Practical issues in quantum-key-distribution postprocessing

Chi-Hang Fred Fung,[1,*] Xiongfeng Ma,[2,†] and H. F. Chau[1,‡]

[1]*Department of Physics and Center of Theoretical and Computational Physics, University of Hong Kong, Pokfulam Road, Hong Kong*

[2]*Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, 200 University Ave W., Waterloo, Ontario, Canada N2L 3G1*

(Received 6 October 2009; published 21 January 2010)

Quantum key distribution (QKD) is a secure key generation method between two distant parties by wisely exploiting properties of quantum mechanics. In QKD, experimental measurement outcomes on quantum states are transformed by the two parties to a secret key. This transformation is composed of many logical steps (as guided by security proofs), which together will ultimately determine the length of the final secret key and its security. We detail the procedure for performing such classical postprocessing taking into account practical concerns (including the finite-size effect and authentication and encryption for classical communications). This procedure is directly applicable to realistic QKD experiments and thus serves as a recipe that specifies what postprocessing operations are needed and what the security level is for certain lengths of the keys. Our result is applicable to the BB84 protocol with a single or entangled photon source.

DOI: 10.1103/PhysRevA.81.012318 PACS number(s): 03.67.Dd, 03.67.Hk

I. INTRODUCTION

In theory, a few quantum key distribution (QKD) protocols, such as the Bennett-Brassard-1984 (BB84) [1], Bennett-Brassard-Mermin-1992 (BBM92) [2], Bennett-1992 (B92) [3], and six-state [4,5] protocols, have been proven to be unconditionally secure in the last decade [6–12]. Security of other protocols, such as the Ekert91 protocol [13] and the device-independent QKD protocol [14], have also been studied. For a review of QKD, one can refer to Refs. [15–17].

QKD schemes can be classified into two types: prepare-and-measure scheme and entanglement-based scheme. In the former, one party, Alice, prepares the quantum signals (say, using a laser source) according to her basis and bit values and sends them through a quantum channel to the other party, Bob, who measures them on reception. In the latter type, an entanglement source emits pairs of entangled signals, which are then measured in certain bases chosen by Alice and Bob separately. There is an important difference in terms of security between the emitted signals with practical sources in the two cases. In the prepare-and-measure case, the signals emitted by Alice (say, a weak coherent-state source) is basis dependent, meaning that the coherent-state signal corresponding to one basis is quantum mechanically different from that of the other basis. An eavesdropper, Eve, can certainly leverage this information to her advantage. New techniques such as the decoy-state method [18–25], strong-reference-pulse QKD [26–29], and DPS [30–33] have recently been invented to allow the use of coherent-state sources securely. On the other hand, entanglement-based QKD involves signals that are basis independent. The security of basis-independent QKD (including entanglement-based QKD with a PDC source and prepare-and-measure QKD with a single-photon source) has been proven in Ref. [34] and the performance of entanglement-based QKD with a PDC source has been analyzed recently [35].

As security analysis of QKD has become mature, it now comes to the stage to consider all the underlying assumptions and to apply the theoretical results to practical QKD experiments. Although standard security proofs (such as Ref. [9]) imply a procedure for distilling a final secret key from measurement outcomes, such a procedure cannot be directly carried out in an *actual* QKD experiment because many of the security proofs focus on the case that the key is arbitrarily long. Although, in theory, there is not a fundamental limit on the length, it is constrained by the computational power in practice. Therefore, it is imperative to quantify the finite-size effect and to provide a precise postprocessing recipe that one can follow for distilling final secret keys with quantified security in real QKD experiments. This is the motivation of this article. We remark that it is not that the security proofs are incorrect, but carrying them out in practice requires more additional consideration on the relation between the actual steps taken and the final security parameter.

Ultimately, QKD system designers would like to know the classical computation and communications needed to transform the measurement results of a QKD experiment to a final key. Furthermore, it is important to know the trade-off between the final key length and the security parameter, since this allows one to estimate the number of initial quantum signals to be sent in order to achieve a certain final key length and security. We provide the solution to this in the current article.

It is important to note that the postprocessing procedure contains many elements including authentication, error correction and verification, phase error rate estimation, and privacy amplification. Integrating all these elements with a security proof is nontrivial and the resultant postprocessing procedure is the main contribution of our article. Our article uses the latest security proof techniques to perform postprocessing analysis, along a similar line to an early work by Lütkenhaus [36]. We emphasize that the main focus of our work is the overall procedure for postprocessing rather than analyzing a security proof in the finite-key situation. We note that, recently, lots of efforts have been spent on the finite-key effect in QKD postprocessing, such as Refs. [37–40].

*chffung@hku.hk
†xfma@iqc.ca
‡hfchau@hkusua.hku.hk

The finite-key-length analysis is important not only from a theoretical point view but also for experiments. For example, the efficient BB84 protocol [41] is proposed to increase the key generation rate. In order to select an optimal bias between the two bases, X and Z, Alice and Bob need to consider statistical fluctuations. We will address this issue in this article. We remark that the proposed postprocessing scheme ties up a few existing results with some modifications. Key features of our work are as follows:

1. a strict bound for the phase error estimation is derived;
2. an authentication scheme is applied for the error verification;
3. the efficiency of the privacy amplification is investigated;
4. parameter optimization is studied.

This article is an expansion of our shorter article [42] which summarizes the essential components of our data postprocessing procedure. All the technical details related to the procedure are presented here.

The article is organized as follows. Section II introduces the assumptions used in the article. Section III discusses the security aspect of our procedure. Section IV outlines the postprocessing procedures. Section V introduces some preliminary tools to be used in later sections. Sectioins VI–X discuss the details of the various postprocessing steps. In Sec. XI, we investigate the parameter optimization problem for this postprocessing procedure. In Sec. XII, we simulate an experiment setup as an example. We conclude in Sec. XIII.

II. ASSUMPTIONS

Here, we examine the underlying assumptions used in the postprocessing scheme we propose in this article. We emphasize that in order to apply the scheme to a QKD system, one needs to compare these assumptions with the real setup. The assumptions used in the article are listed as follows:

1. Alice and Bob perform the BB84 protocol with a perfect single photon source (or basis-independent photon source [35]);
2. The detection system is compatible with the squashing model [43,44] (see also Ref. [45]). For example, the efficiency mismatch is not considered in this article;
3. Alice and Bob use perfect random number generators and perfect key management. They share a certain amount of secure key prior to running their QKD system.

III. SECURITY ASPECT

Our data postprocessing procedure is derived from entanglement distillation protocol- (EDP) based security proofs [8,9,46] (also see Ref. [47]) and thus our procedure is secure against the most general attacks allowed by the laws of quantum mechanics. The original idea [8] casts QKD as distilling Einstein-Podolsky-Rosen (EPR) pairs between Alice and Bob, involving correcting general quantum errors. And the ability to correct general quantum errors is equivalent to the ability to correct bit and phase errors [48,49]. Later, Shor and Preskill [9] show that correcting bit errors and phase errors in the EDP picture correspond to bit error correction and privacy amplification in distilling a secret key. Thus, proving the security of QKD can be cast as showing that both bit and phase errors are corrected in the EDP picture. In this way, provided that a quantum error correction code exists for the specific bit and phase error rates in the EDP picture, the security of the corresponding QKD protocol is proved. However, this places a rather strong requirement on the quantum error correction code since constraints on *both* bit and phase error rates have to be satisfied. Fortunately, Lo [46] further shows that bit and phase errors can be decoupled by simply encrypting the bit error syndrome transmission (without affecting the net key generation rate). Koashi [47] adopts the same decoupling mechanism and further generalizes the notion of phase errors with a simple and yet powerful argument. In this article, we follow this line of security proofs in our finite-key analysis. Essentially, the important ingredients in our analysis are

1. encrypting the bit error syndromes;
2. using a random sampling argument to place bounds on the phase error rate;
3. using a privacy amplification scheme (with structure) and placing bounds on its phase error correcting capacity;
4. integrating authentication in classical communications.

A. Composable security

The finite key analysis is closely related to the definition of security. Currently, the composable security definition of QKD [50,51] is widely accepted as the most stringent security definition in the field. QKD is composable in the sense that the final key generated is indistinguishable from an ideal secret key except with a small probability, and thus the key can be used in a subsequent cryptographic task where an ideal key is expected. A secret key is considered ideal if it is identical between Alice and Bob and is private to Eve (i.e., Eve has no information on it). The notion of composability was first proposed in the classical setting for the study of security when composing classical cryptographic protocols in a complex manner [52,53]. Composability has also been carried over to the quantum setting [50,54]. One essential feature of the composable security definition is that it characterizes the security of a protocol with respect to the ideal functionality. In particular, the security of a composable secret key generated by QKD is measured with the trace distance between the real situation with the real key and the ideal situation with an ideal key [12,51,55].

Definition 1 [12,51,55]. A random variable V (the classical key) drawn from the set $\mathcal{V}$ is said to be ζ secure with respect to an eavesdropper holding a quantum system E if

$$\tfrac{1}{2}\mathrm{Tr}|\rho_{VE} - \rho_U \otimes \rho_E| \leqslant \zeta \tag{1}$$

where $\rho_{VE} = \sum_{v\in\mathcal{V}} P_V(v)|v\rangle\langle v| \otimes \rho_{E|V=v}$, $\rho_U = \sum_{v\in\mathcal{V}} |v\rangle\langle v|/|\mathcal{V}|$ represents an ideal key taking values uniformly over $\mathcal{V}$, and $|\mathcal{V}|$ is the size of $\mathcal{V}$. Here, $\mathrm{Tr}|A| = \sum_i |\lambda_i|$ where λ_i are the eigenvalues of A.

Since the trace distance $\frac{1}{2}\mathrm{Tr}|\rho - \sigma|$ is the maximum probability of distinguishing between the two quantum states ρ and σ, this security definition naturally gives rise to the

operational meaning that the ζ-secure key V is identical to an ideal key U except with probability ζ. This trace-distance security parameter is additive when practical cryptographic protocols are composed [50]. That is to say, suppose we have a key generation protocol (e.g., QKD) and a second cryptographic protocol which consumes an ideal secret key. Furthermore, we suppose that the first protocol realizes an ideal key generation scheme with a security parameter ζ_1 for a particular secret key output, and the second protocol realizes its ideal functionality with a security parameter ζ_2. Then, when the two protocols are composed (i.e., the imperfect key generated in the first protocol is used in the second protocol), the overall security parameter will become $\zeta_1 + \zeta_2$.

Our article is based on EDP-based proofs which often justify security with the fidelity between Alice and Bob's state and the ideal state (the perfect EPR pairs). This fidelity is a direct consequence of the failure probability of the postprocessing procedure. Thus, we need to find a connection of this failure probability with the composability definition in Definition 1.

The generation of the final key in one round of QKD is composed of many steps (cf. Sec. IV and Table I) and each step carries a certain failure probability. This probability represents the case that Alice and Bob believe the step has succeeded but actually not (in other word, a case of undetected failure). A detected failure in any step will lead a premature termination of the QKD process. Successes of all steps will result in a perfect final key that is private to Eve and identical between Alice and Bob. However, since each step may fail without being detected, there is a certain probability that the final key fails to be perfect and this probability is upper bounded by the sum of the failure probabilities of all the steps. Essentially, this sum is the failure probability ε of the entire postprocessing procedure, which needs to be converted to the security parameter of the final key.

In the context of Koashi's proof [47], success of the phase error correcting step as part of the postprocessing procedure guarantees that Alice's m-qubit state ρ_A can be corrected to the pure state $0_X^{\otimes m}$. The final key is then generated by m measurements in the Z basis on ρ_A. Since the entire postprocessing procedure fails with probability at most ε, the component of Alice's state ρ_A corresponding to the pure state $0_X^{\otimes m}$ must satisfy $\langle 0_X^{\otimes m}|\rho_A|0_X^{\otimes m}\rangle \geqslant 1-\varepsilon$. In order to make the connection with the universal composability definition in Definition 1, it has been suggested [55] (also see Ref. [50]) that a bound on the trace distance is obtained from the fidelity using a general inequality relating them [56]:

$$\tfrac{1}{2}\mathrm{Tr}|\rho-\sigma| \leqslant \sqrt{1-F(\rho,\sigma)^2}, \tag{2}$$

where $F(\rho,\sigma)=\mathrm{Tr}\sqrt{\rho^{1/2}\sigma\rho^{1/2}}$ is the fidelity between ρ and σ. Thus, we seek the minimum of the fidelity between ρ_{AE} and $|0_X^{\otimes m}\rangle_A\langle 0_X^{\otimes m}|\otimes\rho_E$ in order to get an upper bound on their trace distance, in accordance with Definition 1. Since the fidelity never decreases under a trace-preserving quantum operation [i.e., $F(\mathcal{E}(\rho),\mathcal{E}(\sigma)) \geqslant F(\rho,\sigma)$], system E can be considered to be the entire purification of system A when the minimum occurs. Assuming this worst case, the joint state is of the form

$$|\Psi_{AE}\rangle = \sqrt{\alpha}\,|0_X^{\otimes m}\rangle_A|0\rangle_E + \sqrt{1-\alpha}\,|\Psi^{\perp}\rangle_{AE}, \tag{3}$$

where $|\Psi^{\perp}\rangle_{AE}$ has unit norm, ${}_A\langle 0_X^{\otimes m}|\Psi^{\perp}\rangle_{AE}=0$, and $\alpha \geqslant 1-\varepsilon$. The fidelity between the real situation and the ideal situation is (see Appendix A for proof)

$$F\left(\rho_{AE}, |0_X^{\otimes m}\rangle_A\langle 0_X^{\otimes m}|\otimes\rho_E\right) \geqslant \alpha \tag{4}$$

$$\geqslant 1-\varepsilon, \tag{5}$$

where $\rho_{AE}=|\Psi_{AE}\rangle\langle\Psi_{AE}|$ and $\rho_E=\mathrm{Tr}_A(\rho_{AE})$. Thus, an upper bound on the failure probability provided by the EDP-based proofs can easily be translated to a composable security measure. By substituting Eq. (5) for the fidelity in Eq. (2) and using the fact that projection onto the eigenstates of the Z basis corresponding to the final measurement does not increase the trace distance,[1] we conclude that the final key is $\sqrt{\varepsilon(2-\varepsilon)}$ secure in accordance with Definition 1.

Lemma 1. When the failure probability of the postprocessing procedure is ε, the final key is $\sqrt{\varepsilon(2-\varepsilon)}$ secure in accordance with Definition 1.

We can apply this lemma when many rounds of QKD are composed. Suppose the postprocessing of each round fails with a probability ε, and Alice and Bob plan to use a QKD system a million times in the manner that the secret key output of one round is fed as input to the next. Since the trace-distance measure is additive when cryptographic protocols are composed [50], the trace-distance security parameter for the key in the last round will be $10^6\sqrt{\varepsilon(2-\varepsilon)}$. Note that this security parameter increases linearly with the number of rounds of the QKD system. This linear dependence is an important feature of the composability security definition.

We note that Mayers's security proof [6,7] has also implicitly mentioned using failure probability to quantify the security.

B. Equivalence of the failure probability and the trace distance as the optimization objective

The failure probability of the postprocessing procedure ε is related to the trace-distance security parameter ζ by $\zeta=\sqrt{\varepsilon(2-\varepsilon)}$. Since $\frac{d\zeta}{d\varepsilon}>0$ and $\frac{d^2\zeta}{d\varepsilon^2}<0$, we have $\zeta^{(1)}>\zeta^{(2)} \Leftrightarrow \varepsilon^{(1)}>\varepsilon^{(2)}$ and there is a one-to-one mapping between these two measures. Thus, minimizing either ε or ζ subject to the same constraints (such as a fixed key length) will produce the same solution.

C. Simple lower bound on failure probability

It is easy to lower bound the failure probability as a function of the secret-key cost k_{initial} by considering that Eve has a $2^{-k_{\mathrm{initial}}}$ chance of guessing the right initial secret key and thus will be able to launch a man-in-the-middle attack successfully. Therefore, the failure probability of any postprocessing scheme should be at least $2^{-k_{\mathrm{initial}}}$. Moreover, the failure probability of our scheme exhibits the same exponential decrease as the lower bound (see the various constituent failure probabilities ε's listed in Table I).

[1]The trace distance never increases under a trace-preserving quantum operation, i.e., $\mathrm{Tr}|\mathcal{E}(\rho)-\mathcal{E}(\sigma)| \leqslant \mathrm{Tr}|\rho-\sigma|$.

TABLE I. List of resource cost and the failure probabilities in the various steps. Lengths of preshared secret key bits are designated with k while the failure probabilities with ε. The relevant equations involving these quantities are also shown.

Step	Message length	Message encrypted?	Tag length	Failure probability
1. Key sift	N	–	–	–
2. Basis sift	$2n$	No	$2k_{bs}$	$2\varepsilon_{bs}$ [Eq. (11)]
3. Bit error correction	k_{ec} [Eq. (12)]	Yes	–	–
4. Error verification	–	–	k_{ev}	ε_{ev} [Eq. (14)]
5. Phase error estimation	–	–	–	ε_{ph} [Eq. (21)]
6. Privacy amplification	$(n_x + n_z + l - 1)$	No	k_{pa}	ε_{pa} [Eq. (31)]

IV. OUTLINE OF POSTPROCESSING PROCEDURE

The postprocessing procedure is listed as follows. We remark that each communication between Alice and Bob consists of a message and an authentication tag, each of which may be of zero length. In our scheme, a tag is transmitted if and only if authentication is used and in this case the authentication tag is always encrypted by a one-time pad, consuming some preshared secret bits. When a message is transmitted, it may or may not be encrypted, and it is assumed to be unencrypted unless otherwise stated. Note that no message but a tag is transmitted in the error verification step (step 4). Figure 1 shows the flow chart of our data postprocessing procedure.

1. Key sift [not authenticated]: Alice sends N quantum signals to Bob, of which n signals produce clicks. Bob discards all no-click events and obtains n-bit raw key by randomly assigning values to the double-click events.[2] Note that other key sift procedures might be applied as well (see, for example, Ref. [57]).
2. Basis sift [authenticated]: Alice and Bob send each other n-bit basis information. In the end of this step, Alice and Bob obtain n_x (n_z)-bit sifted key in the $X(Z)$ basis. Define the bias ratio to be $q_x \equiv n_x/(n_x + n_z)$. Note that this bias ratio differs from the probability the Alice and Bob choose the two bases. Define the probability that Alice and Bob choose the X basis to be p_x, then in the long key limit, $q_x = p_x^2/[p_x^2 + (1 - p_x)^2]$.
3. Error correction [not authenticated but encrypted, Sec. VII]: Alice and Bob perform error correction so that Bob's raw key matches Alice's. The classical messages exchanged in this process are encrypted. If error correction fails, Alice and Bob abort the QKD process.
4. Error verification [Sec. VIII]: Alice and Bob want to make sure (with a high probability) that their keys after the error correction step are identical. If error verification fails, Alice and Bob either go back to the error correction step or abort the QKD process. We note that the idea of using error verification to replace error testing is proposed by Lütkenhaus [36].
5. Phase error rate estimation [no communication, Sec. IX]: Alice and Bob use the bit error rate measured in the $X(Z)$ basis to infer the phase error rate in the $Z(X)$ basis. The uncertainty in bounding the phase error rates are quantified by a random sampling argument.
6. Privacy amplification [authenticated, Sec. X]: Alice randomly generates an $(n_x + n_z + l - 1)$-bit random bit string and sends to Bob through an authenticated channel. Alice and Bob use this random bit string to generate a Toeplitz matrix. The final key (with a size of l) will be the product of this matrix (with a size of $(n_x + n_z) \times l$) and the key string (with a size of $n_x + n_z$).
7. The final secure key length (net growth) is given by

$$NR \geqslant l - k_{bs} - k_{ec} - k_{ev} - k_{pa} \tag{6}$$

with a failure probability of

$$\varepsilon \leqslant \varepsilon_{bs} + \varepsilon_{ev} + \varepsilon_{ph} + \varepsilon_{pa}, \tag{7}$$

where l is given by Eq. (32). Here, the k's are the secret-key costs and the ε's are the failure probabilities for steps 2–6 (see Table I). Throughout the article, ε's with various footnotes stand for various failure probabilities.

[2]In the case of a passive-basis-selection setup, Bob also randomly assigns basis value X or Z for double clicks [44].

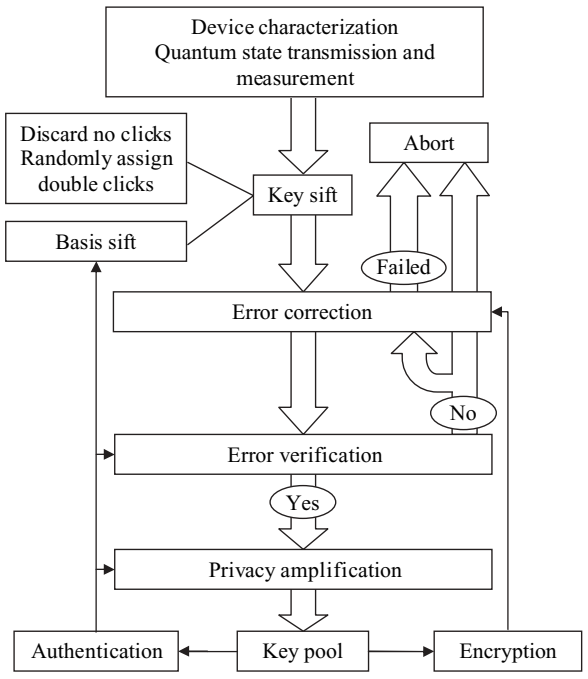

FIG. 1. Flow chart of our data postprocessing procedure [58].

V. PRELIMINARY

A. Data representation

Data are represented as matrices or column vectors of 0's and 1's. Additions are carried out in modulo 2. For example, a raw key x is multiplied by a privacy amplification matrix M to generate the final $y = Mx$, where the ith bit of y is $\sum_j M(i, j)x(j)$ mod 2.

B. Toeplitz matrices

In our framework, we rely heavily on a particular class of hash functions to perform phase error correction, error verification, and authentication. We are interested in using sets of Toeplitz matrices to perform these tasks. Toeplitz matrices are Boolean matrices with a special structure:

$$M = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-m+1} \\ a_1 & a_0 & a_{-1} & \ddots & \\ a_2 & a_1 & \ddots & & \vdots \\ \vdots & \ddots & & & \\ a_{l-1} & & \cdots & & a_{l-m} \end{pmatrix} \tag{8}$$

where $a_i = 0, 1$. It can also be concisely described by $M_{(i,j)} = a_{i-j}$, where $M_{(i,j)}$ is the (i, j) element of M. The advantage of using Toeplitz matrices is that it can be specified by a small number of parameters, namely, $m + l - 1$ bits, as opposed to ml bits for completely random matrices. Hashing of a given column vector x (whose elements are 0 or 1) can be performed by choosing a Toeplitz matrix M randomly and computing the hash value as Mx.

In our postprocessing scheme, we use Toeplitz matrices for three purposes: privacy amplification, error verification, and authentication. We remark that fully random Toeplitz matrices (specified by $m + l - 1$ random bits) are used for privacy amplification, while for error verification and authentication, Toeplitz matrices specified by a smaller number $(2l)$ of random bits are used in order to save secret bits (see Sec. V C below).

C. Authentication

Alice and Bob can authenticate their classical communications with a family of Toeplitz matrices. For every (classical) message they need to authenticate, both parties select a hash function from a fixed family using preshared secret bits. The sending party computes the hash value for the message (called the tag) and sends both to the other party, who also computes the hash value for the received message and can conclude that the message originates from the legitimate party if both hash values are identical.

In authentication, the key component is the construction of hashing function. Wegman and Carter proposed unconditional secure authentication schemes [59,60] by introducing the universal hashing function families, which are also used for privacy amplification. Afterwards, lots of efforts have been spent on how to construct a universal hashing function family effectively. In this article, we use the LFSR-based Toeplitz matrix construction by Krawczyk [61,62] for authentication.

Here we state the result of the LFSR-based Toeplitz matrix construction, which is given by Theorem 9 of Ref. [61] by Krawczyk. The authentication scheme based on the LFSR-based Toeplitz matrix construction is secure with a failure probability of

$$\varepsilon_{au} = n2^{-k+1}, \tag{9}$$

where k is the length of the tag, n is the length of the message. The authentication scheme can be stated as follows. Alice and Bob use a $2k$-bit secure key to construct a Toeplitz matrix with a size of $(k \times n)$ by a LFSR. The authenticated tag is generated by multiplying the matrix and the message. Then they encrypt the tag by another k-bit secure key. Since the tag is encrypted by a one-time pad, the $2k$-bit key used for the Toeplitz matrix construction is still secure [61]. Hence, the net secure-key cost for this scheme is k.

We remark that in Krawczyk's later result [62], the secure key required for the LFSR-based Toeplitz matrix construction can be reduced to an arbitrary number r, with sacrifice of failure probability,

$$\varepsilon_{au} = \frac{1}{2^k} + \frac{k + n - 1}{2^{r/2}} \tag{10}$$

One can see that by choosing $r = 2k$, Eq. (9) gives a slightly tighter bound than Eq. (10) for the failure probability. Since the secure-key cost is at least k in this construction due to one-time pad encryption, for simplicity, we use Eq. (9) for authentication and error verification.

We remark that, as pointed out in Ref. [61], the LFSR-based Toeplitz matrix construction is highly practical in real-life implementation.

VI. BASIS SIFT

Alice and Bob send each other n-bit basis information. Due to the symmetry, we can assume the same failure probability for the two message exchanges [61]:

$$\varepsilon_{bs} = n2^{-k_{bs}+1}. \tag{11}$$

Here, Alice and Bob use a $2k_{bs}$-bit secure key to construct a Toeplitz matrix with a size of $(k_{bs} \times n)$ by a LFSR. The authenticated tag is generated by multiplying the matrix and the message. Then they encrypt the two tags by two k_{bs}-bit secure keys. The total secure-key cost in this step is $2k_{bs}$ (for the one-time-pad encryption of the tags) and the corresponding failure probability is $2\varepsilon_{bs}$. Note that when Alice and Bob use a biased basis choice [41], they can exchange less than n-bit classical information for basis sift by data compression. Since the secure-key cost only logarithmically depends on the length of the message, we simply use n for the following discussion. In the end of this step, Alice and Bob obtain n_x (n_z)-bit sifted key in the X (Z) basis. Define the bias ratio to be $q_x \equiv n_x/(n_x + n_z)$ as in Sec. IV.

VII. ERROR CORRECTION

For simplicity of discussion, we assume that Bob tries to correct his raw key to match Alice's. This means that we assume no advantage distillation [63,64]. Error correction is done by Alice sending parity information of her raw key to Bob encrypted with secret bits from the key pool. The secure-key

cost is given by

$$k_{ec} = n_x f(e_{bx}) H(e_{bx}) + n_z f(e_{bz}) H(e_{bz}), \tag{12}$$

where $f(x)$ is the error correction efficiency, and

$$H(x) = -x \log_2(x) - (1-x)\log_2(1-x) \tag{13}$$

is the binary entropy function. In practice, Alice and Bob only need to count the amount of classical communication used in the error correction. That is, the value of k_{ec} can be directly obtained from the postprocessing. After the error correction, Alice and Bob count the number of errors in the X (Z) basis: $e_{bx}n_x$ ($e_{bz}n_z$). Note that although we assume encryption of the parity information here (cf. Sec. III), it may be avoided by basing the postprocessing on other security proofs. In this case, there may be some restriction on the error correction procedure and more privacy amplification may be required. In practice, there is an advantage to using error correction without encryption, since if Alice and Bob abort the QKD procedure, no secret bits will be lost due to encryption.

There is no failure probability associated with error correction in our postprocessing scheme. Identity between Alice's and Bob's sifted keys is verified with an error verification step (Sec. VIII below).

VIII. ERROR VERIFICATION

Suppose Alice and Bob each holds a bit string **a** and **b**. They can verify the identity of their strings by exchanging shorter strings which are the hash values $f(\mathbf{a})$ and $f(\mathbf{b})$. Identity of the two hash values provides confidence that the two strings are the same. Below we argue that error verification is the same as authentication, and thus we use the same procedure for both purposes. This procedure and the associated properties is described in the authentication section (Sec. V C).

A. Relation to authentication

In QKD postprocessing, authenticated classical communication is required to overcome the man-in-the-middle attack. The objective of the authentication procedure can be stated as follows. Alice sends Bob a message through a (classical) channel, which is accessible to Eve. Alice uses some authentication scheme to make sure (with a high probability) that the message is not modified during the transmission. This classical problem is well studied in the literature [59–62]. One traditional solution is for Alice to add a redundant code (tag) to the message to be sent. The tag-message pair is designed in such a way that whenever the message is modified, Bob can detect it (with a high probability).

Error verification, on the other hand, is the procedure for ensuring (with a high probability) that the bit strings (or keys) owned by Alice and Bob are identical. One natural way to do this is by random hashing. For example, Alice randomly hashes her bit string and sends the hash value to Bob. Bob uses the same hash function to obtain his hash value and compares to the one sent by Alice. The probability that Alice and Bob possess different bit strings (keys) decrease exponentially with the number of rounds of hashing.

By comparing the two procedures, authentication and error verification, one can see their commonality. In order to show the link between the two procedures, we break down the authentication procedure into two parts: Alice sends Bob the message first and then the tag. Let us take a look at the stage where Bob just receives the message sent by Alice (but before the tag). Now, Alice and Bob each has a bit string. In authentication, Alice sends a tag (corresponding to her message) to Bob and Bob verifies it. The claim of a secure authentication scheme is that if the tag passes Bob's test, the probability that Alice and Bob share the same string is high. This is exactly what is asked in the error verification procedure. Therefore, secure authentication schemes can be used for the error verification.

We remark that the only difference between the two procedures is that authentication does not care whether the tag reveals information about the message or not, but error verification does (at least for our use in QKD postprocessing). This difference can be easily overcome by encrypting the tag, which has already been done in some authentication schemes including the one we use in this article.

Thus, in this procedure, Alice sends an encrypted tag of an authentication scheme to Bob. The failure probability for this step, k_{ev}, similar to Eq. (11), is

$$\varepsilon_{ev} = (n_x + n_z) 2^{-k_{ev}+1}. \tag{14}$$

IX. PHASE ERROR RATE ESTIMATION

In the BB84 protocol, Alice and Bob measure the bit error rate in the X basis, e_{bx}, to estimate the phase error rate in the Z basis, e_{pz}, and vice versa. In the infinite length limit, the error *rates*, e_{bx} and e_{pz}, converge to the underlying probabilities, p_{bx} and p_{pz}. Due to the symmetry of BB84, we know that $p_{bx} = p_{pz}$, from which follows $e_{bx} = e_{pz}$ in the asymptotic case. With a finite key size, the rate is fluctuating around the corresponding probability. Now the question can be stated as a random sampling problem: given the bit error rate in the X basis (e_{bx}), the sample size (n_x), and the population size ($n_x + n_z$), upper bound the phase error rate in the Z basis (e_{pz}), with a probability $1 - P_{\theta x}$,

$$P_{\theta x} \equiv \Pr\{e_{pz} \geqslant e_{bx} + \theta_x\}, \tag{15}$$

where θ_x represents the deviation of the phase error rate from the tested value (the bit error rate in another basis) due to the finite-size effect. Here n_x and n_z are the number of sifted key bits in the X and Z basis, respectively. The failure probability $P_{\theta x}$ will be related to the failure probability of the phase error rate estimation step [see Eq. (21)].

A. Random sampling

Define two random variables: $k \equiv e_{bx}n_x$ and $m \equiv e_{pz}n_z + e_{bx}n_x$. The number of bit errors in the X basis, k, can be accurately (with a high probability) counted after the error verification procedure. Note that m denotes the number of bit errors *if* Bob measures all $n_x + n_z$ qubits in the X basis. In the squashing model [43,44,65], Eve prepares the qubit received by Bob. Hence, one can assume Eve chooses a distribution of m, $\Pr\{m\}$, before Bob's detection.

In order to link the probability, $P_{\theta x}$, to the measurement results, n_x, n_z and k (or e_{bx}), we go back to the original

definition of the security parameter in QKD. In the security analysis of QKD, $P_{\theta x}$ denotes the probability that Eve sets up a distribution $\Pr\{m\}$ (by preparing qubits) and then Bob obtains k bit errors in the X basis. Therefore, the mathematical definition for $P_{\theta x}$ is

$$\begin{aligned} P_{\theta x} &= \Pr\{e_{pz} \geqslant e_{bx} + \theta_x, e_{bx}\} \\ &= \Pr\{m \geqslant e_{bx}(n_x + n_z) + \theta_x n_z, k\} \\ &= \sum_{m=e_{bx}(n_x+n_z)+\theta_x n_z}^{e_{bx}n_x+n_z} \Pr\{m,k\} \\ &= \sum_{m=e_{bx}(n_x+n_z)+\theta_x n_z}^{e_{bx}n_x+n_z} \Pr\{k|m\}\Pr\{m\}. \end{aligned} \tag{16}$$

Bob chooses to measure the X basis randomly (without replacement, of course), and thus $\Pr\{k|m\}$ is given by a hypergeometric function,

$$\Pr\{k|m\} = \frac{\binom{m}{k}\binom{n_x+n_z-m}{n_x-k}}{\binom{n_x+n_z}{n_x}}. \tag{17}$$

It is not hard to prove that Eq. (17) is a strict decreasing function of m when $m > (n_x + n_z)e_{bx}$. Thus, from Eq. (16),

$$\begin{aligned} P_{\theta x} &\leqslant \Pr\{k|m = e_{bx}(n_x + n_z) + \theta_x n_z\} \\ &< \frac{\sqrt{n_x + n_z}}{\sqrt{e_{bx}(1 - e_{bx})n_x n_z}} 2^{-(n_x+n_z)\xi_x(\theta_x)}, \end{aligned} \tag{18}$$

where the first equality holds when Eve sets the probability distribution to be a delta function $\Pr\{m\} = \delta_{m,e_{bx}(n_x+n_z)+\theta_x n_z}$. The derivation of the second inequality is presented in Appendix B. Note that all the variables in Eq. (18) can be measured in practice. The function $\xi_x(\theta)$ is given by

$$\begin{aligned} \xi_x(\theta_x) \equiv\ & H(e_{bx} + \theta_x - q_x\theta_x) - q_x H(e_{bx}) \\ & - (1 - q_x)H(e_{bx} + \theta_x), \end{aligned} \tag{19}$$

where $q_x = n_x/(n_x + n_z)$ is the bias ratio.

A similar formula for the failure probability of phase error rate estimation in the X basis, $P_{\theta z}$, can also be derived,

$$P_{\theta z} < \frac{\sqrt{n_x + n_z}}{\sqrt{e_{bz}(1 - e_{bz})n_x n_z}} 2^{-(n_x+n_z)\xi_z(\theta_z)} \tag{20}$$

with $\xi_z(\theta_z)$ is defined by $\xi_z(\theta_z) \equiv H(e_{bz} + \theta_z - q_z\theta_z) - q_z H(e_{bz}) - (1 - q_z)H(e_{bz} + \theta_z)$ and $q_z = n_z/(n_x + n_z)$. Combining the failure probabilities for the X basis and Z basis, the total failure probability of phase error rate estimation, ε_{ph}, is then given by

$$\varepsilon_{ph} \leqslant P_{\theta x} + P_{\theta z}. \tag{21}$$

In case $e_{bx} = 0$ (or $e_{bz} = 0$), one can replace it by $n_x e_{bx} = 1$ (or $n_z e_{bz} = 1$) to get around the singularity as shown in Appendix B. One can see that $\xi_x(\theta_x)$ is positive when $\theta_x > 0$ and $0 \leqslant e_{bx}, e_{bx} + \theta_x \leqslant 1$, due to concavity of the binary entropy function $H(x)$.

B. Large data size approximation

In the limit of large n_x and n_z, θ_x can be chosen to be small. Then we can use Taylor expansion for Eq. (19),

$$\begin{aligned} \xi_x(\theta_x) =\ & H(e_{bx} + \theta_x - q_x\theta_x) - q_x H(e_{bx}) \\ & - (1 - q_x)H(e_{bx} + \theta) \\ =\ & -\frac{1}{2}(1 - q_x)q_x H''(e_{bx})\theta_x^2 + O\left(\theta_x^3\right) \\ =\ & \frac{\ln 2}{2}\frac{(1 - q_x)q_x}{(1 - e_{bx})e_{bx}}\theta_x^2 + O\left(\theta_x^3\right). \end{aligned} \tag{22}$$

When $q_x = 1/2$, i.e., $n_x = n_z$, and θ_x is small, the failure probability is given by

$$P_{\theta_x} < \frac{1}{2\sqrt{2ne_{bx}(1 - e_{bx})}} e^{-\frac{\theta_x^2 n}{4(1-e_{bx})e_{bx}}} \tag{23}$$

Except for the factor $1/[2\sqrt{2ne_{bx}(1 - e_{bx})}]$, this is what is used in the literature, such as Refs. [9,35]. In practice, normally we have $2\sqrt{2ne_{bx}(1 - e_{bx})} > 1$, so the bound given by Eq. (23) is tighter than what is used in the literature.

X. PRIVACY AMPLIFICATION

In view of the EDP picture, we regard privacy amplification as the result of phase error correction. In the following, we focus on using two-universal hashing to perform phase error correction and determine the corresponding failure probability.

A. Two-universal hashing

The family of all Toeplitz matrices $\{M\}$ of size $l \times m$ has 2^{l+m-1} elements and satisfies the following property:

$$\Pr\{Mx = My\} = \frac{1}{2^l} \quad \text{for all} \quad x \neq y, \tag{24}$$

where it is assumed that each matrix is chosen with equal probability. This can be proved by slightly adapting the proof of Claim 7 of Ref. [66]. Note that the family of hash functions performed with Toeplitz matrices is one specific case of a more general class known as the two-universal families of hash functions. A family of hash functions mapping S to T is called two-universal [59] if

$$\Pr\{f(x) = f(y)\} \leqslant \frac{1}{|T|} \quad \text{for all} \quad x \neq y, \tag{25}$$

where $f(x)$ is a hash function chosen in the family of hash functions and in our case $f(x) = Mx$. Two-universal families of hash functions have many useful properties and we will rely on some of the them in this article.

B. Error correction

Suppose Alice holds a bit string, $\mathbf{a}$, and Bob a noisy version of it, $\mathbf{b}$. The difference between the two strings $\mathbf{e} = \mathbf{a} \oplus \mathbf{b}$ is the error pattern. Let S be the set of all possible error patterns. Alice and Bob intend to use a family of two-universal linear hash functions to correct errors in Bob's string with respect to Alice's. A hash function $f(\cdot)$ is selected from the family and Alice and Bob each computes the hash value of their bit strings with the hash function. Alice sends Bob her hash value, to which Bob adds his hash value to arrive at the hash value of the error pattern $f(\mathbf{e}) = f(\mathbf{a}) \oplus f(\mathbf{b})$. Note that this is valid due to the linearity of the hash functions. Using this hash value, Bob can identify the error pattern and thus correct the errors in his string. Suppose that there are $|T|$ possible outputs for this family of hash functions. Using the definition of a two-universal family given in Eq. (25), we can bound the

probability of incorrectly identifying the error pattern as

$$\Pr\left\{\bigcup_{\mathbf{e}'\in S\setminus\mathbf{e}} f(\mathbf{e})=f(\mathbf{e}')\right\}\leqslant\frac{|S|}{|T|},\tag{26}$$

which follows from applying the union bound to Eq. (25). Thus, Bob's error-corrected string matches Alice's with probability at least $1-|S|/|T|$.

Although this hashing-based error correcting procedure may not be as practical and efficient as conventional ones, it is useful for phase error correction in security proofs [9,46,47]. This is because for security purpose we only need to show that the phase error pattern is identified without actually correcting the error [9], and we only need a bound on the probability of successfully identifying the error pattern.

C. Privacy amplification and phase error correction

Suppose we perform privacy amplification using a set of $l\times m$ Toeplitz matrices, a member of which can be selected with $l+m-1$ random bits. Here, l is the final key length and m is the sifted key length. For each matrix M in the privacy amplification set, we associate an $(m-l)\times m$ matrix $M^{\perp}$ that is orthogonal to M. The collection of all these matrices $\{M^{\perp}\}$ forms the set of hash functions for phase error correction. We would like to find out whether this set $\{M^{\perp}\}$ has the property of Eq. (25). If it does, we can determine the successful probability of phase error correction from Eq. (26).

We remark that it does not matter whether the matrices of the set $\{M^{\perp}\}$ have the Toeplitz form or not since we do not need to generate them but only need to make sure that there exists such a set with a certain phase error correcting capability. On the other hand, we do impose the Toeplitz form on the privacy amplification set $\{M\}$ since we actually need to generate this set.

Indeed, it can be shown (see, e.g., Theorem 1 of Ref. [67]) that when M is chosen from a set of random Toeplitz matrices, the corresponding matrices $M^{\perp}$ also form a two-universal set, i.e.,

$$\Pr\{M^{\perp}x=M^{\perp}y\}\leqslant\frac{1}{2^{m-l}}\quad\text{for all}\quad x\neq y.$$

Thus, according to the discussion in Sec. X B, we can use the set $\{M^{\perp}\}$ to identify the phase error pattern and perform the correction. In essence, when (i) there are $|S|$ number of possible phase error patterns, (ii) the sifted key length is m, and (iii) the final key length is l, the failure probability of phase error correction is upper bounded by

$$\varepsilon_{pc}=|S|2^{-(m-l)}.\tag{27}$$

(Note that the sifted key length here will be equal to $m=n_x+n_z$ when it is used in the next subsection.)

For BB84, the bit error rates for the Z bits and X bits are exactly determined from the error correction procedures, up to a certain probability given by the verification step [cf. Eq. (14)]. Focusing on the Z bits, we can estimate its phase error rate e_{pz} from the actual bit error rate of the X bits e_{bx} using the random sampling argument of Sec. IX. Accordingly, the lower and upper bounds on the number of phase errors on the Z bits are, except with probability P_{θ_x} [which is bounded in Eq. (18)],

$$0\leqslant e_{pz}n_z<(e_{bx}+\theta_x)n_z.\tag{28}$$

[Note that the second inequality is a strictly less than due to the definition of P_{θ_x} in Eq. (15).] Therefore, with probability at least $1-P_{\theta_x}$, the number of possible phase error patterns in the Z bits is

$$|S_z|=\sum_{k=0}^{\lceil(e_{bx}+\theta_x)n_z-1\rceil}\binom{n_z}{k}<\binom{n_z}{(e_{bx}+\theta_x)n_z}\tag{29}$$

$$<2^{n_zH(e_{bx}+\theta_x)},\tag{30}$$

where the first inequality holds for $e_{bx}+\theta_x<1/3$ (see Appendix C for proof of the first inequality and see, e.g., Refs. [68,69] for proof of the second inequality). We can similarly obtain the bound for the number of possible phase error patterns in the X bits $|S_x|$. Combining the number of patterns in the Z and X bits, we have $|S|=|S_z||S_x|$ in Eq. (27).

D. Key length

Alice and Bob determine the size of the matrix, $l\times(n_x+n_z)$, used for hashing. Here, l is the key length after the privacy amplification. Alice generates an (n_x+n_z+l-1)-bit random bit string and sends it to Bob through an authenticated channel. Alice and Bob use this random bit string to generate a Toeplitz matrix. The final key (with a size of l) will be the product of this matrix [with a size of $l\times(n_x+n_z)$] and the key string (with a size of n_x+n_z) after passing through the error verification. The overall failure probability of the privacy amplification is the sum of that for authentication [Eq. (9)] and that for phase error correction [Eq. (27)]:

$$\varepsilon_{pa}=(n_x+n_z+l-1)2^{-k_{pa}+1}+2^{-t_{oe}},\tag{31}$$

where k_{pa} is the secure-key cost for the authentication and t_{oe} is related to Eq. (27) by $2^{-t_{oe}}=\varepsilon_{pc}$. By rearranging Eqs. (27) and (30), the final key length is

$$l=n_x[1-H(e_{bz}+\theta_z)]+n_z[1-H(e_{bx}+\theta_x)]-t_{oe}.\tag{32}$$

The first term in Eq. (31) gives the failure probability of the authentication for the (n_x+n_z+l-1)-bit random bit string transmission. The second term in Eq. (31) gives the failure probability of privacy amplification using the Toeplitz matrix. In the equivalent EDP used in the security proof [9,47], the second term in Eq. (31) gives the failure probability of the phase error correction.

XI. OPTIMIZATION

Alice and Bob calibrate the QKD system to get an estimate of the transmittance η, the error rates e_{bx} and e_{bz}. Through some rough calculation of the target length of the final key, they decide the acceptable confidence interval $1-\varepsilon$ and fix the length of the experiment, N, which denotes the number pulses sent by Alice. Then roughly, the length of the raw key is $n=N\eta$. After basis sift, Alice and Bob share an n_x-bit (n_z-bit) key in the X (Z) basis.

Alice and Bob can optimize their postprocessing using either the failure probability or the trace distance as the security measure, since they are directly related to one another as discussed in Sec. III B. Here, we will use the failure probability as the security measure for our discussion. The failure probability ε is chosen by Alice and Bob according to the later practical use of the final key. The desired message security level sets an upper bound threshold value for ε. Thus, the exact value of ε is not strictly predetermined. That is, it can slightly deviate from the predetermined threshold value.

Given n and ε [cf. Eq. (7)], Alice and Bob need to optimize all the parameters for the key generation rate given in Eq. (6). The first parameter they want to optimize is the basis bias ratio, $q_x = n_x/(n_x + n_z)$ which (roughly) determines the probabilities to choose the X and Z bases, p_x and p_z, by $q_x \approx p_x^2/(p_x^2 + p_z^2)$. The bias ratio should be determined before quantum transmission while all other parameters can be determined right after a raw key is obtained. The initial calibration process gives Alice and Bob some idea about the basis ratio which they will use in the subsequent QKD process. The remaining parameters that need to be optimized are as follows: k_{bs}, k_{ec}, k_{ev}, k_{pa}, ε_{bs}, ε_{ev}, ε_{ph}, ε_{pa} and t_{oe}. Alice and Bob need to balance the failure probabilities from each step [cf. Eq. (7)] and the secure-key cost [cf. Eq. (6)]. The optimization problem becomes the following: given the total failure probability

$$\begin{aligned}\varepsilon &\leqslant 2\varepsilon_{bs} + \varepsilon_{ev} + \varepsilon_{ph} + \varepsilon_{pa} \\ &= 2n2^{-k_{bs}+1} + (n_x + n_z)2^{-k_{ev}+1} + \varepsilon_{ph} \\ &\quad + (n_x + n_z + l - 1)2^{-k_{pa}+1} + 2^{-t_{oe}},\end{aligned} \tag{33}$$

maximize the final key length

$$NR \geqslant l - 2k_{bs} - k_{ec} - k_{ev} - k_{pa} \tag{34}$$

Note that the parameters k_{bs}, k_{ev}, k_{pa}, and t_{oe} affect ε and the final key rate in similar ways. Also, error correction and phase error rate estimation mainly depend on the bias ratio. Thus, Alice and Bob can group the secure key costs and failure probabilities into two parts by defining $\varepsilon_3 \equiv 2\varepsilon_{bs} + \varepsilon_{ev} + \varepsilon_{pa}$ and $k_3 \equiv 2k_{bs} + k_{ev} + k_{pa} + t_{oe}$ [see Eqs. (32), (6), and (7)]. The final secure key length can be rewritten as

$$\begin{aligned}NR \geqslant\ & n_x[1 - f(e_{bx})H(e_{bx}) - H(e_{bz} + \theta_z)] \\ & + n_z[1 - f(e_{bz})H(e_{bz}) - H(e_{bx} + \theta_x)] - k_3.\end{aligned} \tag{35}$$

We remark that if the contribution from one basis is negative in Eq. (35), Alice and Bob should use the detections from this basis for parameter estimation only, but not for the key.

We consider the subproblem: given the failure probability

$$\begin{aligned}\varepsilon_3 &\leqslant 2\varepsilon_{bs} + \varepsilon_{ev} + \varepsilon_{pa} \\ &= 2n2^{-k_{bs}+1} + (n_x + n_z)2^{-k_{ev}+1} \\ &\quad + (n_x + n_z + l - 1)2^{-k_{pa}+1} + 2^{-t_{oe}},\end{aligned} \tag{36}$$

minimize the secret-key cost

$$k_3 \geqslant 2k_{bs} + k_{ev} + k_{pa} \tag{37}$$

With the inequality of arithmetic and geometric means, one can show that the optimized secure-key cost for each step is given by

$$\begin{aligned}t_{oe} &= \frac{k_3}{5} - \frac{4}{5} - \frac{1}{5}\log_2 A \\ k_{bs} &= t_{oe} + 1 + \log_2 n \\ k_{ev} &= t_{oe} + 1 + \log_2(n_x + n_z) \\ k_{pa} &= t_{oe} + 1 + \log_2(n_x + n_z + l - 1),\end{aligned} \tag{38}$$

where $A = n^2(n_x + n_z)(n_x + n_z + l - 1)$. The corresponding failure probability is

$$\varepsilon_3 = 5A^{1/5}2^{-(k_3-4)/5} \tag{39}$$

From Eq. (39), we have

$$k_3 = -5\log_2 \varepsilon_3 + \log_2 A + 4 + 5\log_2 5. \tag{40}$$

Note that $n^4/4 < A < 2n^4$ and also $\varepsilon = \varepsilon_3 + \varepsilon_{ph}$. Here if Alice and Bob allow ε to have a small deviation from the predetermined value, they can put a *soft* lower bound for ε_3 in the optimization. The exact value of the *soft* lower bound is not really important here as long as it is within the tolerable fluctuation range of ε. Here, we simply choose the tolerable deviation to be within 1%, which implies that $10^{-2}\varepsilon < \varepsilon_3 < \varepsilon$. Thus,

$$\begin{aligned}&-5\log_2 \varepsilon + 4\log_2 n + 2 + 5\log_2 5 < k_3 \\ &\quad < -5\log_2 \varepsilon + 4\log_2 n + 15 + 15\log_2 5.\end{aligned} \tag{41}$$

This is true for all θ_x, θ_z, and q_x. Note that the difference between the lower bound and the soft upper bound of k_3 is less than 37 bits. When the final key length is much larger than 37 bits, Alice and Bob can set

$$k_3 = -5\log_2 \varepsilon + 4\log_2 n + 50 \tag{42}$$

and the failure probability ε_3 will satisfy $\varepsilon_3 < 10^{-2}\varepsilon$ since the right-hand side of Eq. (42) is larger than the upper bound in Eq. (41).

Since Alice and Bob will recalculate the failure probability in the end and allow the final ε to have a small deviation from the predefined value, they can safely use $\varepsilon_{ph} = \varepsilon$ in the optimization of the basis bias. Thus, the simplified optimization problem only has three parameters to be optimized: q_x, θ_x, and θ_z, given $\varepsilon_{ph} = \varepsilon - \varepsilon_3 \approx \varepsilon$.

In summary, the simplified optimization procedure for a target failure probability ε is as follows:

1. Compute k_3 using Eq. (42);
2. Maximize the key rate in Eq. (35) over q_x, θ_x, and θ_z subject to $\varepsilon_{ph} = \varepsilon$. Here, ε_{ph} is related to the three optimization variables by Eqs. (18), (20), and (21);
3. After the optimization, they can recalculate the final failure probability $\varepsilon = \varepsilon_3 + \varepsilon_{ph}$, where ε_3 is given in Eq. (39).

As discussed above, since one can set $\varepsilon_3 < 10^{-2}\varepsilon$ (when the key length is much larger than 37 bits), the failure probabilities for basis sift, error verification, and privacy amplification are relatively small, and the failure probability for random sampling is the major contribution to the total failure probability.

Observation. The main effect of the finite key analysis for the QKD postprocessing stems from the phase error rate

estimation. Inefficiencies due to authentication, error verification, and privacy amplification are relatively insignificant.

XII. SIMULATIONS

Now let us consider an example of the postprocessing in the simple case of symmetric errors in the two bases.

Suppose $N = 10^{10}$, $\eta = 10^{-3}$ (then $n = N\eta = 10^7$), $e_{bx} = e_{bz} = 4\%$, and $\varepsilon = 10^{-7}$. It is not hard to see that the final key length is much larger than 37 bits. Thus, the simplified optimization is used. First, the secure-key cost $k_3 = 543$ bit, according to Eq. (42). Second, given $n = 10^7$, $e_{bx} = e_{bz} = 4\%$ and $\varepsilon = 10^{-7}$, we optimize the parameters: θ_x, θ_z, and q_x. Through a numerical program, we get $\theta_x = 1.07\%$, $\theta_z = 0.84\%$, and $q_x = 99.8\%$ (or $p_x = 96.0\%$). Note that, in this case, the X and Z bases are interchangeable due to symmetry.

Finally, we can compute the key length and the corresponding security parameter using our postprocessing procedure and compare with the key length using asymptotic assumptions. The final key length using asymptotic assumptions is

$$K_{\text{asymp}} = n[1 - h_2(e_{bx}) - h_2(e_{bz})], \tag{43}$$

where we used the fact that, asymptotically, the phase error rate in one basis is the same as the bit error rate in the other basis and the use of efficient BB84 leads to always matching basis between Alice and Bob. The key length with asymptotic analysis is 5.15 Mb, and the one with the postprocessing procedure is 4.41 Mb and its failure probability is $\varepsilon = 1.0073 \times 10^{-7}$ (roughly $1 + 2^{-7}$ times the predefined value of 10^{-7}). Furthermore, we can get the trace-distance security parameter using Lemma 1 to conclude that this 4.41 Mb key is composable and is (4.4884×10^{-4}) secure in accordance with security Definition 1. Here, for illustrative purposes, the key length using the postprocessing procedure is calculated with the assumption that $n_x = np_x^2$ and $n_z = n(1 - p_x)^2$.

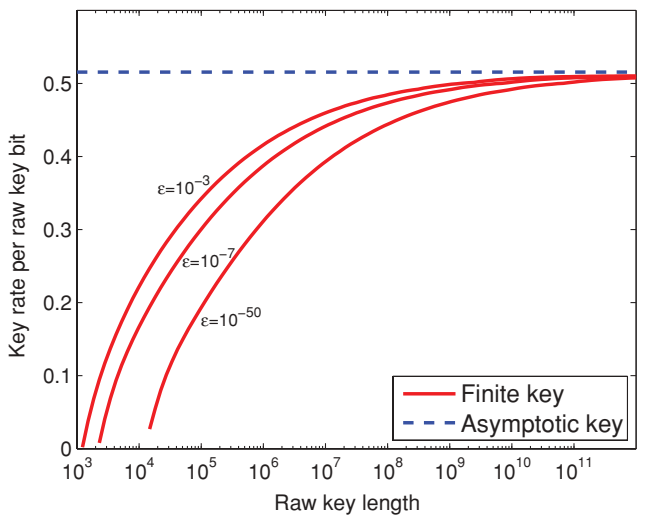

FIG. 2. (Color online) Lower bound for the key rate as a function of the raw key length; parameters used: $e_{bx} = e_{bz} = 4\%$ and the error correction efficiency is 100%. The three curves correspond to three different values of failure probability ε.

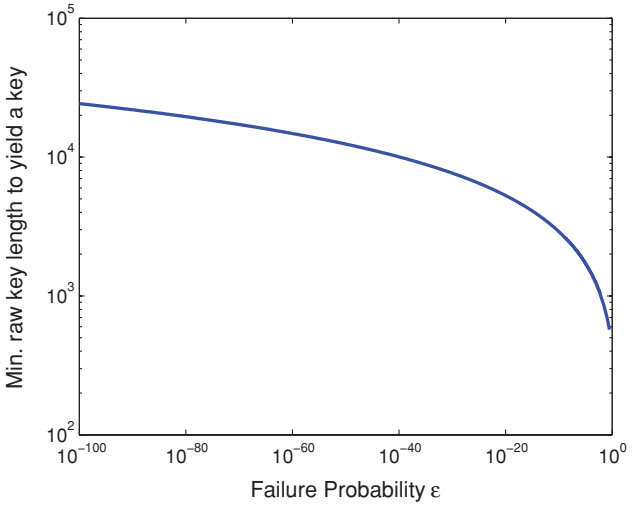

FIG. 3. (Color online) Minimum raw key length to yield a positive key length as a function of ε; parameters used: $e_{bx} = e_{bz} = 4\%$ and the error correction efficiency is 100%.

In the simulation, we assume the error correction efficiency is 100% (Shannon limit). Thus, the difference between the "asymptotic-key" length and the "finite-key" length, 0.74 Mb, comes from the finite statistical analysis. The cost (and the security parameter) due to the finite key analysis mainly comes from the phase error rate estimation. Note that all the remaining cost is only $k_3 = 543$ bit and $\varepsilon_3 = 7.3 \times 10^{-10}$. This point can be clearly seen by comparing Eqs. (21) and (39) in the case of large n. The exponent coefficient in Eq. (21) is $-\frac{\theta^2}{4(1-e_{bx})e_{bx}}$, while in Eq. (39) it is $-\frac{k_3}{5n}$, and also a small change in θ affects the key rate more than that in k_3/n does.

Figure 2 shows the lower bound for the key rate as a function of the raw key length. Note that since we use our simplified

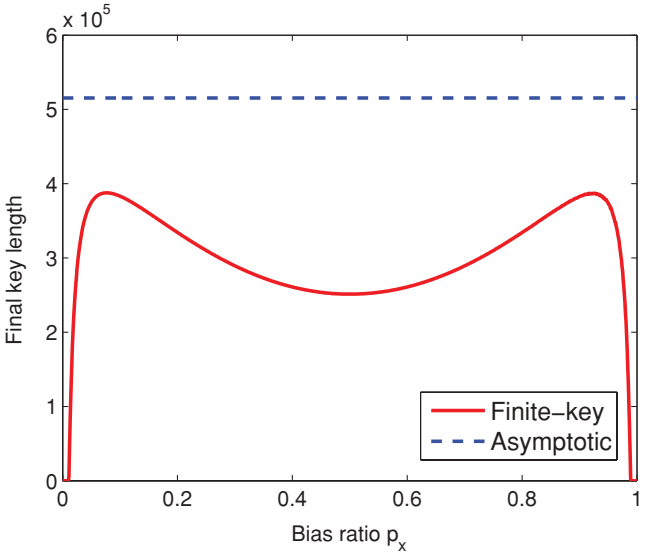

FIG. 4. (Color online) Effect of the bias ratio on the final key length; parameters used: $e_{bx} = e_{bz} = 4\%$, target failure probability $\varepsilon = 10^{-7}$, the raw key length is 10^6, and the error correction efficiency is 100%.

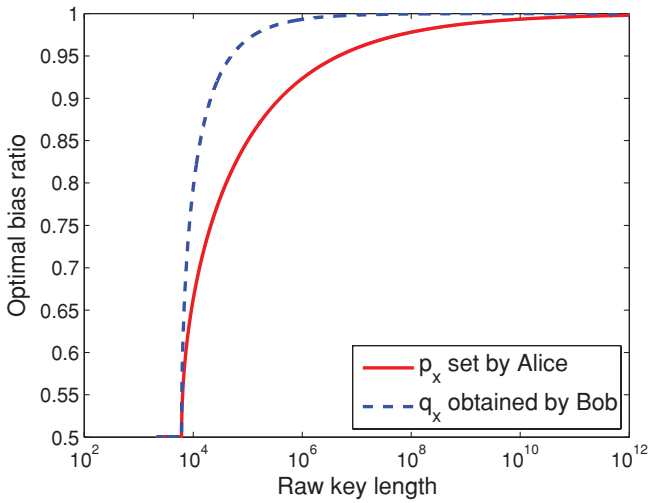

FIG. 5. (Color online) Plot of the optimal bias ratio vs. the raw key length; parameters used: $e_{bx} = e_{bz} = 4\%$, target failure probability $\varepsilon = 10^{-7}$, and the error correction efficiency is 100%.

optimization method, the final security parameter ε for the failure probability deviates slightly from the predefined value. Calculations show that this difference is less than 1% of the predefined values over the entire plotting range for all three curves.

Figure 3 shows the minimum raw key length needed to yield a positive key length as a function of the predefined security parameter ε. In typical applications, a rough security level may be required for a secret key which is to be generated by QKD. Thus, this figure gives the minimum number of signals needed to be detected in order to achieve such a security level.

Figure 4 illustrates the effect of the bias ratio on the final key length. It can be seen that when the optimal bias ratio is used, the final key length increases by over 50% compared to the case when the bias ratio of 0.5 is used. Thus, the bias ratio has a big effect on the key generation performance. Figure 5 shows the optimal bias ratio versus the raw key length. The optimal bias ratio leads to the largest final key length. It can be seen that as the raw key length approaches infinity, the optimal bias ratio tends to one. This makes sense since in the asymptotic case, it is more efficient for Alice and Bob to use one basis with a high probability for key generation in order to avoid wasteful basis mismatch and to use the other basis only for phase error estimation; this is the idea of the efficient BB84 protocol [41]. The optimal bias ratio drops to 0.5 as the raw key length approaches the minimal for positive key generation.

XIII. CONCLUDING REMARKS

In this article, we propose a complete postprocessing procedure for transforming measurement outcomes in a QKD experiment to a final secret key, which we quantify with a security parameter, namely the failure probability of the postprocessing procedure. This failure probability is directly connected to the composability security definition (cf. Lemma 1). Our postprocessing procedure contains many elements, including authentication, the choice of the basis bias ratio, error correction and verification, phase error rate estimation, and privacy amplification. Our procedure results from integrating all these elements with ideas from security proofs. Since the underlying security proofs [8,9,46,47] are secure against the most general attacks, our postprocessing procedure also inherits this important property. Based on our analysis, the main contribution to the finite-size effect comes from the inefficiency of phase error rate estimation, which is a consequence of the random sampling argument for inferring unobserved quantities from observed ones. Further remarks and future directions are listed as follows:

1. In the privacy amplification step, Alice and Bob need a common matrix to generate the final secure key. The current way to construct the matrix is by Alice sending a random bit string to Bob, which requires authenticated classical communication. An alternative way is by each of them generating a matrix with a preshared secret key. Of course, the amount of preshared secret key bits required must be small compared to the generated key length. Also, the failure probability is related to the amount of preshared bits consumed. We leave this investigation for future research. The main advantage of the second method is that no classical communication is needed for the privacy amplification part. In this case, the error verification step can be done either before or after the privacy amplification.
2. In the security proof, the imperfection of X- and Z-basis measurements and efficiency mismatch are not considered. It is interesting to consider the detector efficiency mismatch with the finite key analysis [70].
3. As noted in Ref. [20], the finite-key analysis for the decoy-state QKD is a hard problem. In the decoy-state QKD, the fluctuation comes from not only statistics but also hardware imperfections. The question of interest is where the main contribution of the fluctuation comes from and how to quantify these fluctuations. Since QKD systems with coherent states are most widely used in experiments, investigating the finite key effect in decoy-state QKD is an important step toward a QKD standard.
4. In order to fairly compare our finite-key analysis to others, such as Scarani and Renner [38] and Cai and Scarani [40], one has to make sure the postprocessing elements of different postprocessing procedures carry similar capacities. For example, there are different ways to treat the basis bias ratio, authentication, and random sampling. Therefore, a clear objective must be defined first before making a meaningful comparison. We remark that comparing the performance of various postprocessing procedures as a whole and comparing only the underlying security proofs (which are just one element in a postprocessing procedure) are two different goals. As we have shown in this article, the main contribution to the finite-size effect comes from random sampling in the parameter estimation step. Thus, it may be more interesting in practice to compare different random sampling arguments.
5. Our analysis treats the X basis and Z basis separately, especially when we estimate the phase error rates using the random sampling argument. On the other hand,

one may mix the measurement data of different bases before any analysis. Doing so makes it easy to perform a similar finite-key analysis for other protocols such as the Scarani-Acín-Ribordy-Gisin-2004 (SARG04) protocol [71]. In this case, we can use the Azuma's inequality [72] in place of the random sampling argument to estimate the phase error rate. This is discussed in more detail in Appendix D.

6. In QKD experiments, error correction is often performed in blocks (say, 1 kbit) and privacy amplification is performed on all the blocks together. In some error correction scheme, the failure probability for small blocks is not negligible. That is, after the error correction, some blocks may still have errors, discarding these blocks may have security implication and thus care is required [73]. It is an interesting future topic to give a strict security argument on this issue.
7. Although our analysis uses particular procedures for the steps (e.g., authentication, error correction), our analysis is generic in the sense that each specific procedure may be substituted by another with the same functionality. The new secret-key cost and failure probability will then be used in the analysis of the generation rate and failure probability of the final key.

ACKNOWLEDGMENTS

We thank J.-C. Boileau, C. Erven, N. Godbout, M. Hayashi, D. W. Leung, H.-K. Lo, N. Lütkenhaus, M. Koashi, X. Mo, B. Qi, R. Renner, V. Scarani, D. Stebila, K. Tamaki, W. Tittel, Q. Wang, Y. Zhao and other participants in the workshop *Quantum Works QKD Meeting (Waterloo, Canada)* and *Finite Size Effects in QKD (Singapore)* for enlightening discussions. X. Ma especially thanks H. F. Chau for hospitality and support during his visit at the University of Hong Kong. This work is supported by the NSERC Innovation Platform Quantum Works, the NSERC Discovery grant, the RGC Grant No. HKU 701007P of the HKSAR Government, and the Postdoctoral Grant Program of NSERC.

APPENDIX A: PROOF OF EQ. (4)

First, given that $\langle 0_X^{\otimes m}|\rho_A|0_X^{\otimes m}\rangle = \alpha$, the purification $|\Psi_{AE}\rangle$ of ρ_A is of the general form

$$|\Psi_{AE}\rangle = \sqrt{\alpha}\left|0_X^{\otimes m}\right\rangle_A|0\rangle_E + \sqrt{1-\alpha}|\Psi^{\perp}\rangle_{AE}, \tag{A1}$$

where $|\Psi^{\perp}\rangle_{AE}$ has unit norm and ${}_A\langle 0_X^{\otimes m}|\Psi^{\perp}\rangle_{AE} = 0$. Thus, the fidelity in question is

$$\begin{aligned} &F\left(\rho_{AE}, \left|0_X^{\otimes m}\right\rangle_A\left\langle 0_X^{\otimes m}\right| \otimes \rho_E\right) \\ &= \mathrm{Tr}\sqrt{|\Psi_{AE}\rangle\langle\Psi_{AE}|\left[\left|0_X^{\otimes m}\right\rangle_A\left\langle 0_X^{\otimes m}\right| \otimes \rho_E\right]|\Psi_{AE}\rangle\langle\Psi_{AE}|} \\ &= \mathrm{Tr}\sqrt{|\Psi_{AE}\rangle[\alpha\langle 0|\rho_E|0\rangle]\langle\Psi_{AE}|} = \sqrt{\alpha\langle 0|\rho_E|0\rangle}, \end{aligned} \tag{A2}$$

where $\rho_{AE} = |\Psi_{AE}\rangle\langle\Psi_{AE}| = \rho_{AE}^{1/2}$. Since $\rho_E = \mathrm{Tr}_A(\rho_{AE})$ and

$$\langle 0|\rho_E|0\rangle = \sum_i |\langle\Psi_{AE}|[|i\rangle_A|0\rangle_E]|^2, \tag{A3}$$

where the summation is over all vectors of a basis in system A, by considering a basis having $|0_X^{\otimes m}\rangle_A$ as its basis state, we get $\langle 0|\rho_E|0\rangle \geqslant \alpha$. Substituting this into Eq. (A2), we get Eq. (4).

APPENDIX B: EVALUATION OF HYPERGEOMETRIC FUNCTION

In this appendix, we will evaluate the hypergeometric function

$$P_\theta \leqslant \Pr\{k|m,n,N\} = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}, \tag{B1}$$

with $k = e_{bx}n_x$, $N = n_x + n_z$, $n = n_x$, and $m = e_{bx}(n_x + n_z) + \theta n_z$. Here, we relabel the function for simplicity. Strictly speaking, θ is a discrete variable with a minimum quantum of $1/n_z$, which will keep m to be an integer.

In the following discussion, we assume the integers $N > m > k \geqslant 1$ and $N > n > k$. The only exception that could (though highly unlikely) happen in the realistic case is $k = 0$. In this case, for a given m, $P_\theta(k=0) < P_\theta(k=1)$. Now that we only care about the upper bound of the probability, we can always safely replace $k = 0$ with $k = 1$ in the calculation.

We simplify the hypergeometric function by the Stirling formula [74]

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\lambda_n}, \tag{B2}$$

where

$$\frac{1}{12n+1} < \lambda_n < \frac{1}{12n}. \tag{B3}$$

Then, Eq. (B1) can be expressed as

$$\begin{aligned} P_\theta &\leqslant \frac{\binom{n}{k}\binom{N-n}{m-k}}{\binom{N}{m}} \\ &= \frac{n!(N-n)!(N-m)!m!}{k!(n-k)!(m-k)!(N-n-m+k)!N!} \\ &= \frac{1}{\sqrt{2\pi}}\frac{\sqrt{n}\sqrt{N-n}\sqrt{N-m}\sqrt{m}}{\sqrt{k}\sqrt{n-k}\sqrt{m-k}\sqrt{N-n-m+k}\sqrt{N}} \\ &\quad\cdot\frac{n^n(N-n)^{N-n}(N-m)^{N-m}m^m}{k^k(n-k)^{n-k}(m-k)^{m-k}} \\ &\quad\cdot\frac{1}{(N-n-m+k)^{N-n-m+k}N^N} \\ &\quad\cdot\exp(\lambda_n+\lambda_{N-n}+\lambda_{N-m}+\lambda_m-\lambda_k \\ &\quad-\lambda_{n-k}-\lambda_{m-k}-\lambda_{N-n-m+k}-\lambda_N). \end{aligned} \tag{B4}$$

First, we can prove that

$$\begin{aligned} &\lambda_n+\lambda_{N-n}+\lambda_{N-m}+\lambda_m-\lambda_k \\ &\quad-\lambda_{n-k}-\lambda_{m-k}-\lambda_{N-n-m+k}-\lambda_N < 0 \end{aligned} \tag{B5}$$

with the facts of $m > k \geqslant 1$, $n-k > 1$ and Eq. (B3). Remark: though the left-hand side of Eq. (B5) is negative, it is close to 0 in the order of $O(1/12k)$.

Second, we know that $1/\sqrt{x(1-x)}$ is a decreasing function for $0 < x < 1/2$. Then we can easily see that

$$\begin{aligned}&\frac{\sqrt{n}\sqrt{N-n}\sqrt{N-m}\sqrt{m}}{\sqrt{k}\sqrt{n-k}\sqrt{m-k}\sqrt{N-n-m+k}\sqrt{N}}\\ &\quad\leqslant \frac{\sqrt{N}}{\sqrt{n(N-n)}}\frac{1}{\sqrt{e_{bx}(1-e_{bx})}}\\ &\quad= \frac{1}{\sqrt{N}}\frac{1}{\sqrt{q_x(1-q_x)e_{bx}(1-e_{bx})}}\end{aligned} \tag{B6}$$

with the facts of $e_{bx} = k/n$ and $e_{pz} = (m-k)/(N-n) \geqslant e_{bx}$. Remark: when $e_{pz} = e_{bx}$, the equality holds. From this point of view, the bound is tight.

Third, the remaining term of the failure probability can be expressed by

$$\begin{aligned}&\frac{n^n(N-n)^{N-n}(N-m)^{N-m}m^m}{k^k(n-k)^{n-k}(m-k)^{m-k}(N-n-m+k)^{N-n-m+k}N^N}\\ &\quad= 2^{nH(\frac{k}{n})+(N-n)H(\frac{m-k}{N-n})-NH(\frac{m}{N})}\\ &\quad\equiv 2^{-N\xi_x(\theta)},\end{aligned} \tag{B7}$$

where we use the definition of the binary entropy function $H(x)$. The exponent coefficient is given by

$$\begin{aligned}\xi_x(\theta) \equiv\ & H(e_{bx}+\theta-q_x\theta) - q_xH(e_{bx})\\ &-(1-q_x)H(e_{bx}+\theta)\end{aligned} \tag{B8}$$

with $q_x = n/N$ and $(m-k)/(N-n) = e_{bx}+\theta$. Due to the concavity of $H(x)$, $\xi_x(\theta)$ is negative for $\theta > 0$ and $0 < q_x < 1$.

Therefore, by combining Eqs. (B4), (B5), (B6), and (B7), the failure probability of Eq. (18) is given by

$$P_\theta < \frac{1}{\sqrt{N}}\frac{1}{\sqrt{q_x(1-q_x)e_{bx}(1-e_{bx})}}2^{-N\xi_x(\theta)}, \tag{B9}$$

where $\xi_x(\theta)$ is given by Eq. (B8). Note that $\xi_x(\theta)$ is independent of key size N given the error rates and bias ratio. Now we can see that the failure probability decreases (actually, slightly faster than) exponentially with N.

APPENDIX C: PROOF OF EQ. (29)

We prove Eq. (29) by the following claim.

Claim 1.

$$\sum_{k=0}^{m-1}\binom{n}{k} < \binom{n}{m} \tag{C1}$$

when $m \leqslant n/3$.

Proof. First notice that

$$\frac{\binom{n}{k-1}}{\binom{n}{k}} = \frac{k}{n-k+1} < \frac{1}{2} \tag{C2}$$

is true for all $k \leqslant n/3$. Thus,

$$\sum_{k=0}^{m-1}\binom{n}{k} \leqslant \sum_{k=0}^{m-1}2^{k-m}\binom{n}{m} = \binom{n}{m}\sum_{k=0}^{m-1}2^{k-m} < \binom{n}{m} \tag{C3}$$

is true for $m \leqslant n/3$.

APPENDIX D: ESTIMATION OF PHASE ERROR RATE FOR MIXED-BASIS ANALYSIS

The analysis in the main part of the article treats each of the two bases separately when estimating the phase error rates for them. This is possible in BB84, since the phase errors in one basis are the bit errors in the other basis. And in this case, a random sampling argument suffices to establish some confidence on the unmeasured phase error rate in one basis from the measured bit error rate in the other basis. On the other hand, one may mix all the measurement data in the different bases together before applying any further analysis. This can be done in BB84. For other protocols, this mixing actually leads to a simpler analysis and thus is favorable. Here, we describe how to estimate the phase error rate for the mixed-basis case. When the measurements are mixed, protocols can usually be characterized with a relation between the bit and phase error probabilities $p_p = \alpha p_b$, where $\alpha \geqslant 1$ in general (e.g., $\alpha = 3/2$ for SARG04 [75,76] and $\alpha = 5/4$ for a three-state protocol [77]). (Note that here the error probabilities are the combined values of all bases and thus do not carry a basis designation.) Given such a relation in probabilities, we want to establish a similar relation for the error rates and compute the confidence for it. A useful tool to do this is the Azuma's inequality [72] (see also Refs. [75–78] for the application of it to security proofs), which relates the sum of conditional probabilities to the total number of a particular outcome in many trials. To start, we relate the probability and the rate for the bit error and the phase error separately using the Azuma's inequality as follows:

$$\Pr\{|p_b - e_b| \geqslant \varepsilon_{\mathrm{Az}}\} \leqslant 2\exp\left(\frac{-n\varepsilon_{\mathrm{Az}}^2}{2}\right) \tag{D1}$$

$$\Pr\{|p_p - e_p| \geqslant \varepsilon_{\mathrm{Az}}\} \leqslant 2\exp\left(\frac{-n\varepsilon_{\mathrm{Az}}^2}{2}\right) \tag{D2}$$

where $p_{b,p}$ and $e_{b,p}$ designate the error probabilities and the error rates, respectively, $\varepsilon_{\mathrm{Az}}$ represents a failure probability, and n is the number of measurements made. Because $p_p = \alpha p_b$, we multiple these two inequalities to get the relation between the bit and phase error rates:

$$\Pr\{|e_p - \alpha e_b| \geqslant (1+\alpha)\varepsilon_{\mathrm{Az}}\} \leqslant 4\exp\left(-n\varepsilon_{\mathrm{Az}}^2\right). \tag{D3}$$

For BB84, $\alpha = 1$ and this bound is worse than the random sampling result [cf. Eq. (23)] in typical situations.

[1] C. H. Bennett and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing* (IEEE, New York/Bangalore, 1984), pp. 175.

[2] C. H. Bennett, G. Brassard, and N. D. Mermin, Phys. Rev. Lett. **68**, 557 (1992).

[3] C. H. Bennett, Phys. Rev. Lett. **68**, 3121 (1992).

[4] D. Bruss, Phys. Rev. Lett. **81**, 3018 (1998).
[5] H. Bechmann-Pasquinucci and N. Gisin, Phys. Rev. A **59**, 4238 (1999).
[6] D. Mayers, in *Advances in Crytology-Crypto '96, Lecture Notes in Computer Science* (Springer, Berlin, 1996), Vol. 1109, p. 343.
[7] D. Mayers, J. ACM **48**, 351 (2001).
[8] H.-K. Lo and H. F. Chau, Science **283**, 2050 (1999).
[9] P. W. Shor and J. Preskill, Phys. Rev. Lett. **85**, 441 (2000).
[10] H.-K. Lo, Quantum Inf. Comput. **1**, 81 (2001).
[11] K. Tamaki, M. Koashi, and N. Imoto, Phys. Rev. Lett. **90**, 167904 (2003).
[12] R. Renner, Ph.D. thesis, Swiss Federal Institute of Technology (2005);also available in Int. J. Quantum. Inform. **6**, 1 (2008).
[13] A. K. Ekert, Phys. Rev. Lett. **67**, 661 (1991).
[14] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, Phys. Rev. Lett. **98**, 230501 (2007).
[15] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. **74**, 145 (2002).
[16] H.-K. Lo and N. Lütkenhaus, Phys. Canada **63**, 191 (2007).
[17] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dusek, N. Lütkenhaus, and M. Peev, Rev. Mod. Phys. **81**, 1301 (2009).
[18] W.-Y. Hwang, Phys. Rev. Lett. **91**, 057901 (2003).
[19] H.-K. Lo, X. Ma, and K. Chen, Phys. Rev. Lett. **94**, 230504 (2005).
[20] X. Ma, B. Qi, Y. Zhao, and H.-K. Lo, Phys. Rev. A **72**, 012326 (2005).
[21] X.-B. Wang, Phys. Rev. Lett. **94**, 230503 (2005).
[22] X.-B. Wang, Phys. Rev. A **72**, 012322 (2005).
[23] J. W. Harrington, J. M. Ettinger, R. J. Hughes, and J. E. Nordholt, e-print arXiv:quant-ph/0503002 (2005).
[24] Y. Zhao, B. Qi, X. Ma, H.-K. Lo, and L. Qian, Phys. Rev. Lett. **96**, 070502 (2006).
[25] Y. Zhao, B. Qi, X. Ma, H.-K. Lo, and L. Qian, in *Proc. of IEEE ISIT* (IEEE, 2006), 2094 .
[26] K. Tamaki and N. Lütkenhaus, Phys. Rev. A **69**, 032316 (2004).
[27] M. Koashi, Phys. Rev. Lett. **93**, 120501 (2004).
[28] K. Tamaki, N. Lütkenhaus, M. Koashi, and J. Batuwantudawe, Phys. Rev. A **80**, 032302 (2009).
[29] K. Tamaki, Phys. Rev. A **77**, 032341 (2008).
[30] K. Inoue, E. Waks, and Y. Yamamoto, Phys. Rev. Lett. **89**, 037902 (2002).
[31] K. Inoue, E. Waks, and Y. Yamamoto, Phys. Rev. A **68**, 022317 (2003).
[32] H. Takesue, E. Diamanti, C. Langrock, M. M. Fejer, and Y. Yamamoto, Opt. Express **14**, 9522 (2006).
[33] H. Takesue, E. Diamanti, T. Honjo, C. Langrock, M. M. Fejer, K. Inoue, and Y. Yamamoto, New J. Phys. **7**, 232 (2005).
[34] M. Koashi and J. Preskill, Phys. Rev. Lett. **90**, 057902 (2003).
[35] X. Ma, C.-H. F. Fung, and H.-K. Lo, Phys. Rev. A **76**, 012307 (2007).
[36] N. Lütkenhaus, Phys. Rev. A **59**, 3301 (1999).
[37] M. Hayashi, Phys. Rev. A **74**, 022307 (2006).
[38] V. Scarani and R. Renner, Phys. Rev. Lett. **100**, 200501 (2008).
[39] V. Scarani and R. Renner, Lect. Notes Comput. Sci. **5106**, 83 (2008).
[40] R. Y. Cai and V. Scarani, New J. Phys. **11**, 045024 (2009).
[41] H.-K. Lo, H. F. Chau, and M. Ardehali, J. Cryptology **18**, 133 (2005).
[42] X. Ma, C.-H. F. Fung, J.-C. Boileau, and H. F. Chau, e-print arXiv:0904.1994 (2009).
[43] T. Tsurumaru and K. Tamaki, Phys. Rev. A **78**, 032302 (2008).
[44] N. J. Beaudry, T. Moroder, and N. Lütkenhaus, Phys. Rev. Lett. **101**, 093601 (2008).
[45] M. Koashi, e-print arXiv:quant-ph/0609180 (2006).
[46] H.-K. Lo, New J. Phys. **5**, 36 (2003).
[47] M. Koashi, J. Phys. Conf. Ser. **36**, 98 (2006).
[48] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[49] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997.
[50] M. Ben-Or, M. Horodecki, D. W. Leung, D. Mayers, and J. Oppenheim, in *Second Theory of Cryptography Conference TCC 2005, Lecture Notes in Computer Science* (Springer-Verlag, 2005), Vol. 3378, p. 386 .
[51] R. Renner and R. König, in *Second Theory of Cryptography Conference TCC 2005, Lecture Notes in Computer Science* (Springer-Verlag, 2005), Vol. 3378, p. 407 .
[52] R. Canetti, Tech. Rep. TR01-016, Electronic Colloquium on Computational Complexity (ECCC) (2001), preliminary version in IEEE Symposium on Foundations of Computer Science, pp. 136, 2001.
[53] R. Canetti and H. Krawczyk, in *EUROCRYPT 2002: Lecture Notes in Computer Science* (Springer-Verlag, New York, 2002), Vol. 2332, p. 337 .
[54] M. Ben-Or and D. Mayers, e-print arXiv:quant-ph/0409062 (2004).
[55] R. König, R. Renner, A. Bariska, and U. Maurer, Phys. Rev. Lett. **98**, 140502 (2007).
[56] C. A. Fuchs and J. van de Graaf, IEEE Trans. Inf. Theory **45**, 1216 (1999).
[57] X. Ma, T. Moroder, and N. Lütkenhaus, e-print arXiv:0812.4301 (2008).
[58] This is a slightly modified version of the flow chart proposed by Wolfgang Tittel in the workshop *Quantum Works QKD Meeting* (Waterloo, Canada, 2008).
[59] M. N. Wegman and J. L. Carter, J. Comput. Syst. Sci. **18**, 143 (1979).
[60] M. N. Wegman and J. L. Carter, J. Comput. Syst. Sci. **22**, 265 (1981).
[61] H. Krawczyk, in *Advances in Cryptology—CRYPTO'94, Lecture Notes in Computer Science* (Springer-Verlag, New York, 1994), Vol. 893, p. 129.
[62] H. Krawczyk, in *Advances in Cryptology—EUROCRYPT'95* (Springer-Verlag, New York, 1995), Vol. 921, p. 301.
[63] D. Gottesman and H.-K. Lo, IEEE Trans. Inf. Theory **49**, 457 (2003).
[64] X. Ma, C.-H. F. Fung, F. Dupuis, K. Chen, K. Tamaki, and H.-K. Lo, Phys. Rev. A **74**, 032330 (2006).
[65] D. Gottesman, H.-K. Lo, N. Lütkenhaus, and J. Preskill, Quantum Inf. Comput. **4**, 325 (2004).
[66] Y. Mansour, N. Nisan, and P. Tiwari, in *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing* (ACM, New York, 1990), p. 235.
[67] M. Hayashi, Phys. Rev. A **76**, 012329 (2007).
[68] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. (Wiley, New York, 1968).

[69] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. (Wiley Interscience, New York, 2006).
[70] C.-H. F. Fung, K. Tamaki, B. Qi, H.-K. Lo, and X. Ma, Quantum Inf. Comput. **9**, 0131 (2009).
[71] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, Phys. Rev. Lett. **92**, 057901 (2004).
[72] K. Azuma, Tôhoku Math. J. **19**, 357 (1967).
[73] This issue was raised by Wolfgang Tittel.
[74] H. Robbins, Am. Math. Mon. **62**, 26 (1955).
[75] K. Tamaki and H.-K. Lo, Phys. Rev. A **73**, 010302(R) (2006).
[76] C.-H. F. Fung, K. Tamaki, and H.-K. Lo, Phys. Rev. A **73**, 012337 (2006).
[77] J.-C. Boileau, K. Tamaki, J. Batuwantudawe, R. Laflamme, and J. M. Renes, Phys. Rev. Lett. **94**, 040503 (2005).
[78] C.-H. F. Fung and H.-K. Lo, Phys. Rev. A **74**, 042342 (2006).