Guessing probability in quantum key distribution

Xiang-Bin Wang\textsuperscript{1,2,4,5,\dagger}, Jing-Tao Wang\textsuperscript{1}, Ji-Qian Qin\textsuperscript{1}, Cong Jiang\textsuperscript{1}, Zong-Wen Yu\textsuperscript{1,3}
\textsuperscript{1}State Key Laboratory of Low Dimensional Quantum Physics, Department of Physics, Tsinghua University, Beijing 100084, People’s Republic of China
\textsuperscript{2}Synergetic Innovation Center of Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei, Anhui 230026, China
\textsuperscript{3}Data Communication Science and Technology Research Institute, Beijing 100191, China
\textsuperscript{4}Jinan Institute of Quantum technology, SAICT, Jinan 250101, Peoples Republic of China
\textsuperscript{5}Department of Physics, Southern University of Science and Technology, Shenzhen, 518055, Peoples Republic of China

Based on the existing trace distance result, we present a simple and efficient method to tighten the upper bound of the guessing probability. The guessing probability of final key $k$ can be upper bounded by the guessing probability of another key $k'$, if $k'$ can be mapped from final key $k$. Comparing with the known methods, our result is more tightened by thousands of orders of magnitude. For example, given a $10^{-9}$-secure key from the sifted key, the upper bound of the guessing probability is $2 \times 10^{-3277}$ using our method. This value is smaller than the existing result $10^{-9}$ by more than 3000 orders of magnitude. Our result shows that, in the viewpoint of guessing probability, the performance of the existing trace distance security is actually much better than what was assumed in the past.

I. Introduction

The first Quantum Key Distribution (QKD) protocol was proposed by Bennett and Brassard in 1984, based on the fundamentals of quantum mechanics \cite{Bennett1984}. Since then, the security of QKD has always been the central issue in the quantum cryptographic field \cite{Duong1996}. Trace distance is a very important security criterion \cite{Duong1996, Wang2006}. It provides the universal composable security \cite{Makarov2012}, which can guarantee the security of key whatever its application may be, such as one-time pad (OTP). This is why many studies choose trace distance for security criterion \cite{Xiang2015, Wang2006, Duong1996, Makarov2012}.

In a classical practical cryptosystem, the impact of guessing probability on security is very important \cite{Makarov2012, Wang2006}. Especially, the key generated by QKD protocol is not based on the presumed hardness of mathematical problems, so the eavesdropper Eve can only guess the final key via the measurement result of her probe. The guessing probability intuitively describes the probability that Eve can correctly guesses the final key, which can reflect the number of guesses that Eve needs to get the correct final key.

There are few studies on the guessing probability of QKD. Since there are more regorous security measure such as the trace distance \cite{Makarov2012, Wang2006}, which gives the composable security. This makes the crucially important theoretical foundation for security of QKD. However, in the real application of QKD projects, customers often ask the question of guessing probability. The existing prior art results cannot give them a satisfactory upper bound \cite{Makarov2012}. Consequently, some people questioned the security of QKD by relying on the prior art results of guessing probability \cite{Wang2006}. For example, according to the existing result \cite{Makarov2012}, the guessing probability of an $\varepsilon$-secure key is about $10^{-9}$ if $\varepsilon$ is around $10^{-9}$. Although the final key is far secure than the 30 bits of perfect key, it can be easily mistaken for that there are only 30 bits of security in QKD system \cite{Wang2006}, just because they have the same guessing probability. In practice, it is not unusual that we need the guessing probability to be much smaller such as $10^{-100}$ or $10^{-1000}$. Therefore, it is extremely important to find a more tightened upper bound of guessing probability. In this letter, starting from the composable security through the trace distance \cite{Duong1996}, we give a very efficient and simple method to evaluate the guessing probability. Typically, compared with those previous results, the method proposed here can reduce by hundreds or even thousands orders of magnitude for the upper bound value of the guessing probability.

II. The guessing probability

We consider the security definitions of a practical QKD protocol with finite-size under the framework of composable security \cite{Makarov2012, Wang2006, Duong1996, Wang2006}. Suppose that Alice and Bob get two $N$-bit sifted key strings, $s$ and $s'$. By performing an error correction and private amplification scheme, Alice gets a $n_1$-bit key $k$ and Bob gets an estimate key $\hat{k}$ of $k$ from $s$ and $s'$. The protocol is $\epsilon_{\text{cor}}$-correct if $P(k \neq \hat{k}) \leq \epsilon_{\text{cor}}$. In general, the key $k$ of Alice can be correlated with an eavesdropper system and the density matrix of Alice and Eve is $\rho_{AE}$. The protocol outputs an $\varepsilon$-secure key \cite{Wang2006}, if

\[
\frac{1}{2} \left\| \rho_{AE} - \rho_U \otimes \rho_E \right\|_1 \leq \varepsilon, \quad (1)
\]

where $\| \cdot \|_1$ denotes the trace norm, $\rho_U$ is the fully mixed state of Alice’s system. The protocol is $\epsilon_{\text{tot}}$-secure if $\epsilon_{\text{cor}}$ and $\varepsilon$ satisfies $\epsilon_{\text{cor}} + \varepsilon \leq \epsilon_{\text{tot}}$ which means it is $\epsilon_{\text{tot}}$-indistinguishable from a perfect protocol (which is correct and secret). Without any loss of generality, we consider the case of $\epsilon_{\text{cor}} = \varepsilon$ in this article.
We define the security level:

**Definition 1** If key \( k \) is \( \varepsilon \)-secure, we also say the security level of key \( k \) is \( \varepsilon \).

For symbol clarity, we shall use notation \( \varepsilon_k \) for the security level of key \( k \). With this definition, we shall say key \( k \) is \( \varepsilon_k \)-secure, or its security level is \( \varepsilon_k \).

We define the guessing probability:

**Definition 2** Let the final key generated by QKD protocol be \( k \), the guessing probability of \( k \) is defined as the success probability of the attacker Eve guessing the final key via her measurement result, and is denoted as \( p(k) \).

**Lemma 1** The guessing probability of an \( \varepsilon_k \)-secure key \( k \) with length \( n_1 \) is not larger than \( \frac{1}{4^n} + \varepsilon_k \).

This is a conclusion from Ref. [11]. The proof has been given in Ref. [11] already, for readers convenience, we write the proof again in the Appendix A.

According to Lemma 1, the guessing probability of key \( k \) can be divided into two parts and one part \( 2^{-n_1} \) is related to the length of the key, the other part \( \varepsilon_k(n_1) \) is related to the security level. Under the framework of universally composable security, when calculating the final key length, we often make the security level between \( 10^{-9} \sim 10^{-24} \) which is much bigger than \( 2^{-n_1} \), because \( n_1 \) is often \( 10^3, 10^4 \) or more larger. Therefore, \( 2^{-n_1} \) can be ignored and \( p(k) \leq \bar{p}(k) \sim O(\varepsilon(k)) \). In that case, however, the guessing probability of a secure key with a length of tens of bits can also reach this magnitude. Therefore, when the secure requirements are very high, it is obviously not enough for a key with a length of thousands or even longer if the upper bound of guessing probability only stops at this magnitude. Therefore, we cannot simply use this formula alone to get the upper bound of the guessing probability. Fortunately, we have a much better way for tightening the bound. That is what we will present in the following.

**Lemma 2** If key \( k \) can be mapped to string \( k' \) by a map \( M \) that is known to Eve, then the guessing probability of \( k \) cannot be larger than the guessing probability of string \( k' \) i.e.,

\[
p(k) \leq p(k').
\]

Here \( p(k), p(k') \) are guessing probabilities of \( k \) and \( k' \), respectively.

**Proof.** This lemma is obvious, just because when Eve can correctly guess the key, Eve can obtain the \( k' \) definitely due to knowing the map \( M \). Otherwise, Eve can still correctly guess the \( k' \) with a probability not less than 0, i.e. \( p(k') = p(k) + \delta, \delta \geq 0 \).

**Theorem 1:** If an \( \varepsilon_k \)-secure key \( k \) with length \( n_1 \) can be mapped to an \( \varepsilon_{k'} \)-secure key \( k' \) with length \( n_2 \), the guessing probability of \( k \) cannot be larger than \( k' \), i.e.

\[
p(k) \leq \bar{p}(k') = \frac{1}{2^{n_2}} + \varepsilon_{k'}.
\]

**Proof.** This theorem actually requests two conditions: i) the final key \( k \) can be mapped to the string \( k' \), ii) the string \( k' \) can be regarded as a \( \varepsilon_{k'} \)-secure key.

With these two conditions, the proof is very simple. Given the condition i), we can apply Lemma 2 to obtain

\[
p(k) \leq p(k').
\]

Given the condition ii), we can apply Lemma 1 to obtain

\[
p(k') \leq \bar{p}(k') = \frac{1}{2^{n_2}} + \varepsilon_{k'},
\]

And according to Eqs. (4) and (5), we can obtain

\[
p(k) \leq \bar{p}(k') = \frac{1}{2^{n_2}} + \varepsilon_{k'}.
\]

This ends our proof of Theorem 1.

As discussed above if the length of the final key \( k \) and the string \( k' \) is very large, then \( 2^{-n_1} \) and \( 2^{-n_2} \) can be ignored. Meanwhile, when \( n_2 < n_1 \), we have \( \varepsilon_{k'} < \varepsilon_k \). Thus, \( 2^{-n_2} + \varepsilon_{k'} < \varepsilon_k \) \( \sim 2^{-n_1} + \varepsilon_k \). That is, Theorem 1 gives a more tightened upper bound of guessing probability.

**III. Method** With Theorem 1, it is now possible for us to get the upper bound of the guessing probability of an \( \varepsilon_k \)-secure key \( k \) much tightly. Instead of directly applying Lemma 1, we choose to first map \( k \) to a \( n_2 \)-bit string \( k' = M(k) \). If string \( k' \) itself can be regarded as an \( \varepsilon_{k'} \)-secure final key, we can apply Theorem 1 now through calculating \( \bar{p}(k') \). And we can obtain much smaller upper bound of the guessing probability of \( k \) if \( \varepsilon_{k'} \) is very small and \( n_2 \) is not too small. Now the problems remaining for us is what is the map \( M \), and how to make sure that \( k' = M(k) \) is another key which is \( \varepsilon_{k'} \)-secure, and how to calculate \( \varepsilon_{k'} \). We start our method with the hashing function in the key distillation.

**Our hashing function.** We take the key distillation with the random matrix. Denote \( R_n \) as the \( n \times N \) random matrix with each element being either 0 or 1, randomly. Also, we represent the \( N \)-bit sifted string \( s \) by a column vector \( S \) which contains \( N \) elements. To obtain a \( n \)-bit final key, we simply take the calculation \( R_n S \). As one may check it easily here that our random matrix is a class of two-universal hashing function family [2].

Suppose we have distilled out the \( n_1 \)-bit key \( k \) from the \( N \)-bit sifted key \( s \) through hashing by our random matrix \( R_n \). We can map the \( n_1 \)-bit key \( k \) into the \( n_2 \)-bit string \( k' = M(k) \) simply by deleting the last \( n_1 - n_2 \) bits from key string \( k \). Obviously, this string \( k' \) mapped from \( k \) can also be regarded as another final key distilled from sifted key \( S \) by the \( n_2 \times N \) random hashing matrix \( R_{n_2} \), which is a submatrix of \( R_n \). In summary, we have

\[
k' = M(k) = R_{n_2} S.
\]

This means, on the one hand, \( k' \) is a string mapped from key \( k \). And on the other hand, \( k' \) can be regarded as another final key of length \( n_2 \) distilled from sifted key \( s \). Since the two conditions in Theorem 1 are satisfied, according to Theorem 1, we can get a tightened upper
bound of $p(k)$ with Eq. (4), if we know the security level of key $k'$, i.e., the value of $\varepsilon_{k'}$. Since our random matrix is a class of two-universal hashing function, the value $\varepsilon_{k'}$ is dependent on $n_2$. Details are shown in the Appendix B on the calculation of $\varepsilon_{k'}$ given $n_2$. Hence, in a QKD protocol using random hashing matrix presented here, to get the upper bound of the guessing probability of the $n_1$-bit final key $k$, we can summarize the procedure above by the following scheme:

**Scheme** 1) Given the $n_1$-bit final key $k$, we delete its last $n_1 - n_2$ bits and obtain string $k'$. 2) We regard $k'$ as another possible final key which is $\varepsilon_{k'}$-secure. Compute the $\varepsilon_{k'}$ value of $k'$ with the input parameters $N$ and $n_2$. 3) Calculate $\bar{p}(k)$ by Theorem 1 through Eq. (3).

Since in our scheme, the value of $\varepsilon_{k'}$ is dependent on $n_2$ as shown in the Appendix B, we can now replace $\varepsilon_{k'}$ by a functional form, $\varepsilon_{k'}(n_2)$. To obtain a tightened upper bound value of the guessing probability in scheme 1, we need to choose an appropriate $n_2$ value. In our calculation, we set the condition

$$2^{-n_2} = \varepsilon_{k'}(n_2),$$

for the appropriate $n_2$.

On the one hand, for any $n > n_2$, we have $\varepsilon_k(n) > \varepsilon_{k'}(n_2) = 2^{-n_2}$; on the other hand, for any $n < n_2$, we have $2^{-n} > 2^{-n_2}$. In conclusion, if $n \neq n_2$, $2^{-n} + \varepsilon_k(n) > 2^{-n_2}$. Therefore, in this study, we set $2^{-n_2} = \varepsilon_{k'}(n_2)$, and get a tightened guessing probability $2^{-n_2+1}$.

Once we find the value $n_2$ and the corresponding $\varepsilon_{k'}(n_2)$, we calculate $\bar{p}(k')$ by Eq. (3). Obviously, this is the upper bound of guessing probability of all those final key $k$ of length $n_1$ provided that

$$n_1 > n_2.$$  

Thus we can actually use a more efficient scheme to get the upper bound of the guessing probability of key $k$, as the following Theorem 2:

**Theorem 2:** In a QKD protocol, if the $n_1$-bit final key $k$ is distilled from the sifted key $s$ by using random matrix $R_{n_1,N}$, the guessing probability of $k$ can be upper bounded by

$$p(k) \leq \bar{p}(k') = 2^{-(n_2-1)},$$

where $k' = M(k) = R_{n_2,N}S$ and $n_2$ satisfies $2^{-n_2} = \varepsilon_{k'}(n_2)$, $n_2 < n_1$.

There are two important points need to be noticed. The first one is when applying our theorem to obtain the non-trivial upper bound of the guessing probability for the final key $k$, we do not really need to map $k$ to another string $k'$ and we only need the existence of a map that can map $k$ to $k'$. As shown above the existence has been proven. The second one is that, in this letter, we use the random matrix $R_{n_1,N}$ as a family of two-universal hash function to distill the key for illustrating our conclusion more intuitively. Of course, we can also use the modified toepitiz matrix $[8]$ instead of random matrix $R_{n_1,N}$. In that case, the final key $k$ can also be mapped to the string $k'$ and the string $k'$ can also be regarded as a $\varepsilon_{k'}$-secure key. It means that the proposed theorem in this study still holds. Therefore, the flow chart of our method of bounding the guessing probability can be shown as Fig. 1.

**FIG. 1:** Flow chart of our method of bounding the guessing probability.

| $N_{tol}$ | $10^4$ | $10^5$ | $10^6$ |
|-----------|--------|--------|--------|
| $n$       | $2.01 \times 10^3$ | $4.06 \times 10^4$ | $4.90 \times 10^5$ |
| $P_{guessing}$ | $10^{-6}$ | $10^{-6}$ | $10^{-6}$ |
| $P_{guessing}$ | $10^{-9}$ | $10^{-9}$ | $10^{-9}$ |
| $P_{Thm.2}$ | $2 \times 10^{-32}$ | $2 \times 10^{-327}$ | $2 \times 10^{-327}$ |

**TABLE I:** Comparison of the guessing probability in the case of $Q_{tol} = 2.14\%$ is the channel error tolerance, $N_s = 0.22N_{tol}$ is the length of string used to do parameter estimation, $N_{tol}$ is the total length of sifted key, $N = 0.78N_{tol}$ is the length of the string for key generation, $\varepsilon = 10^{-9}$ is the security level, $n$ is the length of $10^{-9}$-secure key, and $p_{guessing}$ is the probability that correctly guess the final key. In particular, $P_{Thm.2}$ is the result of Theorem 2 of this work.

**IV. Discussion** Table I describes the upper bounds of the guessing probability calculated by different $N_{tol}$, where $N_{tol}$ is the length of total string that includes the sifted keys for key generation and the string used to do parameter estimation. In Table I, $N_{tol} = 10^4, 10^5, 10^6$ respectively. We can see from Table I that when $N_{tol} = 10^6$, $n = 4.90 \times 10^5$ and the guessing probabilities obtained using the methods of [12] and [11] are about $10^{-6}$ and $10^{-9}$ respectively. However, using our method the guessing probability can be reduced to $2 \times 10^{-3277}$ which is more tightened by thousands of orders of magnitude than prior art methods. As the length of $N_{tol}$ increases, the length of final key also increases, but the guessing probabilities in [12] and [11] almost remain un-
the invalidity of the prior art result. "new upper bound" is trivial and meaningless rather than larger than the prior art result, it only means that the bound, but not meaningful: if any new upper bound is guessing probability. Such a value is correct for upper conclude a large-value upper bound of 100% for Eve's guessing probability can show the insecurity, a Note that, if one does not make any effort, one can also large value of upper bound of Eve's guessing probability cannot show the insecurity. A value is correct for upper bound does not in any sense make an evidence to question the validity of existing the security proof of KQD.

Acknowledgement: We acknowledge the financial support in part by Ministration of Science and Technology of China through The National Key Research and Development Program of China grant No. 2017YFA0303901; National Natural Science Foundation of China grant No. 11474182, 11774198 and U1738142.

Appendix A: the proof of Lemma 1

Lemma 1 The guessing probability of an $\varepsilon_k$-secure key $k$ with length $n_1$ is not larger than $\frac{1}{2^n} + \varepsilon_k$. This is a conclusion from Ref. [11]. The proof has been given in Ref. [11] already, here for readers convenience, we write the proof again.

Proof. Let $n$-bit string $x$ be the $\varepsilon_x$-secure key in $\mathcal{X}$. And the density matrix of Alice and Eve is $\rho_{XE}$ and satisfies

$$
\rho_{XE} = \sum_{x \in \mathcal{X}} |x\rangle \langle x| \otimes \rho^x_E.
$$

(A1)

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
$$

where $\rho_{U_x}$ is the fully mixed state in $\mathcal{X}$. Then we have

$$
\frac{1}{2} \| \rho_{XE} - \rho_{U_x} \otimes \rho_E \|_1 \leq \varepsilon_x,
Eve’s guessing probability of string $x$ is $q(x)$, and the maximum guessing probability is $p_{\text{guessing}} = \max_{x \in X} \{q(x)\}$. Without any loss of generality, assuming that the maximum guessing probability is $q(x')$. Note that $\sum_{x \in X} q(x) = 1$, then the following holds

$$\frac{1}{2} \sum_{x \in X} |q(x) - \frac{1}{2^n}|$$

$$\geq \frac{1}{2} [q(x') - \frac{1}{2^n}] + \frac{1}{2} \sum_{x \in X, x \neq x'} |q(x) - \frac{1}{2^n}|$$

$$\geq |q(x') - \frac{1}{2^n}|. \quad (A3)$$

From the Eqs. (A1)-(A3), we have $p_{\text{guessing}} \leq 2^{-n_1} + \varepsilon_{x}$ and thus for the $n_1$-bit $\varepsilon_k$-secure key $k$, the guessing probability satisfies

$$p(k) \leq \bar{p}(k) = \frac{1}{2^{n_1}} + \varepsilon_k, \quad (A4)$$

where $\bar{p}(k)$ is the upper bound of $p(k)$. This ends our proof of Lemma 1.

**Appendix B: The calculation of $\varepsilon_{k'}$**

We consider the security definitions of a practical QKD protocol with finite-size under the framework of composable security [4][13][14]. Suppose that Alice and Bob get two $N$-bit sifted key strings. By performing an error correction and private amplification scheme, Alice get a $n$-bit final key $k$ and Bob get an estimate $\hat{k}$ of $k$. The protocol is $\varepsilon_{\text{cor}}$-correct if $P[k \neq \hat{k}] \leq \varepsilon_{\text{cor}}$. In general, the key $k$ of Alice can be correlated with an eavesdropper $z$ and the density matrix of Alice and Eve is $\rho_{AE}$.

The protocol outputs an $\varepsilon_k$-secure key [13], if

$$\frac{1}{2} \parallel \rho_{AE} - \rho_U \otimes \rho_{Ez} \parallel_1 \leq \varepsilon_k, \quad (B1)$$

where $\parallel \cdot \parallel_1$ denotes the trace norm, $\rho_U$ is the fully mixed state of Alice’s system. The protocol is $\varepsilon_{\text{tol}}$-secure if $\varepsilon_{\text{cor}}$ and $\varepsilon_k$ satisfies $\varepsilon_{\text{cor}} + \varepsilon_k \leq \varepsilon_{\text{tol}}$ which means it is $\varepsilon_{\text{tol}}$-indistinguishable from a ideal protocol. Without any loss of generality, we consider the case of $\varepsilon_{\text{cor}} = \varepsilon_k$.

From Lemma 1, we can calculate $\bar{p}(k)$ given the $n$-bit $\varepsilon_k$-secure key $k$. In this situation, $\bar{p}(k) = 2^{-n} + \varepsilon_k$. However, in our method with random matrix $R$, we only know $N$, the length of sifted key $n_2$, the length of string $k'$ mapped from $k$ String $k'$ itself can also be regarded as another final key distilled from sifted key. According to Ref. [4], with $N$ and $n_2$, the final key is $\varepsilon_{k'}$-secure if $\varepsilon_{k'}$ satisfies the following equation:

$$n_2 \leq N[1 - h(Q_{\text{tol}} + \mu)] - f \log \frac{2}{\varepsilon_{k'}}. \quad (B2)$$

where $\mu = \frac{N + N_z N_z}{N z N_z} \ln \frac{2}{\varepsilon_k}$, $N_z$ is the length of string used for parameter estimation, $h$ denotes the binary Shannon entropy function, $h(x) = -x \log x - (1 - x) \log(1 - x)$ and $Q_{\text{tol}}$ represents the channel error tolerance. To obtain non-trivial results, we take equality in Eq. (B2) to calculate the value of $\varepsilon_{k'}$, given the input $n_2$. Since $\varepsilon_{k'}$ is dependent on $n_2$, we use notation $\varepsilon_{k'}(n_2)$ for $\varepsilon_{k'}$. Here, $\varepsilon_{k'}(n_2)$ means, given $n_2$, we find the value of $\varepsilon_{k'}$ numerically by Eq. (B2)

In our calculation, we choose specific $n_2$-value that satisfies

$$2^{-n_2} = \varepsilon_{k'}(n_2). \quad (B3)$$

Combining Eq. (B2), we obtain the following equation for a tightened $\varepsilon_{k'}$ value:

$$-\log \varepsilon_{k'} = N[1-h(Q_{\text{tol}}+\mu)]-f \log(Q_{\text{tol}}) - \log \frac{2}{\varepsilon_{k'}}, \quad (B4)$$

and we can calculate the value of $\varepsilon_{k'}$ and then calculate the guessing probability by Eq. (8) in our main body text.

[1] C. H. Bennett and G. Brassard, in *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing* (1984), pp. 175–179.
[2] R. Renner, International Journal of Quantum Information 6, 1 (2008).
[3] M. Curty, F. Xu, W. Cui, C. C. W. Lim, K. Tamaki, and H.-K. Lo, Nature communications 5, 3732 (2014).
[4] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, Nature communications 3, 634 (2012).
[5] M. Ben-Or, M. Horodecki, D. W. Leung, D. Mayers, and J. Oppenheim, in *Theory of Cryptography Conference* (Springer, 2005), pp. 386–406.
[6] R. Renner and R. König, in *Theory of Cryptography Conference* (Springer, 2005), pp. 407–425.
[7] R. König, R. Renner, A. Bariska, and U. Maurer, Physical Review Letters 98, 140502 (2007).
[8] M. Hayashi and T. Tsurumaru, New Journal of Physics 14, 093014 (2012).
[9] M. Alimomeni and R. Safavi-Naini, in *International Conference on Information Theoretic Security* (Springer, 2012), pp. 1–13.
[10] I. Issa and A. B. Wagner, IEEE Transactions on Information Theory 63, 3783 (2017).
[11] C. Portmann and R. Renner, arXiv preprint
arXiv:1409.3525 (2014).

[12] H. P. Yuen, IEEE Access 4, 724 (2016).

[13] R. Canetti, in Proceedings 2001 IEEE International Conference on Cluster Computing (IEEE, 2001), pp. 136–145.

[14] J. Müller-Quade and R. Renner, New Journal of Physics 11, 085006 (2009).