Lietzen, Jari; Vehkalahti, Roope; Tirkkonen, Olav

A Two-way QKD Protocol Outperforming One-way Protocols at Low QBER

Published in:
2020 IEEE International Symposium on Information Theory, ISIT 2020 - Proceedings

DOI:
10.1109/ISIT44484.2020.9174522

Published: 01/06/2020

Document Version
Peer reviewed version

Please cite the original version:
Lietzen, J., Vehkalahti, R., & Tirkkonen, O. (2020). A Two-way QKD Protocol Outperforming One-way Protocols at Low QBER. In *2020 IEEE International Symposium on Information Theory, ISIT 2020 - Proceedings* (pp. 1106-1111). Article 9174522 (IEEE International Symposium on Information Theory). IEEE. https://doi.org/10.1109/ISIT44484.2020.9174522

A Two-way QKD Protocol Outperforming One-way Protocols at Low QBER

Jari Lietzén, Roope Vehkalahti and Olav Tirkkonen
Aalto University
Department of Communications and Networking
Maarintie 8, 02150 Espoo, Finland
Email: {jari.lietzen,roope.vehkalahti,olav.tirkkonen}@aalto.fi

***Abstract*—Two-way quantum key distribution (QKD) protocols can provide positive secret key rates for considerably higher quantum bit error rates (QBER) than one-way protocols. However, when QBER is low, only modest key rate gains have been achieved. This is one of the major obstacles for using two-way protocols. In this paper we introduce a new two-way QKD protocol which overcomes this shortcoming. Under the assumption that the eavesdropper can only perform individual symmetric quantum attacks, our protocol performs quantum key distribution with a secret key rate that is higher than the information theoretical bound limiting the performance of any one-way protocol. This holds true also for very low QBER values.**

I. Introduction

In Quantum Key Distribution (QKD) two entities, Alice and Bob, try to generate a shared secret key using a quantum channel and an authenticated error free classical channel, while a third party Eve is eavesdropping both of these channels.

In the first and most used QKD protocol BB84 [1] Alice generates a random bit sequence and sends it to Bob through the quantum channel. Eve may perform quantum attacks on this communication. In the *sifting phase* following the quantum phase, Bob and Alice have an estimate of the quantum bit error probability (QBER) between their bit strings. Based on the QBER, and due to the nature of the quantum channel, Alice and Bob can estimate how much information Eve has gained during the quantum stage. Using this information and their corresponding bit strings Alice and Bob will use the classical authenticated channel to generate a shared secret key of which Eve should have very little information. If the error rate is too high, it is not possible to produce any secret key and the protocol is aborted.

A number of effective key growing and error correction algorithms to be used with BB84 have been developed [2]–[6]. Most of the considered schemes belong to the category of one-way protocols, which are used in most practical applications.

A one-way protocol takes Alice's bit string as a raw key and the differences in Bob's bit string are corrected as the protocol is run. In principle, only one-way communication is needed for error correction and key distillation in these protocols. The highest rate achieved with such protocols is given in [7]; the highest QBER of which such protocols can provide a positive key rate is 11%. Assuming that we can also perform classical pre-processing the highest reported key rate is given in [6], and a positive key rate can be achieved up to QBER 12.4%. The key rate of all one-way protocols is upper bounded by a general information theoretic bound; the upper bound for a positive key rate of any one-way protocol is at QBER 14.6% [8, p.39].

If Alice and Bob are trying to agree on a common bit string based on message exchanges enabling two-way post-processing, instead of trying to directly perform error correction, this upper bound does not apply. A number of works [9], [10] have considered the problem of extending the positive key rate region by using two-way post-processing. In [10] the authors demonstrate that a two-way system can achieve positive key rate even when QBER is 20%. However, typically when the QBER is low these protocols have a very low key rate, which hinders their applicability for practical QKD.

In [11] the authors demonstrate that a two-way protocol can achieve higher key rate than the best one-way protocol [6], also when QBER is low. However, the improvement in the key rate is small, and does not break the one-way protocol bound [8] at low QBER. In general, the achievable key rate of the best possible two-way protocol is not known [2, p.23], [12, p.96].

In this paper, we consider the BB84 protocol and a scenario where Eve can only perform individual symmetric attacks [2, p.23]. Under this assumption, the achievable secret key rates are typically higher than for more generic attacks. Furthermore, with individual symmetric attacks, complete secrecy analysis can be performed in the realm of classical information theory [12, p.205] while the proofs against general attacks will use quantum information theory [6].

The information theoretical upper bound of [8] still limits the key rate of any one-way protocol, but in this scenario it can be achieved. Moreover, in [13] it was proven that under the assumption of individual attacks a classical two-way protocol can produce a secret key with a positive rate up to 25% QBER, which is the absolute upper limit [9]. However, as far as we know no one- or two-way protocol exists that would break the one-way bound for low QBER values, e.g. below 10%.

In this paper we will introduce a new two-way protocol that can generate secret key for QBER up to 19%. More importantly, the protocol performs better than the theoretical bound [8, p.39] that limits the performance of any one-way protocol, even when QBER is small. Our security proof is only against individual attacks. For example, the protocol in [11] has far lower rate than ours, but is has been proven secure against general quantum attacks. Our protocol is a key growing protocol — a considerable amount of pre-shared key is needed to initiate key growing. This is a rather standard assumption in the QKD literature. However, our protocol relies on pre-shared keys more than most protocols in the literature. We do not have a closed form formula for the asymptotic performance of our protocol, for detailed performance analysis numerical methods are needed.

II. System Model and one-way and two-way protocols

We assume the classical BB84 QKD protocol [1], where Alice and Bob share a quantum channel and an authenticated error free classical channel. Alice creates a random string of bits, encodes it on the polarity of photons by randomly using two different coding bases. During the transmission, Eve will attack the transmitted bits using a quantum attack. We assume that Eve attacks each of the bits individually, symmetrically and always by the same method (while the method is freely chosen by Eve) [2, p.23]. Bob randomly chooses a measurement basis from the two possibilities used by Alice, and performs quantum measurements using the selected basis. After the transmission has ended Alice and Bob will reveal the basis they have used, and discard all the bits where their measurement bases were not the same. This is called the *sifting phase*. After that they perform a random, but jointly chosen permutation of their bit vectors. They then randomly choose some fraction of bits from these vectors and compare them through the classical channel in order to get an estimate of the number of differences. The bits used in the comparison are discarded.

Meanwhile Eve can hold the quantum states of the wire-tapped photons long enough to perform her measurements after the sifting phase has taken place, but not longer.

After Eve's measurement, the whole system can be modelled in terms of classical random variables and a probability density function (pdf) $p(X, Y, Z)$, where Z represents Eve's measurement results and possible side information. It is expected that Eve completely knows this density function, while Alice and Bob only have some partial information that can be read, for example, from the transition probabilities. The probability of bit differences between the codewords of Alice and Bob is called the QBER.

A. Secret key rate of one-way and two-way protocols

In general the secrecy analysis of a QKD protocol has to happen in the quantum realm as in [6]. However, when we assume that Eve is performing her measurements directly after the sifting phase the analysis can be performed in terms of classical information theory [12, p.205]. With this assumption we will now describe how typical one-way and two-way protocols work and what is meant with the secret key rates. We will use the standard definitions for repetitive secret key distillation [12, p.94-96].

A one-way protocol begins with an error correction phase, which follows after the sifting. At the beginning Alice has a length n bit vector x and Bob has an erroneous version y. Alice and Bob then communicate trough the classical channel and try to correct the errors in Bob's vector y. The amount of communication needed depends on the QBER. Eve can listen, but not alter, this communication. After the error correction phase, often referred to as an Information Reconciliation (IR) protocol, Bob's codeword can be modelled as a random vector Y', where $P(X \neq Y') < \epsilon$, for some predetermined ϵ. Based on the observed QBER and the amount of leaked information during the reconciliation, Alice and Bob can now estimate how much information Eve has of X. In order to erase this information Alice and Bob then use a randomly selected *2-universal* hash function [12, p.88], to map their vectors x and y' to length n_{fin} bit-vectors k and k', where k is a binary i.i.d vector with equal probabilities of 1 and 0. This is called a Privacy Amplification (PA) protocol. The probability density function after the IR and PA protocols is $p'(K, K', Z')$. Here Z' represents Eve's original random variable Z and all the additional data she managed to acquire during the execution of the IR and PA protocols, including the choice of the hash function. The constant n_{fin} was selected in such way that $I(K; Z') < \epsilon$.

We say that a QKD protocol achieves a key rate R if for every ϵ we can find $n(\epsilon)$ so that for all $n > n(\epsilon)$ we have that $P(K \neq K') < \epsilon$, $I(K; Z') < \epsilon$ and $\frac{n_{fin}}{n} \geq R - \epsilon$.

The secret key rate of a two-way protocol is defined similarly, but the process does not begin with an error correction process. Instead Alice and Bob use two-way classical communication and simply agree on key words k and k' so that the corresponding random variables satisfy $P(K \neq K') < \epsilon$, $I(K; Z') < \epsilon$ and $I(K'; Z') < \epsilon$.

III. One-way QKD protocol

In this section we discuss a classical one-way error correction and privacy amplification protocol of Lütkenhauss [14], which have become rather standard [15, p.43]. We point out that the presented results are purely information theoretical without any quantum component.

While the protocol in [14] was originally presented as a standalone one-way protocol to be performed after the sifting phase, we will use it as a part of our new two-way protocol. Hence in the following sections we state these results in a general form, where the related finite valued length n random vectors X, Y and Z might not be direct results of the quantum communication and eavesdropping. However, when we are applying results of this section we will always assume that they satisfy the following conditions.

1) X is a random vector with i.i.d binary random variables with equal probabilities for 1 and 0.
2) Random vector Y corresponds to X received through a Binary Symmetric Channel (BSC) with transition probability p.

3) Random vector Z is a sequence of independent identical random variables and for every x and z, $p(x|z) = \prod_{i=1}^{n} p(x_i|z_i)$.

Throughout we also assume that X is the random vector Alice has, Y is Bob's vector and Z Eve's. Furthermore, Eve knows perfectly the probability density function $p(X,Y,Z)$, and for every realization of x and y knows the locations of errors in Bob's word y. If these random vectors are presenting the vectors after sifting, then Condition 2 follows from Eve's attacks being symmetric and Condition 3 from the attacks being individual.

Eve's information of X can now be measured in terms of a collision probability [14, p.9]. The collision probability of X with respect to z is $p_c^x(z) = \sum_x p(x|z)^2$, and the average collision probability can be defined as

$$\langle p_c^x(z) \rangle_z := E_z[p_c^x(z)]. \tag{1}$$

From conditions 1 and 3 it follows that $\langle p_c^x(z) \rangle_z = (p_c^x)^n$, where $p_c^x = \langle p_c^{x_i}(z_i) \rangle_{z_i}$ is the expected collision probability of a single bit.

At the beginning of the protocol Alice and Bob know the transition probability p and an upper bound p_{col} for the average bit collision probability p_c^x.

A. Cost of Reconciliation

Let us now assume Alice has a sequence of bits x of length n and Bob has a possibly erroneous version y. Even without knowing Bob's codeword, Alice can send some information to Bob through the error free classical channel, so that Bob can correct his errors. This problem can be seen as an example of *source coding with side information*, where X is the source and Y is the side information the decoder has.

According to the Slepian-Wolf Theorem [16] the number of bits needed to correct all the errors of Bob's words is at least

$$nh(p) = n(-p \log_2 p - (1-p) \log_2(1-p) \ ), \tag{2}$$

where $h(p)$ is the binary entropy function. When we let n become arbitrarily large, the probability that all errors in Bob's word will be corrected with $nh(p)$ bits will approach 1. Hence if Y' is Bob's random vector after the error correction, then $P(X \neq Y')$ can be pushed arbitrarily close to zero.

As in [14] we assume that Alice and Bob already share secret key material and Alice will encrypt the bits and send them to Bob through the public channel. Note that this consumes existing secret key, and has to be taken into account when the rate of generating key material is ultimately computed. When using this approach Eve will not gain any information of the word x except about the number of errors Bob has. As we assumed at the beginning that Eve already possesses this information, the error correction process does not increase Eve's information of X.

B. Cost of privacy amplification

As described in the previous section the used error correction method did not leak any additional information to Eve and therefore the original pdf $p(X,Z)$ still presents Eve's knowledge of X. In the following we will shortly recall from [14] how knowing p_{col} allows us to measure how much we have to compress X so that Eve's information of the result will be as small as we want.

We will denote with G a random variable that presents 2-universal hash functions [17] that map length n binary vectors into length n_{fin} binary vectors, and use a shorthand $G(X) = K$. Eve's knowledge of K is now presented by the probability distribution $p(k|z,g)$, which is the probability that the key is k given Eve's measurement and side-information z and knowledge of the random hash g.

In [14, Eqn. 9] it was shown that

$$I(K;Z,G) \leq \log_2[2^{n_{\text{fin}}} \langle p_c^x(z) \rangle_z + 1]. \tag{3}$$

For a detailed proof of this result we refer the reader to the Appendix. Selecting a security parameter n_{S}, and setting $n_{\text{fin}} = (1-\tau_1)n - n_{\text{S}}$, where $\tau_1 = 1 + \frac{1}{n}\log_2(\langle p_c^x(z) \rangle_z)$, we have

$$I(K;Z,G) \leq \log_2(2^{-n_S} + 1) \approx \frac{2^{-n_S}}{\ln(2)}.$$

Here we see that shortening the final key with n_{S} will exponentially decrease the average information that Eve has. Hence when calculating the final key rate we are satisfied with reducing Eve's information of K to a single bit by selecting $n_{fin} = -\log_2(\langle p_c^x(z) \rangle_z) = -n \log_2(p_{\text{col}}))$.

C. The Achievable Rate

As most QKD protocols, the protocol here is not about key generation but about key growing. Hence we are measuring the long term achievable key rate taking into account how much previously generated key we are using when generating new key. Furthermore we consider asymptotic performance with arbitrarily long sequences X, and assume capacity approaching codes that guarantee that the error correcting process of Section III-A can be performed with $nh(p)$ bits with arbitrarily high probability. After the error correction phase we have to reduce Eve's information of the corrected key to at most 1 bit by selecting $n_{fin} = -n \log_2(p_{\text{col}}))$. The achievable secret key rate now becomes

$$R_0 \geq -\log_2(p_{\text{col}}) - h(p), \tag{4}$$

where the term $h(p)$ describes the amount of previously generated key the protocol consumes.

IV. Two-way Protocol with Parity Bit Reconciliation

The novel Two-way Protocol with Parity bit Reconciliation (2PPR) uses the secrecy distillation method from [9], where on each round Alice and Bob randomly divide their codewords into blocks, calculate parity bits of these blocks, and in the case where the parity bits differ, jointly discard these blocks. The surviving bits will then be used as input for the next round of the protocol.

In contrast to [9] and following [11] we use the knowledge that the parity bits between Alice and Bob are strongly correlated and only send sufficient information that Bob can correct his parity bits. Moreover this data is transmitted through the public channel secretly by using secret key from previous rounds. The biggest difference compared to previous protocols is that we are collecting our secret key from the parity bits and not from Alice's original string. Essentially in each round we are applying the one-way protocol of Section III to reconcile the parity bits. The final secret key is then a concatenation of the bits collected in each round.

A. Initial Collision and Error Probabilities

The 2PPR protocol begins after the sifting phase. The length of random vectors that Alice, Bob and Eve have is denoted by $N_{\rm sif}$. The QBER has been estimated during the sifting process and is assumed to be known by all players. The sequences satisfy the conditions of Section III and have a joint pdf $p(X,Y,Z)$. The transition probability between Alice and Bob is denoted with p.

Following [14, Eqn. (59)], Eve's bitwise collision probability of X satisfies

$$p_{col} := E_z[p_c^x(z)] \leq \frac{1}{2} + 2p - 2p^2 \ , \tag{5}$$

where $p \leq 1/2$. This was derived under the assumption that Eve has spoiling information and knows the locations where Bob has done an error in the receiving. In [14], this result was stated for Eve's collision probability *after* the error correction process. However, this result is also valid in the case when error correction has not yet been performed. This follows from the fact that [14] uses encrypted error correction that does not leak any additional information for Eve during the error correction process.

B. Advantage Distillation with Secret Keys from Parity Bits

At the beginning of the protocol, we have $f_{\rm in}^1 = 1$, $p_{\rm in}^1 = p$ and $x_{\rm in}^1 = p_{col}$. Round j is then as follows.

1 Alice has a fraction $f_{\rm in}^j$ remaining of the $N_{\rm sif}$ sifted bits. Bob's QBER is estimated to be $p_{\rm in}^j$, and Eve's collision probability is $x_{\rm in}^j$.

2 Alice and Bob randomly, but jointly, segment their codewords to blocks with 2 bit segments.

3 Alice and Bob compute parity check bits for each block.

4 Alice and Bob compute the error probability $p_{\rm par}^j$ of their parity check bits.

5 Alice computes the collision probability $x_{\rm par}^j$ of the parity check bits.

6 Alice computes the secret key rate of the parity bits,

$$R_{\rm par,j} = (f_{\rm in}^j/2) \max\left(-\log_2(x_{\rm par}^j) - h(p_{\rm par}^j),\ 0\right)$$

7 If $R_{\rm par,j} > 0$, Alice sends $N_{\rm sif}(f_{\rm in}^j/2)h(p_{\rm par}^j)$ redundancy bits to Bob, to correct the parity check bits. The redundancy bits are transmitted by using one-time pad encryption with existing key bits.

8 Alice and Bob both save the corrected parity bits.

9 If $R_{\rm par,j} \leq 0$, Alice sends the parity bit information over the public channel.

10 Both Alice and Bob remove the blocks with erroneous parity bits, sharing this information over the public channel.

11 Alice and Bob select at random (but jointly) one bit from each kept block. These bits are kept for the next repetition. These constitute a fraction $f_{\rm out}^j$ of the sifted bits, their bit-error rate is $p_{\rm out}^j$, and the collision probability is upper bounded by $x_{\rm out}^j$.

The bits collected in step 8 are all equal for Bob and Alice. Eve's information of them will be erased by using classical privacy amplification, presented in Section III, after each round. The resulting bit strings will be concatenated to produce the final key. The bits from Step 11 are fed to the protocol and the protocol continues until nothing is left of the sifted bits. The bit-error rate $p_{\rm out}^j$ and the collision probability $x_{\rm out}^j$ are used as input parameters in Step 1 as the protocol enters a new round.

C. Finding the Values for Error and Collision Probabilities

In order to perform the protocol Alice and Bob have to be able to calculate the values $p_{\rm par}^j$ and $x_{\rm par}^j$, given the estimates for $p_{\rm in}^j$ and $x_{\rm in}^j$ at the beginning of the round. They also have to be able to find $p_{\rm out}^j$ and $x_{\rm out}^j$ based on $p_{\rm in}^j, x_{\rm in}^j$ and the initial values $p_{\rm col}$ and QBER p. However, if we want to apply the results from Section III for collecting key on Step 8, we also have to check that the corresponding random vectors satisfy the conditions of Section III. We have collected these results to the following two Lemmas whose proofs can be found in the Appendix.

Lemma 1: At the beginning of each round of the protocol the random vectors of Alice, Bob and Eve satisfy the conditions of Section III. Furthermore

$$p_{\rm out}^j = \frac{(p_{\rm in}^j)^2}{(1-p_{\rm par}^j)} \text{ and } x_{\rm out}^j \leq (1-p_{\rm out}^j)\frac{p_{\rm col}}{1-p}. \tag{6}$$

Lemma 2: Alice's and Bob's parity bits calculated on step 3 and Eve's information of them can modelled with random vectors satisfying the conditions of Section III. Furthermore

$$p_{\rm par}^j = 2(1-p_{\rm in}^j)p_{\rm in}^j \tag{7}$$

and

$$x_{\rm par}^j = (x_{\rm in}^j)^2 + (1-x_{\rm in}^j)^2. \tag{8}$$

V. Analysis of the secret key rate of the protocol

In this section we are finding the secret key rate of the 2PPR protocol in a theoretical sense. In the analysis we assume that the protocol is run for arbitrarily long sequences and that we therefore can apply the one-way protocol of Section III optimally. We also assume that during a single run, only a small predetermined number of rounds is used. Hence the assumption that we can use asymptotic analysis will hold during the whole run of the protocol. In the following analysis, we use the standard definition for the secret key rate of a two-way system as described in [12, Def. 5, p.94] and Section II-A.

Before proceeding we note that while we are only measuring the information Eve has of Alice's sequence it is easy to see that Eve cannot have more information about Bob's sequence.

Theorem 5.1: During n rounds, the 2PPR protocol asymptotically achieves secret key rate

$$\sum_{j=1}^{n} (f_{\text{in}}^{j}/2) \max\left(-\log_2(x_{\text{par}}^{j}) - h(p_{\text{par}}^{j}),\ 0\right), \qquad (9)$$

where the values for x_{par}^{j} and p_{par}^{j} are from Lemma 2 and

$$f_{\text{out}}^{j} = (f_{\text{in}}^{j}/2)\left(1 - p_{\text{par}}^{j}\right) \ . \qquad (10)$$

Proof: During the running of the protocol the secret bits are collected in Step 8, where we apply the one way protocol of Section III to the parity bits. According to Lemma 2 we can use the secret key rate analysis from that section and find that during a single round Alice and Bob collect

$$N_{sif}(f_{\text{in}}^{j}/2) \max\left(-\log_2(x_{\text{par}}^{j}) - h(p_{\text{par}}^{j}),\ 0\right)$$

bits. Here we have also taken into account the bits consumed by the error correction. Due to the properties of the used one-way protocol the bits collected by Alice are i.i.d with probability $\frac{1}{2}$ for 1 and 0, and are equal for Alice and Bob with arbitrarily small error probability. In each round we leak at most one bit of information. As we are running the protocol a small number of times, we can apply asymptotic analysis such that this leakage can be erased with no visible reduction in the achievable rate as in Section III.

Next we need to prove that we can concatenate the secret key bits collected during each round to form the final secret key. Due to Step 11 the component vectors are uncorrelated. There Alice and Bob randomly select from each kept pair of bits one for the round $j + 1$. By elementary probability theory this selected bit is independent of the parity bit of the corresponding pair. As the key collected in round j consists of processed parity bits it follows that the raw key passed to round $j + 1$ is independent of the bits collected during round j and any previous round. Therefore Alice and Bob can simply concatenate the bits collected in each round to form the final key and the collected secret key is still uncorrelated. Furthermore, Eve's information of the concatenated key is at most the sum of the information of the components. Hence for a finite number of rounds we can push Eve's information of the whole key to an arbitrarily low value without reducing the rate. For the final key rate we divide the number of collected bits with the number of initial bits N_{sif}.

Finally, during the protocol run Alice and Bob can keep track of f_{in}^{j}. However, we have to prove that we can use value (10) for estimating the achievable key rate. In steps 10 and 11 we discard the blocks with erroneous parity bits, and choose at random one bit from each remaining block. Direct calculation gives that the expected fraction of the original bits that are kept for the next round is $f_{\text{out}}^{j} = (f_{\text{in}}^{j}/2)\left(1 - p_{\text{par}}^{j}\right)$.

By applying the law of large numbers, we can see that when analysing the achievable rate we can replace the actual values of f_{out}^{j} with the average value (10) in each round. ■

VI. Numerical performance analysis

Theorem 5.1 does not allow easy analytic key rate analysis due to its recursive nature. We have therefore evaluated the performance of the 2PPR protocol by using numeric analysis. For a given QBER, and number of rounds, the recursion can be computed. The parameters f_{out}^{j}, f_{in}^{j} and p_{par}^{j} change from round to round and depend on the error rate and the collision probability. The values for these parameters can be found in Section IV-C. For each round, the key rate function in Equation (9) is evaluated. The initial value for the collision probability for the first round comes from Equation (5). The numerical result for twelve rounds of the protocol is presented in Fig. 1 in a QBER range up to 20%. The 2PPR protocol is giving a positive key rate up to approximately 19.09% QBER and outperforms the theoretical upper bound for one-way protocols.

The theoretical upper bound for a one-way protocol at a given QBER value is the difference between mutual information between Alice and Bob $I(\alpha, \beta) = 1 - h(p)$ and between Alice and Eve $I^{\max}(\alpha, \epsilon)$. The mutual information between Alice and Eve is calculated according to [8, Eq. 64, p.39]

$$I^{\max}(\alpha, \epsilon) = 1 - h\left(\frac{1 + \sin(x)}{2}\right), \qquad (11)$$

where $h(p)$ is the binary entropy function as in (2) and $x = \arccos(1 - 2p)$ [8, p.39]. The theoretical upper bound for the key rate for a one-way algorithm is then $R = I(\alpha, \beta) - I^{\max}(\alpha, \epsilon)$, which is presented in Fig. 1 with a red curve. Watanabe *et al.* demonstrated in [11] that a two-way protocol can achieve higher key rate than the best one-way protocol. The achieved key rate for Watanabe *et al.* from [11, fig. 2] is shown in Fig. 1 for comparison. This protocol is not able to break the theoretical one-way protocol bound, while our 2PPR protocol with 12 rounds outperforms the one-way bound for almost the whole QBER range.

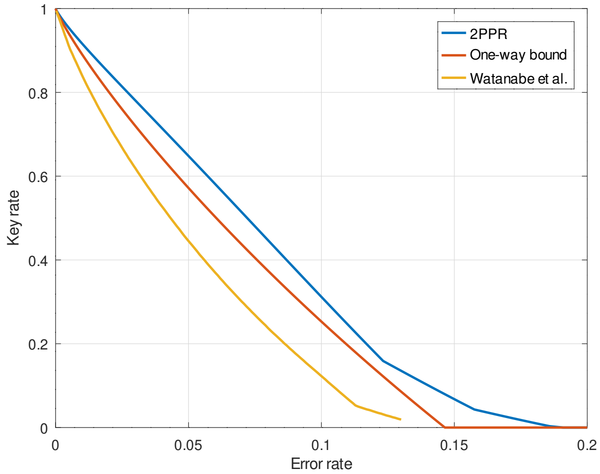

Fig. 1. The key rate of the proposed 2PPR protocol.

Acknowledgment

This work was funded in part by MATINE, and the Academy of Finland (grant 319578).

VII. Appendix

A. A proof of equation (3)

In Section III-B we were using equation

$$I(K;Z,G) \leq \log_2[2^{n_{\rm fin}}\langle p_c^x(z)\rangle_z + 1]. \tag{12}$$

The original proof can be found from [14], but for completeness sake we will prove it here.

Before proceeding let us recall the connection between collision probability and *Rényi entropy*. Let X be a finite random variable with alphabet $\mathcal{X}$ and probability distribution P_X. The *collision probability* $P_c(X)$ of X is defined as the probability that in two independent experiments X takes the same value [17]

$$P_c(X) = \sum_{x \in X} P_X(x)^2. \tag{13}$$

The collision probability of X with respect to z where Z is a random variable is $p_c^x(z) = \sum_x p(x|z)^2$. The Rényi entropy of order two (”Rényi entropy” for short) of a random variable X is defined as

$$R(X) = -\log_2(P_c(X)), \tag{14}$$

where $P_c(X)$ is the collision probability of X. Obviously this equation holds also for conditioned random variables and hence

$$R(X|z) = -\log_2(P_c(X|z)).$$

Let us now proceed with the proof of Equation (12).

Proof: Eve's expected Shannon information of K is

$$I(K;G,Z) = H(K) - H(K|G,Z), \tag{15}$$

where $H(K)$ is the entropy of the secret key and $H(K|G,Z)$ is the conditional entropy of the secret key given hash functions and Eve's knowledge of the reconciled key. Hashing produces a uniformly distributed key and the Shannon entropy of the secret key is therefore

$$H(K) = n_{\rm fin}. \tag{16}$$

In order to find an upper bound for the mutual information (15) we now only need a lower bound for $H(K|G,Z)$.

From Bennet *et al.* [17] Theorem 3 and Corollary 4, we have

$$H(K|G,Z=z) \geq n_{\rm fin} - \log_2\left(1 + 2^{n_{\rm fin} - R(X|z)}\right). \tag{17}$$

Averaging Equation 17 over values of z and noting that $P_c(X|z) = 2^{-R(X|z)}$, we have

$$\begin{aligned} H(K|G,Z) &= E_z[H(K|G,Z=z)] \\ &\geq n_{\rm fin} - \log_2\left(1 + E_z[2^{n_{\rm fin}}P_c(X|z)]\right) \\ &= n_{\rm fin} - \log_2\left(1 + 2^{n_{\rm fin}}\langle p_c^x(z)\rangle_z\right), \end{aligned}$$

where the first inequality follows from Jensen's inequality. Combining now this result and Equation 16, gives us the claim. ■

B. Proof of Lemma 1

1) Proof of the first equation of Lemma 1: Alice's and Bob's bit sequences coming to the Step 1 are either results of the original BB84 protocol or then punctured versions of these sequences. In both cases it is easy to see that they are i.i.d random binary sequences and Bob's sequence is like Alice's but received through BSC channel with transition probability $p_{\rm in}^j$. As at the beginning of the protocol we know $p_{\rm in}^1$, we can assume we know this value. Let us now consider one of the blocks that was selected in Step 3 of the protocol. The probability that the parity bits of Alice and Bob agree is $(1-p_{\rm par}^j)$. In Step 11 we randomly select a bit of such block. Then the probability that the value differs between Alice and Bob is

$$p_{\rm out}^j = \frac{(p_{\rm in}^j)^2}{(1-p_{\rm par}^j)}, \tag{18}$$

which is what we claimed.

Before proceeding we need the following Lemma, which is true for $p_{\rm in}^1$ as we assumed that QBER is below 25% and therefore true for all $p_{\rm in}^j$ due to the Lemma.

Lemma 3: As long as $p_{\rm in}^j < 1/3$, we have that

$$p_{\rm out}^j \leq p_{\rm in}^j.$$

Proof: Inserting Equation (7) to Equation (18) we have that

$$p_{\rm out}^j = \frac{(p_{\rm in}^j)^2}{(1-2(p_{\rm in}^j)(1-p_{\rm in}^j))}.$$

As we assumed that $p_{\rm in}^j < 1/3$, we can see that

$$1-2(p_{\rm in}^j)(1-p_{\rm in}^j) > 1-2p_{\rm in}^j > p_{\rm in}^j,$$

and the claim follows.

■

Note that while we have not yet proven Equation (7), we are not doing anything criminal as in the proof of that result we do not need Lemma 3.

2) Proof of the second equation of Lemma 1: On the first round of the protocol we assume that Eve's and Alice's random vectors satisfy the third condition of Section III. Let us now assume that Eve's and Alice's sequences satisfy this condition on round j and consider two bits x_1 and x_2 that are surviving for the round $j+1$. These bits are selected of pairs of bits (x_1^1, x_1^2) and (x_2^1, x_2^2). At the beginning of the protocol, Eve's information of these pairs of bits can be presented as random variables that are independent of each other. During the Steps from 1 to 6 we will only reveal information of the locations of bits where Bob's bit differs from Alice's, but we assumed already in the beginning that Eve has this knowledge. Hence the only additional data we possibly leak for Eve is in Step 9 when we reveal parity bits of these pairs, which can create correlations inside the pairs of bits. However when we select one bit of each pair randomly, the resulting bits are uncorrelated. It follows that at the beginning of the round $j+1$ Eve's information of Alice's bits can still be presented in the form of condition 3 of Section III.

Let us now prove an upper bound for x^j_{out}. Let x_i be Alice's random bit on the first round of the protocol and z_i Eve's corresponding random variable. We can further divide the random variable z_i so that $z_i = (z'_i, e)$, where e is a binary random variable presenting whether the bit x_i was received incorrectly (1) by Bob or not (0). Using this notation and Definition (1), Eve's collision probability with respect to x_i can be divided to two parts

$$p_{\text{col}} = p x_{c,1} + (1-p) x_{c,0}, \tag{19}$$

where $x_{c,1}$ is the collision probability averaged over all z_i, where $z_i = (z'_i, 1)$ and $x_{c,0}$ is defined in similar fashion. The original estimate for p_{col} is gotten from Equation (5) and p is the original transition probability between Alice and Bob. The proof will now get divided to two parts depending whether Step 7 or Step 9 is selected.

Let us now assume that Step 7 is selected. During the Steps from 1 to 6 we will only reveal information of the locations of bits where Bob's bit differs from Alice's. However, our estimate for the collision probability assumed that Eve already at the beginning of the protocol had this information. We can therefore conclude that if a random variable x_i presents a bit that was not cut away in Steps 10 and 11, then Eve's information of this bit can be presented with the original probability density function $p(x_i|z_i)$ of Section IV-A. However, during the cutting process we have affected how often $e = 0$ and how often $e = 1$.

Therefore, for the bits surviving for the next round we get

$$x^j_{\text{out}} = p^j_{\text{out}} x_{c,1} + (1 - p^j_{\text{out}}) x_{c,0}. \tag{20}$$

From equation (19) we have that

$$\frac{(1-p^j_{\text{out}})}{(1-p)} p x_{c,1} + (1-p^j_{\text{out}}) x_{c,0} = (1-p^j_{\text{out}}) \frac{p_{\text{col}}}{1-p}.$$

From Lemma 3 we see that $p^j_{\text{out}} \le p^j_{\text{in}}$ and hence $p^j_{\text{out}} \le p$. It follows that

$$x^j_{\text{out}} = p^j_{\text{out}} x_{c,1} + (1-p^j_{\text{out}}) x_{c,0} \le (1-p^j_{\text{out}}) \frac{p_{\text{col}}}{1-p}. \tag{21}$$

Let us now assume that the protocol chooses Step 9 instead of Step 7 and that we have a block of bits (x_i, x_j) whose value of parity is the same for Bob and Alice. Alice then reveals this information. This would clearly increase Eve's information of the pair x_i and x_j. However, when we randomly, with probability half, choose one of the bits for the next round we erase Eve's extra information of the chosen bit. Therefore just like in the previous section we can conclude that the collision probability x_{out} is given by Equation (20) and an upper bound by (21).

C. Proof of Lemma 2

According to Lemma 1 Alice's and Bob's bit sequences coming to the Step 1 satisfy the conditions of Section III with transition probability p^j_{in}. In Step 3 Alice and Bob form bit sequences by taking parity bits of their corresponding bit sequences. As the bits in their original sequences are i.i.d so are the parity bits. Furthermore Bob's sequence is again like Alice's received through BSC channel. Only the error probability has now changed.

Let us concentrate on a single block of two bits in Step 2 of the protocol. Bob's parity bit differs from Alice's, if one bit is in error in the parity check block. Accordingly, we have that

$$p^j_{\text{par}} = 2(1-p^j_{\text{in}}) p^j_{\text{in}}. \tag{22}$$

As we did see in Section VII-B2 also the random vectors of Alice and Eve satisfy the conditions III. It directly follows that when Alice forms a sequence of random variables consisting of the parity bits of her original sequence, these parity bits and Eve's information of them satisfy the conditions III as well. However, one should realize that given a pair of bits (x_s, x_t) and their corresponding parity bit x_k then Eve's corresponding random variable is $(z_s, z_t) = z_k$.

It follows that the collision probability of the parity bits can be calculated in a bitwise manner.

Let us now assume that we have two i.i.d binary random variables x_1 and x_2 and two i.i.d random variables z_1 and z_2, that satisfy $p(x_1, x_2|z_1, z_2) = p(x_1|z_1)p(x_2|z_2)$.

Let us define a random variable $d = x_1 + x_2$, where the sum is calculated modulo 2.

Lemma 4: We have that

$$\langle p^d_c(z) \rangle_z = (p^x_c)^2 + (1 - p^x_c)^2, \tag{23}$$

where $z = (z_1, z_2)$ and $(p^x_c) = \langle p^{x_i}_c(z_i) \rangle_{z_i}$.

Proof: For fixed z_1 we have that

$$\begin{aligned} p^{x_1}_c(z_1) &= p(x_1 = 1|z_1)^2 + (1 - p(x_1 = 1|z_1))^2 \\ &= 1 - 2p(x_1 = 1|z_1) + 2p(x_1 = 1|z_1)^2, \end{aligned}$$

and similar expression for the pair x_2 and z_2. It follows that

$$p^x_c = E_{z_1}[1 - 2p(x_1 = 1|z_1) + 2p(x_1 = 1|z_1)^2].$$

Let us now find the collision probability for d. First we have that

$$\begin{aligned} p(d = 1|z) &= p(x_1 + x_2 = 1|z_1, z_2) \\ &= p(x_1 = 1|z_1)(1 - p(x_2 = 1|z_2)) \\ &+ p(x_2 = 1|z_2)(1 - p(x_1 = 1|z_1)). \end{aligned}$$

By direct calculation we then get that

$$\begin{aligned} p^d_c(z) &= p(d = 1|z)^2 + (1 - p(d = 1|z))^2 \\ &= p^{x_1}_c(z_1) p^{x_2}_c(z_2) + (1 - p^{x_1}_c(z_1))(1 - p^{x_2}_c(z_2)). \end{aligned}$$

Due to the independence assumptions we made we therefore have the following average

$$\begin{aligned} \left\langle p^d_c(z) \right\rangle_z &= \left\langle p^{x_1}_c(z_1) \right\rangle_{z_1} \left\langle p^{x_2}_c(z_2) \right\rangle_{z_2} \\ &+ (1 - \left\langle p^{x_1}_c(z_1) \right\rangle_{z_1})(1 - \left\langle p^{x_2}_c(z_2) \right\rangle_{z_2}). \end{aligned}$$

■

References

[1] C. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*. IEEE, 1984, pp. 175–179.

[2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of modern physics*, vol. 81, no. 3, p. 1301, 2009.

[3] K. Nguyen, G. V. Assche, and N. J. Cerf, "Side-information coding with turbo codes and its application to quantum key distribution," in *in Proc. International Symposium on Information Theory and its Applications*, 2004.

[4] D. Elkouss, J. Martínez Mateo, and V. Martin, "Information reconciliation for quantum key distribution," *Quantum Information and Computation*, vol. 11, 07 2010.

[5] P. Jouguet and S. Kunz-Jacques, "High performance error correction for quantum key distribution using polar codes," *Quantum Information and Computation*, vol. 14, 04 2012.

[6] R. Renner, N. Gisin, and B. Kraus, "Information-theoretic security proof for quantum-key-distribution protocols," *Phys. Rev. A*, vol. 72, p. 012332, Jul 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.72.012332

[7] P. W. Shor and J. Preskill, "Simple Proof of Security of the BB84 Quantum Key Distribution Protocol," *Physical Review Letters*, vol. 85, pp. 441–444, Jul 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.85.441

[8] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Mod. Phys.*, vol. 74, pp. 145–195, Mar 2002. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.74.145

[9] D. Gottesman and H.-K. Lo, "Proof of security of quantum key distribution with two-way classical communications," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 457–475, Feb 2003.

[10] J. Bae and A. Acín, "Key distillation from quantum channels using two-way communication protocols," *Phys. Rev. A*, vol. 75, p. 012334, Jan 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.75.012334

[11] S. Watanabe, R. Matsumoto, T. Uyematsu, and Y. Kawano, "Key rate of quantum key distribution with hashed two-way classical communication," *Phys. Rev. A*, vol. 76, p. 032312, Sep 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.76.032312

[12] G. Van Assche, *Quantum cryptography and secret-key distillation*. Cambridge University Press, 2006.

[13] N. Gisin and S. Wolf, "Quantum cryptography on noisy channels: quantum versus classical key-agreement protocols," *Physical Review Letters*, vol. 83, no. 20, p. 4200, 1999.

[14] N. Lütkenhaus, "Estimates for practical quantum cryptography," *Phys. Rev. A*, vol. 59, pp. 3301–3319, May 1999. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.59.3301

[15] M. Pivk, "Quantum key distribution," in *Applied Quantum Cryptography*. Springer, 2010, pp. 23–47.

[16] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[17] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.