards the document and its corresponding encrypted hash value to CA and exchanges quantum keys with CA; both of them independently verify the validity of the file and its corresponding hash value according to the exchanged quantum keys. (b) Timestamp generation: Upon successful verification, TSA generates a timestamp document incorporating the verified digest, timestamp, request ID, and the identities of the authority and verifier. (c) Timestamp verification: TSA sends the timestamp and its encrypted hash to the subscriber, who forwards it to the verifier and exchanges keys; both independently verify the hash, ensuring timestamp validity.

$h_{q,u}(TS(doc))$ and str_2 . The polynomial $q'(x)$ is generated using str_2 and the hash function $h'_{q,u}$ is obtained using $q'(x)$ and u' . Apply $h'_{q,u}$ to $TS(doc)$ and verify the equivalence of the hash value $h'_{q,u}(TS(doc))$ with $h_{q,u}(TS(doc))$. If they are the same, the authentication succeeds. If they are different, the authentication fails.

(c) The timestamp is accepted only when both the subscriber and the verifier successfully verify it.

Note that the communication between the verifier and CA during file verification, and between the subscriber and verifier during timestamp verification, should utilize classical authenticated channels.

III. SECURITY ANALYSIS

The security of our protocol rests on the secrecy and asymmetric relationship of correlated quantum keys, with the resulting non-repudiation and unforgeability properties of the file and timestamp. We analyze the security of each step of the scheme, considering both the internal dishonest party and the external adversary. As at most three parties are involved in each verification step, only one internal dishonest party can be allowed because two dishonest parties can perfectly collude to defraud the third party [28, 34]. We present the security

analysis with perfect keys here. Our protocol can also employ imperfect keys obtained from a QKG without a privacy amplification step. The condition with imperfect keys is almost the same, except for minor differences in the specific security parameter for unforgeability. We note that in the protocol without privacy amplification, the same security as that using privacy amplification can be achieved by simply increasing the number of bits consumed, so security is not affected [30].

A. Security of file verification

File verification involves a three-party certification process using the messaging stage of the OTUH-QDS [29, 30], which needs to consider the denial by the subscriber and tampering by the verifier.

A dishonest subscriber may deny the doc that both the verifier and CA have accepted, and claim that the timestamp created in the next stage is for another file. However, successful denial requires errors during the key exchange between the verifier and CA, as both the verifier and CA with correlated quantum keys must independently compute identical hash values of the file to

validate it. The probability of successful denial is

$$\varepsilon_{\text{FV}}^{\text{den}} = 2\varepsilon_{\text{CC}}, \quad (2)$$

where ε_{CC} represents the failure probability of the classical communication, which is negligible (typically 10^{-11}) and usually ignored in comprehensive communication systems [29]. Therefore, the non-repudiation probability makes our protocol secure under the denial attack from a dishonest subscriber.

On the other hand, a dishonest verifier aims to intercept the transmitted data via classical channel and alter the transmitted information, str_1 , the hash value $h_{p,s}(doc)$, and the file doc , into a new, self-consistent set: str'_1 , $h_{p',s}(doc')$, and doc' . This is prevented by the unforgeability property, and Ref. [29] has proven that an optimal attack strategy requires knowledge of str_1 . This necessitates guessing half of the key of r (with length n), used to encrypt str_1 . The probability of successful guessing is $P_{guess} = 2^{-n}$ for perfectly secret keys. The probability of a successful tampering attack on file verification is

$$\varepsilon_{\text{FV}}^{\text{tam}} = L \cdot 2^{1-n}. \quad (3)$$

Therefore, the probability of successful file verification, P_{FV} , is given by

$$P_{\text{FV}} = (1 - \varepsilon_{\text{FV}}^{\text{den}}) (1 - \varepsilon_{\text{FV}}^{\text{tam}}) \geq 1 - (\varepsilon_{\text{FV}}^{\text{den}} + \varepsilon_{\text{FV}}^{\text{tam}}). \quad (4)$$

where the scaling is tight enough because both $\varepsilon_{\text{FV}}^{\text{den}}$ and $\varepsilon_{\text{FV}}^{\text{tam}}$ are exceedingly small. The overall failure probability for the file verification step, ε_{FV} , can therefore be approximated as the sum of the individual failure probabilities: $\varepsilon_{\text{FV}} = \varepsilon_{\text{FV}}^{\text{den}} + \varepsilon_{\text{FV}}^{\text{tam}}$.

B. Security of timestamp verification

Timestamp verification involves a three-party certification process, which needs to consider the denial by TSA and tampering by any of the other two participants (the subscriber or the verifier) and an external adversary.

A dishonest TSA might falsely deny having issued a valid timestamp, even if both the subscriber and verifier have independently verified its authenticity. Similar to file verification, successful denial requires errors during the key exchange between the subscriber and the verifier, whose probability is

$$\varepsilon_{\text{TV}}^{\text{den}} = 2\varepsilon_{\text{CC}}. \quad (5)$$

A dishonest subscriber, verifier or external adversary may try to tamper with the transmitted information to cheat. The scenario is also the same as the analysis of the security of file verification, which is prevented by the unforgeability property. Since the attacker also needs

to know str_2 for the implementation of optimal cheating methods, we can denote the probability of tampering as

$$\varepsilon_{\text{TV}}^{\text{tam}} = k \cdot 2^{1-m}. \quad (6)$$

Successful timestamp verification demands that both malicious denial and tampering fail. The probability of successful file verification is

$$P_{\text{TV}} = (1 - \varepsilon_{\text{TV}}^{\text{den}}) (1 - \varepsilon_{\text{TV}}^{\text{tam}}) \geq 1 - (\varepsilon_{\text{TV}}^{\text{den}} + \varepsilon_{\text{TV}}^{\text{tam}}), \quad (7)$$

and the overall failure probability for the timestamp verification is $\varepsilon_{\text{TV}} = \varepsilon_{\text{TV}}^{\text{den}} + \varepsilon_{\text{TV}}^{\text{tam}}$.

C. Composable security

Our quantum timestamp protocol's security depends not only on the success of the file and timestamp verification steps but also on the underlying QKG stage. All components must function correctly to guarantee secure timestamps. The QKG stage involves estimating parameters such as bit and phase error rates, subject to statistical fluctuations [15–18], which are used for subsequent error correction and privacy amplification. Although our scheme can employ imperfect keys without privacy amplification, the error correction step is still needed, and these parameters should also be estimated. The security bound of the QKG stage, denoted as ε_{QKG} , must account for both perfect and imperfect key scenarios. Therefore, the overall security bound ε_{sec} for our quantum timestamp scheme considering QKG stage [65] is

$$\varepsilon_{\text{sec}} = \varepsilon_{\text{QKG}} + \varepsilon_{\text{FV}} + \varepsilon_{\text{TV}}. \quad (8)$$

IV. NUMERICAL SIMULATION

This section presents numerical simulations of the proposed quantum timestamp scheme, providing a quantitative assessment of its efficiency. A security threshold is established by setting $\varepsilon_{\text{QKG}} = \varepsilon_{\text{FV}} = \varepsilon_{\text{TV}} = 10^{-10}$. The simulations incorporate realistic experimental parameters: detector efficiency $\eta_{de} = 70\%$, dark count rate $p_d = 10^{-8}$, misalignment error rate $e_t = 2\%$, and error correction efficiency $f_{\text{cor}} = 1.1$. For simplicity, the channel loss and distance between each pair of the four parties are assumed to be equal. To reflect practical scenarios, the simulations model the timestamp of a 100 Mb file (doc) with a 1 Kb timestamp. The total number of pulses is set to 10^{14} to reflect the finite analysis. We quantify the efficiency of a timestamp scheme by the quantum timestamp rate (QTSR), defined as the number of secure quantum timestamps generated per second. This can be calculated by

$$\text{QTSR} = \frac{\min(\text{KGR})}{l_{\text{key}}} = \frac{\min(\text{KGR})}{6(n+m)}, \quad (9)$$

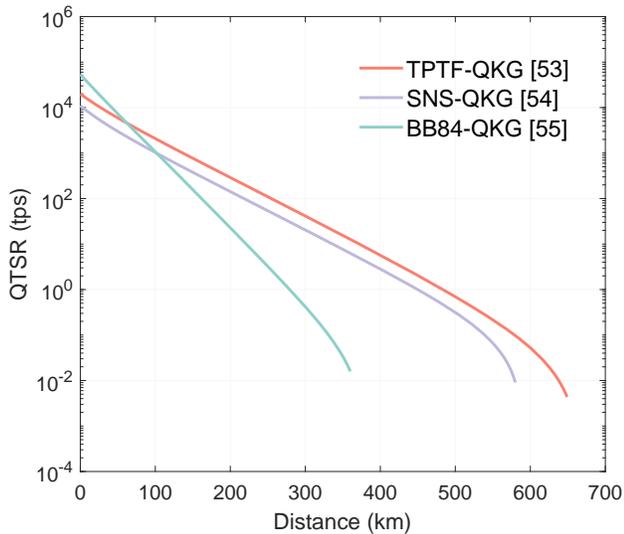


FIG. 2. The efficiency of the proposed timestamp scheme. We use QTSR to denote the number of timestamps generated per second. The results of TPTF-QKG [53], SNS-QKG [54] and BB84-QKG [55] are presented. Our protocol enables more than 10^2 - 10^3 times of timestamping per second at intercity distances (100-300 km), and is available over 600 km using TPTF-QKG.

where KGR stands for the key generation rate and $l_{\text{key}} = 6(n + m)$ is the length of quantum keys needed to complete a timestamp stage, including two $3n$ -bit (s_1, r_1 and s_2, r_2) and two $3m$ -bit (u_1, v_1 and u_2, v_2) quantum keys. While the distances between each party are set to be the same, to account for channel variations among parties-resulting in differing channel losses and KGR, a conservative lower bound on QTSR is established by using the minimum KGR observed across all channels.

Assuming a laser repetition rate of 1 GHz and a fiber-loss coefficient of $\alpha = 0.165$ dB/km, Figure 2 depicts the calculated QTSR for transmission distances ranging from 0 to over 600 km (where key generation is feasible). The results demonstrate a QTSR exceeding 100 timestamps per second for inter-city distances (approximately 100 to 300 km). Specifically, when employing TPTF-QKG, 10^4 times of successful stamping can be achieved, and our protocol is still available over 600 km spatial distance. Our simulations show the practicality and high efficiency of our protocol, building a theoretical foundation for the experimental implementation of quantum timestamps.

V. DISCUSSION AND CONCLUSIONS

Secure timestamp protocols are essential for applications demanding verifiable chronological ordering of events. Classical schemes, however, are vulnerable to quantum attacks. This work proposes a quantum timestamp protocol offering information-theoretic security against such threats. The protocol employs one-

time universal₂ hashing combined with one-time pad encryption using quantum keys, and can also time-stamp a document of arbitrary length with high efficiency. Our simulations, which quantify the timestamp generation rate, highlight the efficiency of our work, particularly over intercity distances. The practicality of our protocol is further enhanced by its simple experimental requirements. It utilizes only coherent quantum states, avoiding the complexities of high-dimensional qudits or entangled states, and is compatible with various QKG methods.

Leveraging these advantages, our information-theoretically secure quantum timestamping protocol represents a significant step forward for applications that require high security and integrity of time records against quantum attacks, such as digital financial systems, distributed databases, and military command infrastructures. Its practical design enables seamless integration into existing quantum networks, meeting the demands for secure timestamps in the quantum era.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 12274223), the Program for Innovative Talents, Entrepreneurs in Jiangsu (No. JSSCRC2021484) and the Key Research and Development Program of Nanjing Jiangbei New Area (No. ZDYD20210101). We gratefully acknowledged the free icons designed by Flaticon from www.flaticon.com.

Appendix A: One-time universal₂ hash function

A family of hash functions, \mathcal{H} , mapping a set \mathbb{P} to a set \mathbb{Q} , is defined as universal₂ [66] if, for any distinct $\mathcal{M}, \mathcal{M}' \in \mathbb{P}$:

$$\Pr_{h \in \mathcal{H}}[h(\mathcal{M}) = h(\mathcal{M}')] \leq \frac{1}{|\mathbb{Q}|} \quad (\text{A1})$$

where $|\mathbb{Q}|$ denoting the number of elements in set \mathbb{Q} . Universal₂ hash functions efficiently map arbitrarily long inputs to short hash values with low collision probability. Traditional random matrices form a universal₂ family, requiring $i \cdot j$ random bits to map a j -bit input to an i -bit hash. To reduce this cost, Toeplitz matrices [67] are used instead, requiring only $i + j - 1$ random bits and exhibiting a collision probability of 2^{-i} . Moreover, the LFSR-based Toeplitz matrices [68] offer another alternative, acting as an almost universal₂ hash function. These matrices are defined by an irreducible polynomial of degree i over $\text{GF}(2)$ and an i -bit random initial vector, and their collision probability is $j/2^{i-1}$.

Analogous to one-time pad encryption, where a shared key is used only once to encrypt a message, the one-time universal₂ hash function proposed in Ref. [29] employs a randomly chosen initial vector and irreducible polynomial for each hash computation. This approach enhances security by making each hash computationally infeasible to predict or reverse engineer. Participating parties share the keys used to generate the universal₂ hash function, with these keys discarded after a single use within the protocol.

Appendix B: Generation of an irreducible polynomial

The generation of irreducible polynomials is a critical step in various cryptographic applications, particularly those utilizing LFSR-based Toeplitz matrices for authentication. An irreducible polynomial of degree i over the $\text{GF}(2)$ is a polynomial that cannot be factored into polynomials of lower degree over the same field. Such polynomials are represented in the standard form: $p(x) = x^i + a_{i-1}x^{i-1} + \dots + a_1x + a_0$, where each coefficient is binary. This polynomial can be concisely represented by an i -bit string, $str = (a_{i-1}, a_{i-2}, \dots, a_1, a_0)$, where each bit corresponds to a coefficient. Since a_0 is always 1 for irreducible polynomials, only $i - 1$ random bits are actually needed, effectively using the string $str = (a_{i-1}, \dots, a_1, 1)$. A QRNG is employed to efficiently generate these $i - 1$ random bits, forming the basis for constructing candidate irreducible polynomials. Subsequently, an irreducibility test is performed on each generated candidate polynomial. A polynomial $p(x)$ of degree i over $\text{GF}(2)$ is irreducible if and only if it satisfies the following two conditions:

$$\begin{cases} \text{(A)} & x^{2^i} \equiv x \pmod{p(x)} \\ \text{(B)} & \gcd(x^{2^{i/d}} - x, p(x)) = 1 \end{cases} \quad (\text{B1})$$

where d is any prime factor of i and $\gcd(a(x), b(x))$ means the greatest common divisor (GCD) of two polynomials $a(x)$ and $b(x)$. The efficient computation of both conditions (A) and (B) leverages optimized algorithms such as fast modular exponentiation (for condition A) and the extended Euclidean algorithm (for condition B) [69]. If a generated polynomial does not satisfy both conditions, a new polynomial is generated and tested until an irreducible polynomial is found.

However, this method may be computationally expensive and require many random numbers. A more efficient method for generating irreducible polynomials, proposed in Ref. [70], can be adopted, assuming participants have a pre-stored irreducible polynomial $p_1(x)$ of order i . This method generates a new irreducible polynomial using only one i -bit random string. This string is used as the coefficients of a random polynomial $p_2(x)$. Then, $2i$ values are computed: $e_0 = f_0(0)$, $e_1 = f_1(0)$, ..., $e_{2n-1} = f_{2i-1}(0)$, where $f_j(x) = p_2^j(x) \pmod{p_1(x)}$ and $j \in \{0, 1, \dots, 2i - 1\}$. These values can be efficiently computed using the Berlekamp-Massey algorithm [71] to determine the i -order minimal polynomial of $p_2(x)$, which is irreducible.

Appendix C: Construction of the LFSR-Based Toeplitz Hash matrix

The LFSR-based Toeplitz hash function, a crucial component of our authentication mechanism, leverages the properties of LFSR-based Toeplitz matrices to efficiently map a j -bit message, $M = (M_0, M_1, \dots, M_{j-1})^T$ (the superscript means the transpose of the vector), to an i -bit hash value. The $i \times j$ Toeplitz matrix, $H_{i \times j}$, is dynamically generated based on the i -bit coefficient string str of an irreducible polynomial $p(x)$, and an i -bit binary string $K = (k_{i-1}, \dots, k_1, k_0)$ obtained by QKG. The function operates as

$$h_{p,K}(M) = H_{i \times j} \cdot M \pmod{2}, \quad (\text{C1})$$

where \cdot denotes the matrix multiplication and $\pmod{2}$ means every element of the result is calculated modulo 2.

Now we describe the procedure to construct such a hash function. The LFSR is initialized to the state vector S_1 , which is set to be $(k_{i-1}, \dots, k_1, k_0)^T$. Then, the LFSR generates a sequence of i -bit vectors with S_1 being the first by shifting down every element of the previous vector and adding a new element at the top. More specif-

ically, the LFSR generates the following vectors:

$$\begin{aligned} S_2 &= (str \cdot S_1, k_{i-1}, \dots, k_1)^T \pmod{2}, \\ S_3 &= (str \cdot S_2, str \cdot S_1, k_{i-1}, \dots, k_2)^T \pmod{2}, \\ &\dots, \\ S_j &= (str \cdot S_{j-1}, str \cdot S_{j-2}, \dots, str \cdot S_{j-i})^T \pmod{2}. \end{aligned} \quad (C2)$$

Finally, the LFSR-based Toeplitz hash matrix is constructed as

$$H_{i \times j} = (S_1, S_2, \dots, S_j). \quad (C3)$$

The use of an irreducible polynomial and the inherent randomness from the QRNG ensure the almost universal₂ property of this hash function. The specific collision probability is $j \cdot 2^{1-i}$ [68]. This construction ensures efficient computation while maintaining a low collision probability, making it suitable for cryptographic applications.

Appendix D: Simulation details

The following sections detail the numerical simulation calculations for TPTF-QKG, BB84-QKG, and SNS-QKG, both with and without privacy amplification. For clarity, the calculations are presented for the generation of a single n -bit string. The simulation of generating m -bit string is simply to substitute the corresponding parameters.

1. TPTF-QKG

The TPTF-QKG calculation follows the TPTF-QKD method of Ref. [53]. For intensities (k_1, k_2) and phase difference θ , the left (L) and right (R) single-detector click gains are:

$$\begin{aligned} Q_{k_1 k_2}^{L\theta} &= y_{k_1 k_2} (e^{\omega_{k_1 k_2} \cos \theta} - y_{k_1 k_2}), \\ Q_{k_1 k_2}^{R\theta} &= y_{k_1 k_2} (e^{-\omega_{k_1 k_2} \cos \theta} - y_{k_1 k_2}), \end{aligned} \quad (D1)$$

where $y_{k_1 k_2} = e^{-(\eta_1 k_1 + \eta_2 k_2)/2} (1 - p_d)$, $\omega_{k_1 k_2} = \sqrt{\eta_1 k_1 \eta_2 k_2}$, $\eta_i (i \in \{1, 2\})$ are the channel transmittances for U_1 and U_2 , and p_d is the dark count probability. The overall gain is

$$\begin{aligned} Q_{k_1 k_2} &= \frac{1}{2\pi} \int_0^{2\pi} (Q_{k_1 k_2}^{L\theta} + Q_{k_1 k_2}^{R\theta}) d\theta \\ &= 2y_{k_1 k_2} [I_0(\omega_{k_1 k_2}) - y_{k_1 k_2}], \end{aligned} \quad (D2)$$

where $I_0(x)$ is the zeroth-order modified Bessel function of the first kind. The total event count for $\{k_1, k_2\}$ is $x_{k_1 k_2} = N_{\text{tot}} p_{k_1} p_{k_2} Q_{k_1 k_2}$.

We define the correct Z -basis events as $\{\mu_1 \mathbf{o}_1, \mathbf{o}_2 \mu_2\}$, $\{\mathbf{o}_1 \mu_1, \mu_2 \mathbf{o}_2\}$, and incorrect Z -basis events as

$\{\mu_1 \mathbf{o}_1, \mu_2 \mathbf{o}_2\}$, $\{\mathbf{o}_1 \mu_1, \mathbf{o}_2 \mu_2\}$. The Z -basis correct (n_C^Z) and incorrect (n_E^Z) event counts are calculated as

$$n_C^Z = x_{\min} \frac{x_{\mathbf{o}_1 \mu_2} x_{\mu_1 \mathbf{o}_2}}{x_0 x_1} = \frac{x_{\mathbf{o}_1 \mu_2} x_{\mu_1 \mathbf{o}_2}}{x_{\max}}, \quad (D3)$$

$$n_E^Z = x_{\min} \frac{x_{\mathbf{o}_1 \mathbf{o}_2} x_{\mu_1 \mu_2}}{x_0 x_1} = \frac{x_{\mathbf{o}_1 \mathbf{o}_2} x_{\mu_1 \mu_2}}{x_{\max}}, \quad (D4)$$

where $x_0 = x_{\mathbf{o}_1 \mu_2} + x_{\mathbf{o}_1 \mathbf{o}_2}$, $x_1 = x_{\mu_1 \mathbf{o}_2} + x_{\mu_1 \mu_2}$, $x_{\min} = \min(x_0, x_1)$, $x_{\max} = \max(x_0, x_1)$. s_{11}^Z means the count of single-photon clicks for the Z -basis. The Z -basis bit error rate is $E^Z = m^Z/n^Z$, where $m^Z = (1 - e_d^Z)n_E^Z + e_d^Z n_C^Z$ denoting the number of Z -basis bit errors, and $n^Z = n_C^Z + n_E^Z$ is the total number of Z -basis events and e_d^Z is the misalignment error.

The effective X -basis event count (n^X) and error count (m^X) are:

$$\begin{aligned} n^X &= \frac{1}{\pi} \int_0^\delta x_{\nu_1 \nu_2}^\theta d\theta = \frac{N_{\text{tot}} p_{\nu_a} p_{\nu_b}}{\pi} \int_\sigma^{\sigma+\delta} y_{\nu_1 \nu_2} \\ &\times (e^{\omega_{\nu_1 \nu_2} \cos \theta} + e^{-\omega_{\nu_1 \nu_2} \cos \theta} - 2y_{\nu_1 \nu_2}) d\theta. \end{aligned} \quad (D5)$$

and

$$\begin{aligned} m^X &= \frac{1}{\pi} \int_\sigma^{\sigma+\delta} x_{\nu_1 \nu_2}^\theta p_E d\theta = \frac{2N_{\text{tot}} p_{\nu_a} p_{\nu_b}}{\pi} \int_\sigma^{\sigma+\delta} y_{\nu_1 \nu_2} \\ &\times \left[\frac{(1 - y_{\nu_1 \nu_2})^2}{e^{\omega_{\nu_1 \nu_2} \cos \theta} + e^{-\omega_{\nu_1 \nu_2} \cos \theta} - 2y_{\nu_1 \nu_2}} - 1 \right] d\theta, \end{aligned} \quad (D6)$$

where $p_E = \frac{2q_{\nu_1 \nu_2}^{L\theta} q_{\nu_1 \nu_2}^{R\theta}}{q_{\nu_1 \nu_2}^\theta q_{\nu_1 \nu_2}^\theta}$.

x^* , \underline{x} , and \bar{x} denote the expected value, lower bound, and upper bound of x , respectively. $x_{k_1 k_2}$ represents the number of $\{k_1, k_2\}$ events. For post-matched events $\{k_1^{l_1} k_1^{l_2}, k_2^{l_1} k_2^{l_2}\}$, $n_{k_1^{l_1} k_1^{l_2}, k_2^{l_1} k_2^{l_2}}$ and $m_{k_1^{l_1} k_1^{l_2}, k_2^{l_1} k_2^{l_2}}$ denote the event count and error count, respectively. For brevity, when $k_1^{l_1} = k_1^{l_2}$ and $k_2^{l_1} = k_2^{l_2}$, we use $2k_1, 2k_2$.

First, we calculate the following parameters:

1. \underline{s}_{11}^Z : Lower bound on the number of single-photon pairs in the Z -basis. This is calculated from the lower bounds of the expected values of z_{10} (z_{01}), which denotes the times where U_1 (U_2) emits a single photon and U_2 (U_1) emits a vacuum state in an $\{\mu_1, \mathbf{o}_2\}$ ($\{\mathbf{o}_1, \mu_2\}$) event:

$$\underline{z}_{10}^* = N_{\text{tot}} p_{\mu_1} p_{\mathbf{o}_2} \mu_1 e^{-\mu_1} \underline{y}_{10}^*, \quad (D7)$$

$$\underline{z}_{01}^* = N_{\text{tot}} p_{\mathbf{o}_1} p_{\mu_2} \mu_2 e^{-\mu_2} \underline{y}_{01}^*, \quad (D8)$$

where \underline{y}_{10}^* and \underline{y}_{01}^* are the yields, estimated using the decoy-state method:

$$\begin{aligned} \underline{y}_{01}^* &\geq \frac{\mu_2}{N_{\text{tot}} (\mu_2 \nu_2 - \nu_2^2)} \\ &\left(\frac{e^{\nu_2} \underline{x}_{\mathbf{o}_1 \nu_2}^*}{p_{\mathbf{o}_1} p_{\nu_2}} - \frac{\nu_2^2}{\mu_2^2} \frac{e^{\mu_2} \bar{x}_{\delta_a \mu_2}^*}{p_{\delta_a} p_{\mu_2}} - \frac{\mu_2^2 - \nu_2^2}{\mu_2^2} \frac{\bar{x}_{\mathbf{o}\mathbf{o}}^{d*}}{p_{\mathbf{o}_1 \mathbf{o}_2}^d} \right), \end{aligned}$$

$$\underline{y}_{10}^* \geq \frac{\mu_1}{N_{\text{tot}}(\mu_1\nu_1 - \nu_1^2)} \left(\frac{e^{\nu_1} \underline{x}_{\nu_1 o_2}^*}{p_{\nu_1} p_{o_b}} - \frac{\nu_1^2}{\mu_1^2} \frac{e^{\mu_1} \bar{x}_{\mu_1 \hat{o}_b}^*}{p_{\mu_a} p_{\hat{o}_b}} - \frac{\mu_1^2 - \nu_1^2}{\mu_1^2} \frac{\bar{x}_{o_o}^{d*}}{p_{o_1 o_2}^d} \right),$$

where $x_{\mathbf{o}\mathbf{o}}^d = x_{\hat{o}_1 \hat{o}_2} + x_{\hat{o}_1 o_2} + x_{o_1 \hat{o}_2}$ is the count of events with at least one declared-vacuum state is sent, and $p_{\mathbf{o}\mathbf{o}}^d = p_{\hat{o}_1} p_{\hat{o}_2} + p_{\hat{o}_1} p_{o_2} + p_{o_1} p_{\hat{o}_2}$ is its probability. The lower bound of s_{11}^{Z*} is

$$\underline{s}_{11}^{Z*} = \frac{\underline{z}_{10}^* \underline{z}_{01}^*}{x_{\text{max}}}. \quad (\text{D9})$$

2. $\underline{s}_{0\mu_2}^Z$: The number of Z -basis events where U_1 sends a vacuum state in both time bins, and U_2 's total intensity is μ_2 . Let z_{00} ($z_{0\mu_2}$) be the number of events where U_1 's state measured to vacuum state in an $\{\mu_1, \mathbf{o}_2\}$ ($\{\mu_1, \mu_2\}$) event. The lower bounds on the expected values are: $\underline{z}_{00}^* = p_{\mu_1} p_{\mathbf{o}_2} e^{-\mu_1} \underline{x}_{o_o}^{d*} / p_{o_1 o_2}^d$ and $\underline{z}_{0\mu_2}^* = p_{\mu_1} p_{\mu_2} e^{-\mu_1} \underline{x}_{\mathbf{o}_a \mu_2}^* / p_{\mathbf{o}_a} p_{\mu_2}$. Using $\underline{x}_{\mathbf{o}_a \mu_2}^* = p_{\mathbf{o}_a} \underline{x}_{\hat{o}_a \mu_2}^* / p_{\hat{o}_a}$, and $\underline{x}_{\mathbf{o}_a \mathbf{o}_2}^* = p_{\mathbf{o}_a} p_{\mathbf{o}_2} \underline{x}_{o_o}^{d*} / p_{o_o}^d$, the lower bound on $s_{0\mu_2}^{Z*}$ is

$$\underline{s}_{0\mu_2}^{Z*} = \frac{\underline{x}_{\mathbf{o}_1 \mu_2}^* \underline{z}_{00}^*}{x_{\text{max}}} + \frac{\underline{x}_{\mathbf{o}_1 \mathbf{o}_2}^* \underline{z}_{0\mu_2}^*}{x_{\text{max}}}. \quad (\text{D10})$$

3. \underline{s}_{11}^X : To determine the expected number of single-photon pairs in the X -basis, we consider the phase difference between U_1 and U_2 , given by $\theta = \theta_1 - \theta_2 + \phi_{12}$. The X -basis events are categorized according to their phase difference θ , where $\theta \in [-\delta, \delta] \cup [\pi - \delta, \pi + \delta]$, and $x_{k_1 k_2}^\theta$ represents the count of events with intensities $\{k_1, k_2\}$ and phase difference θ . Post-processing only retains pairs of events that share the same phase difference θ . Assuming a uniform distribution of θ and a misalignment angle σ , the expected number of single-photon pairs, \underline{s}_{11}^X , is calculated as follows.

$$\begin{aligned} \underline{s}_{11}^{X*} &= \frac{1}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_1 \nu_2}^\theta \times 2 \frac{\nu_2 e^{-(\nu_1+\nu_2)} \underline{y}_{01}^*}{q_{\nu_1 \nu_2}^\theta} \nu_1 e^{-(\nu_1+\nu_2)} \underline{y}_{10}^* d\theta \\ &= \frac{N_{\text{tot}} p_{\nu_a} p_{\nu_b}}{\pi} \int_{\sigma}^{\sigma+\delta} \frac{2\nu_1 \nu_2 e^{-2(\nu_1+\nu_2)} \underline{y}_{01}^* \underline{y}_{10}^*}{q_{\nu_1 \nu_2}^\theta}, \end{aligned} \quad (\text{D11})$$

This calculation uses the gain $q_{\nu_1 \nu_2}^\theta$, which represents the probability of a successful detection event (exactly one detector clicks) given that U_1 selects intensity ν_1 , U_2 selects intensity ν_2 , and the phase difference between their pulses is θ . The total number of events with intensities $\{\nu_1, \nu_2\}$ and phase difference θ is given by $x_{\nu_1 \nu_2}^\theta = N_{\text{tot}} p_{\nu_1} p_{\nu_2} q_{\nu_1 \nu_2}^\theta$, where N_{tot} is the total number of time bins, and p_{ν_1} and p_{ν_2} are the probabilities that U_1 and U_2 choose intensities ν_1 and ν_2 , respectively.

4. \bar{e}_{11}^X : The upper bound on the X -basis bit error rate is derived from the equivalence between the expected Z -basis phase error rate and the expected X -basis bit error

rate for single-photon pairs. The upper bound on the number of single-photon pair errors in the X -basis, \bar{t}_{11}^X , is

$$\bar{t}_{11}^X \leq m^X - \frac{(m_{\nu_1 0, \nu_2 0} + m_{0\nu_1, 0\nu_2})}{\pi} + \bar{m}_{00,00}, \quad (\text{D12})$$

where $m_{\nu_1 0, \nu_2 0}$ and $m_{0\nu_1, 0\nu_2}$ represent error counts for vacuum states in time bins l_1 and l_2 of $\{2\nu_1, 2\nu_2\}$ events, respectively, and $m_{00,00}$ is the error count for vacuum states in both time bins. The expected values of these error counts are:

$$\begin{aligned} \frac{(n_{\nu_1 0, \nu_2 0} + n_{0\nu_1, 0\nu_2})^*}{\pi} &= \frac{2}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_1 \nu_2}^\theta \frac{e^{-(\nu_1+\nu_2)} \underline{q}_{00}^*}{q_{\nu_1 \nu_2}^\theta} d\theta \\ &= \frac{\delta N_{\text{tot}} p_{\nu_a} p_{\nu_b} e^{-(\nu_1+\nu_2)} \underline{q}_{00}^*}{\pi}, \end{aligned} \quad (\text{D13})$$

and

$$\begin{aligned} \bar{n}_{00,00}^* &= \frac{1}{\pi} \int_{\sigma}^{\sigma+\delta} x_{\nu_1 \nu_2}^\theta \left(\frac{e^{-(\nu_1+\nu_2)} \bar{q}_{00}^*}{q_{\nu_1 \nu_2}^\theta} \right)^2 d\theta \\ &= \frac{N_{\text{tot}} p_{\nu_a} p_{\nu_b}}{\pi} \int_{\sigma}^{\sigma+\delta} \frac{e^{-2(\nu_1+\nu_2)} (\bar{q}_{00}^*)^2}{q_{\nu_1 \nu_2}^\theta} d\theta, \end{aligned} \quad (\text{D14})$$

respectively. Here $q_{00}^* = x_{o_1 o_2}^{d*} / (N_{\text{tot}} p_{o_o}^d)$. Because the vacuum state error rate is $1/2$, $\frac{(m_{\nu_1 0, \nu_2 0} + m_{0\nu_1, 0\nu_2})^*}{\pi} = \frac{1}{2} (n_{\nu_1 0, \nu_2 0} + n_{0\nu_1, 0\nu_2})^*$ and $\bar{m}_{00,00}^* = \frac{1}{2} \bar{n}_{00,00}^*$. Therefore, we have:

$$\bar{e}_{11}^X = \bar{t}_{11}^X / \underline{s}_{11}^X. \quad (\text{D15})$$

5. $\bar{\phi}_{11}^Z$: The upper bound on the Z -basis phase error rate is obtained using random sampling without replacement [72], setting a failure probability ε :

$$\bar{\phi}_{11}^Z \leq \bar{e}_{11}^X + \gamma^U (\underline{s}_{11}^Z, \underline{s}_{11}^X, \bar{e}_{11}^X, \varepsilon), \quad (\text{D16})$$

where

$$\gamma^U(n_1, n_2, \lambda, \varepsilon) = \frac{\frac{(1-2\lambda)AG}{n_1+n_2} + \sqrt{\frac{A^2 G^2}{(n_1+n_2)^2} + 4\lambda(1-\lambda)G}}{2 + 2\frac{A^2 G}{(n_1+n_2)^2}}, \quad (\text{D17})$$

with $A = \max\{n_1, n_2\}$ and $G = \frac{n_1+n_2}{n_1 n_2} \ln \frac{n_1+n_2}{2\pi n_1 n_2 \lambda(1-\lambda)\varepsilon^2}$.

6. \underline{s}_{11}^{Zn} , $\underline{s}_{0\mu_2}^{Zn}$ and $\bar{\phi}_{11}^{Zn}$: Within the n -bit string extracted from the Z -basis key, $\underline{s}_{0\mu_2}^{Zn}$ and \underline{s}_{11}^{Zn} denote the lower bounds for the number of vacuum events and single-photon pair events, respectively. The upper bound for the phase error rate of these single-photon pairs is represented by $\bar{\phi}_{11}^{Zn}$. They are calculated by

$$\begin{aligned} \underline{s}_{11}^{Zn} &\geq n \left[\underline{s}_{11}^Z / n^Z - \gamma^U(n, n^Z - n, \underline{s}_{11}^Z / n^Z, \varepsilon) \right], \\ \underline{s}_{0\mu_2}^{Zn} &\geq n \left[\underline{s}_{0\mu_2}^Z / n^Z - \gamma^U(n, n^Z - n, \underline{s}_{0\mu_2}^Z / n^Z, \varepsilon) \right], \\ \bar{\phi}_{11}^{Zn} &\leq \bar{\phi}_{11}^Z + \gamma^U(\underline{s}_{11}^{Zn}, \underline{s}_{11}^Z - \underline{s}_{11}^{Zn}, \bar{\phi}_{11}^Z, \varepsilon). \end{aligned} \quad (\text{D18})$$

The length of the final key for TPTF-QKG with privacy amplification is

$$l_{\text{key}} = \underline{s}_{0\mu_2}^Z + \underline{s}_{11}^Z \left[1 - H(\bar{\phi}_{11}^Z) \right] - n^Z f_{\text{cor}} H(E^Z) - \log_2 \frac{2}{\epsilon_{\text{cor}}} - 2 \log_2 \frac{1}{2\epsilon_{PA}}, \quad (\text{D19})$$

where ϵ_{PA} and ϵ_{cor} are the privacy amplification and error correction failure probability, respectively.

2. SNS-QKG

In the SNS-QKD protocol of Ref. [54], Alice and Bob choose a signal window with probability p_z and a decoy window with probability $1 - p_z$. They send a vacuum state with probability p_{z0} or a phase-randomized weak coherent state of intensity μ_z with probability $1 - p_{z0}$ for the signal window. For the decoy window, they send a vacuum state with probability p_0 , a coherent state of intensity μ_1 with probability p_1 , and a coherent state of intensity μ_2 with probability $1 - p_0 - p_1$. Following the approach in Ref. [54], N_{jk} events are recorded for each intensity pair (jk) , where j and k represent the intensities sent by U_1 and U_2 , respectively, and $jk \in \{00, 01, 10, 02, 20\}$ (intensities 1 and 2 are used). After sifting, n_{jk} one-detector heralded events remain. The counting rate for each intensity pair is defined as $S_{jk} = n_{jk}/N_{jk}$. Then:

$$\begin{aligned} N_{00} &= [(1 - p_z)^2 p_0^2 + 2(1 - p_z)p_z p_0 p_{z0}] N_{\text{tot}}, \\ N_{01} = N_{10} &= [(1 - p_z)^2 p_0 p_1 + (1 - p_z)p_z p_{z0} p_1] N_{\text{tot}}, \\ N_{02} = N_{20} &= [(1 - p_z)^2 (1 - p_0 - p_1)p_0 \\ &\quad + (1 - p_z)p_z p_{z0} (1 - p_0 - p_1)] N_{\text{tot}}. \end{aligned} \quad (\text{D20})$$

To determine the upper bound of e_1^{ph} , two subsets of X_1 windows, denoted as $C_{\Delta+}$ and $C_{\Delta-}$, are defined. The number of instances within each subset, $N_{\Delta\pm}$, is calculated as follows.

$$N_{\Delta\pm} = \frac{\Delta}{2\pi} (1 - p_z)^2 p_1^2 N_{\text{tot}}. \quad (\text{D21})$$

Let $n_{\Delta+}^R$ denote the number of effective events where the right detector clicks within the $C_{\Delta+}$ subset, and $n_{\Delta-}^L$ denote the number of effective events where the left detector clicks within the $C_{\Delta-}$ subset. The overall counting error rate for these subsets, T_{Δ} , is then calculated as

$$T_{\Delta} = \frac{n_{\Delta+}^R + n_{\Delta-}^L}{2N_{\Delta\pm}}. \quad (\text{D22})$$

The lower bound for the expected counting rate of untagged single photons, denoted as \underline{s}_1^{Z*} , is given by

$$\underline{s}_1^{Z*} \geq \underline{s}_1^{Z*} = \frac{1}{2\mu_1\mu_2(\mu_2 - \mu_1)} [\mu_2^2 e^{\mu_1} (\underline{S}_{01}^* + \underline{S}_{10}^*) - \mu_1^2 e^{\mu_2} (\bar{S}_{02}^* + \bar{S}_{20}^*) - 2(\mu_2^2 - \mu_1^2) \bar{S}_{00}^*]. \quad (\text{D23})$$

In this equation, S_{jk}^* represents the expected value of the counting rate S_{jk} . The upper and lower bounds of this expected value, estimated from the observed data, are denoted by \bar{S}_{jk}^* and \underline{S}_{jk}^* , respectively. These bounds account for statistical uncertainties in the experimental measurements.

The upper bound for the expected phase-flip error rate of untagged single photons, denoted as \bar{e}_1^{ph*} , is expressed as

$$\bar{e}_1^{ph*} = \frac{\bar{T}_{\Delta}^* - \frac{1}{2} e^{-2\mu_1} \underline{S}_{00}^*}{2\mu_1 e^{-2\mu_1} \underline{s}_1^{Z*}}, \quad (\text{D24})$$

using the vacuum state error rate is $1/2$.

With total transmittance η :

$$n_{00} = 2p_d(1 - p_d)N_{00}, \quad (\text{D25})$$

$$n_{01} = n_{10} = 2 \left[(1 - p_d)e^{\eta\mu_1/2} - (1 - p_d)^2 e^{-\eta\mu_1} \right] N_{01}, \quad (\text{D26})$$

$$n_{02} = n_{20} = 2 \left[(1 - p_d)e^{\eta\mu_2/2} - (1 - p_d)^2 e^{-\eta\mu_2} \right] N_{02}, \quad (\text{D27})$$

$$n^Z = n_{\text{signal}} + n_{\text{error}}, \quad (\text{D28})$$

$$E_z = \frac{n_{\text{error}}}{n_t}, \quad (\text{D29})$$

$$n_{\Delta+}^R = n_{\Delta-}^L = [T_X(1 - 2e_d) + e_d S_X] N_{\Delta\pm}, \quad (\text{D30})$$

where we have:

$$\begin{aligned} n_{\text{signal}} &= 4N_{\text{tot}} p_z^2 p_{z0} (1 - p_{z0}) \\ &\quad \left[(1 - p_d)e^{-\eta\mu_z/2} - (1 - p_d)^2 e^{-2\eta\mu_z} \right], \\ n_{\text{error}} &= 2N_{\text{tot}} p_z^2 (1 - p_{z0})^2 \\ &\quad \left[(1 - p_d)e^{-\eta\mu_z} I_0(\eta\mu_z) - (1 - p_d)^2 e^{-2\eta\mu_z} \right] \\ &\quad + 2N_{\text{tot}} p_z^2 p_{z0} p_d (1 - p_d), \end{aligned} \quad (\text{D31})$$

and

$$\begin{aligned} T_X &= \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} (1 - p_d) e^{-2\eta\mu_1 \cos^2 \frac{\delta}{2}} d\delta - (1 - p_d)^2 e^{-2\eta\mu_1}, \\ S_X &= \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} (1 - p_d) e^{-2\eta\mu_1 \sin^2 \frac{\delta}{2}} d\delta - (1 - p_d)^2 e^{-2\eta\mu_1} + T_X. \end{aligned} \quad (\text{D32})$$

In SNS-QKG, the values of an n -bit key extracted from an n^Z -bit key are calculated as

$$\begin{aligned} \underline{s}_{1n}^{Z*} &= n \left[\underline{s}_1^{Z*} / n^Z - \gamma^U (n, n^Z - n, \underline{s}_1^{Z*} / n^Z, \epsilon) \right], \\ \bar{e}_{1n}^{ph*} &= \bar{e}_1^{ph*} + \gamma^U \left(\underline{s}_{1n}^{Z*}, \underline{s}_1^{Z*} - \underline{s}_{1n}^{Z*}, \bar{e}_1^{ph*}, \epsilon \right), \end{aligned} \quad (\text{D33})$$

and the amount of secret information is

$$H_n = \underline{s}_{1n}^{Z*} \left[1 - H(\bar{e}_{1n}^{ph*}) \right] - \lambda_{EC}, \quad (\text{D34})$$

where $\lambda_{EC} = n f_{\text{cor}} H(E_z)$. The key length is then calculated by

$$l_{\text{key}} = \underline{s}_1^{Z*} \left[1 - H(\bar{e}_1^{ph*}) \right] - \lambda_{EC} - \log_2 \left(\frac{2}{\epsilon_{\text{cor}}} \right) - 2 \log_2 \frac{1}{\sqrt{2\epsilon_{PA}\hat{\epsilon}}}. \quad (\text{D35})$$

3. BB84-QKG

Following [55], the count of vacuum and single-photon clicks of the X -basis are

$$s_{X,0} \geq \tau_0 \frac{\mu_2 n_{X,\mu_3}^- - \mu_3 n_{X,\mu_2}^+}{\mu_2 - \mu_3}, \quad (\text{D36})$$

$$s_{X,1} \geq \frac{\tau_1 \mu_1 \left[n_{X,\mu_2}^- - n_{X,\mu_3}^+ - \frac{\mu_2 - \mu_3}{\mu_1^2} (n_{X,\mu_1}^+ - \frac{s_{X,0}}{\tau_0}) \right]}{\mu_1 (\mu_2 - \mu_3) - \mu_2^2 + \mu_3^2}. \quad (\text{D37})$$

where $\tau_n := \sum_{\mu \in \mathcal{K}} e^{-\mu} \mu^n p_\mu / n!$ is the probability of the n -photon state in a pulse with intensity μ , and

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

We use μ_1 , μ_2 and μ_3 to represent the signal and two decoy intensities.

Using Z -basis statistics and following a similar calculation procedure, the number of vacuum events ($s_{Z,0}$) and

single-photon events ($s_{Z,1}$) for $\mathcal{Z} = \bigcup_{k \in \mathcal{K}} \mathcal{Z}_k$ are determined. The single-photon phase error rate in the X -basis, $\phi_{X,1}$, is then calculated as follows.

$$\phi_{X,1} := \frac{c_{X,1}}{s_{X,1}} \leq \frac{v_{Z,1}}{s_{Z,1}} + \gamma^U \left(s_{Z,1}, s_{X,1}, \frac{v_{Z,1}}{s_{Z,1}}, \epsilon_{\text{sec}} \right), \quad (\text{D38})$$

where

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

and

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

The total number of events in the X -basis is given by $n_X = \sum_{k \in \mathcal{K}} n_{X,k}$, and the total number of errors is $m_X = \sum_{k \in \mathcal{K}} m_{X,k}$. For a selected n -bit string in BB84-QKG, the following parameters are estimated by

$$s_{X,0}^n \geq n \left[\frac{s_{X,0}}{n^Z} - \gamma^U \left(n, n^Z - n, \frac{s_{X,0}}{n^Z}, \epsilon \right) \right], \quad (\text{D39})$$

$$s_{X,1}^n \geq n \left[\frac{s_{X,1}}{n^Z} - \gamma^U \left(n, n^Z - n, \frac{s_{X,1}}{n^Z}, \epsilon \right) \right], \quad (\text{D40})$$

$$\phi_{X,1}^n \leq \phi_{X,1} + \gamma^U \left(s_{X,1}^n, s_{X,1} - s_{X,1}^n, \bar{\phi}_{X,1}, \epsilon \right). \quad (\text{D41})$$

Thus we have

$$H_n = \underline{s}_{X,0}^n + \underline{s}_{X,1}^n \left[1 - H(\bar{\phi}_{X,1}^n) \right] - \lambda_{EC}, \quad (\text{D42})$$

where $\lambda_{EC} = n f_{\text{cor}} H(m_X/n_X)$. The key length is then calculated by

$$l_{\text{key}} = \underline{s}_{X,0} + \underline{s}_{X,1} \left[1 - H(\bar{\phi}_{X,1}^n) \right] - \lambda_{EC} - \log_2 \left(\frac{2}{\epsilon_{\text{cor}}} \right) - 6 \log_2 \frac{22}{\epsilon_{PA}}. \quad (\text{D43})$$

-
- [1] D. Bayer, S. Haber, and W. S. Stornetta, Improving the efficiency and reliability of digital time-stamping, in *Sequences II*, edited by R. Capocelli, A. De Santis, and U. Vaccaro (Springer New York, New York, NY, 1993) pp. 329–334.
- [2] Y. Zhang, C. X. Xu, N. Cheng, H. Li, H. Yang, and X. Shen, Chronos⁺⁺: An accurate blockchain-based time-stamping scheme for cloud storage, *IEEE Transactions on Services Computing* **13**, 216 (2020).
- [3] G. Estevam, L. M. Palma, L. R. Silva, J. E. Martina, and M. Vigil, Accurate and decentralized timestamping using smart contracts on the ethereum blockchain, *Inf. Process. Manag.* **58**, 102471 (2021).
- [4] G. M. Y. R. Qiang Wu, Zhaoyang Han, Distributed

- timestamp mechanism based on verifiable delay functions, *Comput. Syst. Sci. Eng.* **44**, 1633 (2023).
- [5] G. Culot, F. Fattori, M. Podrecca, and M. Sartor, Addressing industry 4.0 cybersecurity challenges, *IEEE Eng. Manag. Rev.* **47**, 79 (2019).
- [6] P. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.
- [7] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review* **41**, 303 (1999).
- [8] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* **79**, 325 (1997).

- [9] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, Experimental realization of Shor's quantum factoring algorithm using qubit recycling, *Nat. Photon.* **6**, 773 (2012).
- [10] A. K. Fedorov, E. O. Kiktenko, and A. I. Lvovsky, Quantum computers put blockchain security at risk, *Nature* **563**, 465 (2018).
- [11] E. Gouzien and N. Sangouard, Factoring 2048-bit rsa integers in 177 days with 13 436 qubits and a multimode memory, *Phys. Rev. Lett.* **127**, 140503 (2021).
- [12] M.-G. Zhou, X.-Y. Cao, Y.-S. Lu, Y. Wang, Y. Bao, Z.-Y. Jia, Y. Fu, H.-L. Yin, and Z.-B. Chen, Experimental quantum advantage with quantum coupon collector, *Research* **2022**, 9798679 (2022).
- [13] C. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (IEEE Press, New York, 1984) pp. 175–179.
- [14] H.-L. Yin, T.-Y. Chen, Z.-W. Yu, H. Liu, L.-X. You, Y.-H. Zhou, S.-J. Chen, Y. Mao, M.-Q. Huang, W.-J. Zhang, H. Chen, M. J. Li, D. Nolan, F. Zhou, X. Jiang, Z. Wang, Q. Zhang, X.-B. Wang, and J.-W. Pan, Measurement-Device-Independent Quantum Key Distribution Over a 404 km Optical Fiber, *Phys. Rev. Lett.* **117**, 190501 (2016).
- [15] 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).
- [16] 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).
- [17] 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).
- [18] M.-Y. Li, X.-Y. Cao, Y.-M. Xie, H.-L. Yin, and Z.-B. Chen, Finite-key analysis for coherent one-way quantum key distribution, *Phys. Rev. Res.* **6**, 013022 (2024).
- [19] Z.-Q. Zhong, X.-H. Zhan, J.-L. Chen, S. Wang, Z.-Q. Yin, J.-Q. Geng, D.-Y. He, W. Chen, G.-C. Guo, and Z.-F. Han, Hyperentanglement quantum communication over a 50 km noisy fiber channel, *Optica* **11**, 1056 (2024).
- [20] X.-H. Zhan, Z.-Q. Zhong, J.-Y. Ma, S. Wang, Z.-Q. Yin, W. Chen, D.-Y. He, G.-C. Guo, and Z.-F. Han, Experimental demonstration of long distance quantum communication with independent heralded single photon sources, *npj Quantum Inf.* **11**, 1 (2025).
- [21] D. Pan, G.-L. Long, L. Yin, Y.-B. Sheng, D. Ruan, S. X. Ng, J. Lu, and L. Hanzo, The evolution of quantum secure direct communication: On the road to the qinternet, *IEEE Communications Surveys & Tutorials* **26**, 1898 (2024).
- [22] Z.-Z. Sun, Y.-B. Cheng, M. Wang, L. Qian, D. Ruan, D. Pan, and G.-L. Long, Quantum blockchain relying on quantum secure direct communication network, *IEEE Internet of Things Journal*, 1 (2025).
- [23] D. Pan, Y.-C. Liu, P. Niu, H. Zhang, F. Zhang, M. Wang, X.-T. Song, X. Chen, C. Zheng, and G.-L. Long, Simultaneous transmission of information and key exchange using the same photonic quantum states, *Sci. Adv.* **11**, eadt4627 (2025).
- [24] V. Dunjko, P. Wallden, and E. Andersson, Quantum digital signatures without quantum memory, *Phys. Rev. Lett.* **112**, 040502 (2014).
- [25] H.-L. Yin, Y. Fu, and Z.-B. Chen, Practical quantum digital signature, *Phys. Rev. A* **93**, 032316 (2016).
- [26] J.-Q. Qin, C. Jiang, Y.-L. Yu, and X.-B. Wang, Quantum digital signatures with random pairing, *Phys. Rev. Applied* **17**, 044047 (2022).
- [27] 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).
- [28] 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).
- [29] 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).
- [30] B.-H. Li, Y.-M. Xie, X.-Y. Cao, C.-L. Li, Y. Fu, H.-L. Yin, and Z.-B. Chen, One-time universal hashing quantum digital signatures without perfect keys, *Phys. Rev. Appl.* **20**, 044011 (2023).
- [31] M. Fitzi, N. Gisin, and U. Maurer, Quantum solution to the Byzantine agreement problem, *Phys. Rev. Lett.* **87**, 217901 (2001).
- [32] S. Gaertner, M. Bourennane, C. Kurtsiefer, A. Cabello, and H. Weinfurter, Experimental demonstration of a quantum protocol for byzantine agreement and liar detection, *Phys. Rev. Lett.* **100**, 070504 (2008).
- [33] E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. Lvovsky, and A. K. Fedorov, Quantum-secured blockchain, *Quantum Sci. Technol.* **3**, 035004 (2018).
- [34] C.-X. Weng, R.-Q. Gao, Y. Bao, B.-H. Li, W.-B. Liu, Y.-M. Xie, Y.-S. Lu, H.-L. Yin, and Z.-B. Chen, Beating the fault-tolerance bound and security loopholes for Byzantine agreement with a quantum solution, *Research* **6**, 0272 (2023).
- [35] X. Jing, C. Qian, C.-X. Weng, B.-H. Li, Z. Chen, C.-Q. Wang, J. Tang, X.-W. Gu, Y.-C. Kong, T.-S. Chen, H.-L. Yin, D. Jiang, B. Niu, and L.-L. Lu, Experimental quantum byzantine agreement on a three-user quantum network with integrated photonics, *Sci. Adv.* **10**, eadp2877 (2024).
- [36] X.-Y. Cao, B.-H. Li, Y. Wang, Y. Fu, H.-L. Yin, and Z.-B. Chen, Experimental quantum e-commerce, *Sci. Adv.* **10**, eadk3258 (2024).
- [37] F. Pastawski, N. Y. Yao, L. Jiang, M. D. Lukin, and J. I. Cirac, Unforgeable noise-tolerant quantum tokens, *Proc. Natl. Acad. Sci.* **109**, 16079 (2012).
- [38] J.-Y. Guan, J. M. Arrazola, R. Amiri, W. Zhang, H. Li, L. You, Z. Wang, Q. Zhang, and J.-W. Pan, Experimental preparation and verification of quantum money, *Phys. Rev. A* **97**, 032338 (2018).
- [39] M. Bozzio, A. Orioux, L. Trigo Vidarte, I. Zaguine, I. Kerenidis, and E. Diamanti, Experimental investigation of practical unforgeable quantum money, *npj Quantum Inf.* **4**, 5 (2018).
- [40] M. Bozzio, E. Diamanti, and F. Grosshans, Semi-device-independent quantum money with coherent states, *Phys. Rev. A* **99**, 022336 (2019).

- [41] K. Horodecki and M. Stankiewicz, Semi-device-independent quantum money, *New J. Phys.* **22**, 023007 (2020).
- [42] P. Schiаны, J. Kalb, E. Sztatecsny, M.-C. Roehsner, T. Guggemos, A. Trenti, M. Bozzio, and P. Walther, Demonstration of quantum-digital payments, *Nat. Commun.* **14**, 3849 (2023).
- [43] M. Hillery, V. Bužek, and A. Berthiaume, Quantum secret sharing, *Phys. Rev. A* **59**, 1829 (1999).
- [44] R. Cleve, D. Gottesman, and H.-K. Lo, How to share a quantum secret, *Phys. Rev. Lett.* **83**, 648 (1999).
- [45] K.-J. Wei, H.-Q. Ma, and J.-H. Yang, Experimental circular quantum secret sharing over telecom fiber network, *Opt. Express* **21**, 16663 (2013).
- [46] B. P. Williams, J. M. Lukens, N. A. Peters, B. Qi, and W. P. Grice, Quantum secret sharing with polarization-entangled photon pairs, *Phys. Rev. A* **99**, 062311 (2019).
- [47] M. De Oliveira, I. Nape, J. Pinnell, N. TabeBordbar, and A. Forbes, Experimental high-dimensional quantum secret sharing with spin-orbit-structured photons, *Phys. Rev. A* **101**, 042303 (2020).
- [48] 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).
- [49] 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).
- [50] C.-L. Li, Y. Fu, W.-B. Liu, Y.-M. Xie, B.-H. Li, M.-G. Zhou, H.-L. Yin, and Z.-B. Chen, Breaking the rate-distance limitation of measurement-device-independent quantum secret sharing, *Phys. Rev. Res.* **5**, 033077 (2023).
- [51] Y.-R. Xiao, Z.-Y. Jia, Y.-C. Song, Y. Bao, Y. Fu, H.-L. Yin, and Z.-B. Chen, Source-independent quantum secret sharing with entangled photon pair networks, *Opt. Lett.* **49**, 4210 (2024).
- [52] Y.-Z. Wang, X.-R. Sun, X.-Y. Cao, H.-L. Yin, and Z.-B. Chen, Experimental coherent-state quantum secret sharing with finite pulses, *Phys. Rev. Appl.* **22**, 044018 (2024).
- [53] 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).
- [54] C. Jiang, Z.-W. Yu, X.-L. Hu, and X.-B. Wang, Unconditional security of sending or not sending twin-field quantum key distribution with finite pulses, *Phys. Rev. Appl.* **12**, 024061 (2019).
- [55] 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).
- [56] X.-B. Wang, Beating the photon-number-splitting attack in practical quantum cryptography, *Phys. Rev. Lett.* **94**, 230503 (2005).
- [57] H.-K. Lo, X. Ma, and K. Chen, Decoy state quantum key distribution, *Phys. Rev. Lett.* **94**, 230504 (2005).
- [58] G. Brassard and L. Salvail, Secret-key reconciliation by public discussion, in *Advances in Cryptology — EUROCRYPT '93*, edited by T. Helleseth (Springer Berlin Heidelberg, Berlin, Heidelberg, 1994) pp. 410–423.
- [59] 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 (2008) pp. 637–641.
- [60] T. Lunghi, J. B. Brask, C. C. W. Lim, Q. Lavigne, J. Bowles, A. Martin, H. Zbinden, and N. Brunner, Self-testing quantum random number generator, *Phys. Rev. Lett.* **114**, 150501 (2015).
- [61] Z. Cao, H. Zhou, X. Yuan, and X. Ma, Source-independent quantum random number generation, *Phys. Rev. X* **6**, 011020 (2016).
- [62] M. Herrero-Collantes and J. C. Garcia-Escartin, Quantum random number generators, *Rev. Mod. Phys.* **89**, 015004 (2017).
- [63] Y. Liu, Q. Zhao, M.-H. Li, J.-Y. Guan, Y. Zhang, B. Bai, W. Zhang, W.-Z. Liu, C. Wu, X. Yuan, *et al.*, Device-independent quantum random-number generation, *Nature* **562**, 548 (2018).
- [64] W.-B. Liu, Y.-S. Lu, Y. Fu, S.-C. Huang, Z.-J. Yin, K. Jiang, H.-L. Yin, and Z.-B. Chen, Source-independent quantum random number generator against tailored detector blinding attacks, *Opt. Express* **31**, 11292 (2023).
- [65] B. Korzh, C. C. W. Lim, R. Houlmann, N. Gisin, M. J. Li, D. Nolan, B. Sanguinetti, R. Thew, and H. Zbinden, Provably secure and practical quantum key distribution over 307 km of optical fibre, *Nat. Photonics* **9**, 163 (2015).
- [66] J. Carter and M. N. Wegman, Universal classes of hash functions, *J. Comput. Syst. Sci.* **18**, 143 (1979).
- [67] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography* (CRC press, 2018).
- [68] H. Krawczyk, LFSR-based hashing and authentication, in *Annual International Cryptology Conference* (Springer, 1994) pp. 129–139.
- [69] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed. (Cambridge University Press, 2013).
- [70] V. Shoup, On fast and provably secure message authentication based on universal hashing, in *Advances in Cryptology — CRYPTO '96*, edited by N. Kobitz (Springer Berlin Heidelberg, Berlin, Heidelberg, 1996) pp. 313–328.
- [71] J. Massey, Shift-register synthesis and bch decoding, *IEEE Trans. Inf. Theory* **15**, 122 (1969).
- [72] 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).