Calculation of key reduction for B92 QKD protocol

Miralem Mehic*, Pavol Partila, Jaromir Tovarek, Miroslav Voznak
VŠB-Technical University of Ostrava, 17. listopadu 15, 708 00 Ostrava-Poruba, Czech Republic

ABSTRACT

It is well known that Quantum Key Distribution (QKD) can be used with the highest level of security for distribution of the secret key, which is further used for symmetrical encryption. B92 is one of the oldest QKD protocols. It uses only two non-orthogonal states, each one coding for one bit-value. It is much faster and simpler when compared to its predecessors, but with the idealized maximum efficiencies of 25% over the quantum channel. B92 consists of several phases in which initial key is significantly reduced: secret key exchange, extraction of the raw key (sifting), error rate estimation, key reconciliation and privacy amplification. QKD communication is performed over two channels: the quantum channel and the classical public channel. In order to prevent a man-in-the-middle attack and modification of messages on the public channel, authentication of exchanged values must be performed. We used Wegman-Carter authentication because it describes an upper bound for needed symmetric authentication key.
We explained the reduction of the initial key in each of QKD phases.

Keywords: B92, QKD, Quantum Key Distribution, Symmetric Cryptography

INTRODUCTION

Until recently, cryptography was the point where security engineering meets mathematics. Today, this definition is extended to include the principles of quantum physics. Quantum cryptography uses quantum physical principles to establish symmetrical binary keys between legitimate users that will use these keys to encrypt theirs communication data. Therefore, this technology can be better described as "Quantum Key Distribution" or just QKD.
The concept of quantum cryptography was originally proposed in 1960s by Stephen Wiesner, a student of Columbia University, and finally published in 1983[1], thought its real development is recorded from 1984 when Charles Bennett (IBM) and Gilles Brassard (University of Montreal) presented the first QKD protocol[2]. Five years later they made a first practical demonstration of QKD by exchanging[i] secret key over 30cm through the air. This protocol is even today most widely used. Eight years later Bennet developed B92[3] protocol that is simpler and faster than its predecessor.
B92 is one of the oldest QKD protocol which consists of several post-QKD[ii] operations in which initial key is significantly reduced. In this article, we analyzed these phases and we provided equation which can be used for calculation of the length of the final key based on parameter "level of security" and quantum bit error rate (QBER) in the quantum channel for B92 protocol.

SECRET KEY EXCHANGE

In order to simplify BB84 protocol, Benner developed B92 protocol by catching the essence of non-distinguishable quantum states in the simplest way. In order to establish a secret key using B92 protocol, the sender hereinafter named Alice and the recipient hereinafter named Bob, must follow following communication steps. First Alice needs to generate a random binary sequence S_A=[11001011] of length Q and use non-orthogonal polarizations to modulate the photons, where *i* refers to the i^{th} bit of a sequence S_A:

*miralem.mehic.st@vsb.cz; phone +420-59732-7255; www.vsb.cz

[i] To be precise, establishing a secret key.

[ii] Error-rate estimation, Key reconciliation, Privacy amplification, Authentication

$$|\varphi\rangle = \begin{cases} |0\rangle & if\ S_A[i] = 0 \\ \dfrac{|0\rangle + |1\rangle}{\sqrt{2}} & if\ S_A[i] = 1 \end{cases} \quad (1)$$

Then Alice sends modulated photons over the quantum channel to Bob. Since Bob has no information about the sequence S_A which Alice used, he needs to generate his random sequence S_B=[00101010] and apply identical non-orthogonal polarizations rules (1) to his sequence S_B in order to measure the incoming photons.
If Bob chooses the correct basis, he will measure the incoming photon. However, if he chooses the wrong basis, he will not measure anything; a condition in quantum mechanics which is known as an erasure [4(p20)]. Finally, after completion of the measurement, Bob will inform Alice about the positions in which he received incoming photons. Let Q be the length of the raw key, then $\log_2 Q$-bits are necessary to tell Alice the measured positions.
Then, Alice and Bob will discard the bits corresponding to the photons which Bob measured with an incorrect basis. Note that Bob does not reveal anything about the basis he used.
However, we can conclude that the original length of the key Q is significantly decreased since Bob reliably receives approximately 25% of the original key. In the rest of this paper, B represents the length of Bob's reliably received key.

ERROR RATE ESTIMATION

Errors in the quantum channel may occur due to a disturbance of the quantum channel, noise in the detectors or an optical misalignment and other reasons. Also, errors may also occur due to eavesdropping by eavesdropper Eve. In order to detect the presence of Eve, Alice and Bob need to determine the error rate *p* of the quantum channel. The threshold of bit error rate (p_{max}) for quantum channel without presence of Eve is known in forward, so Alice and Bob must compare a small portion of their key in order to estimate the quantum bit error rate (QBER). If the error rate is higher than a given threshold ($p>p_{max}$), Alice and Bob revealed the presence of Eve and the key distribution process starts all over again. If not, then the channel was secure, and Alice and Bob can continue with the further distillation of the key.
Alice and Bob need to decide about the length of the sample block which is going to be used to estimate error rate. If they choose a short sample block (uncover 0 bits), then they will know nothing about the QBER and the possible presence of Eve. On the contrary, if they choose a long sample block (uncover all n bits), they are sure completely sure about QBER value in the channel. But then they will shorten the key even more since the values of the sample block must be announced publicly and Eve may have access to these values.
According to [5], it is necessary to define "level of security" $\hat{S}(k)$ parameter that determines the length of the sample block. This parameter is defined in Equation (1) where the value k can be presented as the percentage of the key used for the QBER estimation.

$$\hat{S}(k) = \frac{-\sum_{k=1}^{n} \frac{k}{n} \log \frac{k}{n}}{n} \quad (2)$$

Two levels of this parameter are defined (basic and advanced) and they are bounded with the percentage of the key assigned to comparison as shown in table 1:

Table 1. Minimal and maximal values of "level of Security" $\hat{S}(k)$ parameter

Level	Min value	Part of key assigned to comparison (%)	Max value	Part of key assigned to comparison (%)
Basic	0.01	8.16	0.1	39.10%
Advanced	0.1	39.10	0.24	85.49%

However, Alice and Bob must delete the part of the key which they used for estimation of the error rate. It means that the original key will be shortened even more. We use notation R to mark the length of the key after this phase.

KEY RECONCILIATION

If error rate is below a given threshold, Alice and Bob assume that Eve was not eavesdropping. However, their key still contains some errors and the key is not totally symmetrical. Alice and Bob need to find the position of errors in the key and then correct the errors. This phase is known to be highly interactive and time-consuming since the discussion about the location of errors in the key is performed through the public channel. So, it is necessary to use simple and efficient solution like Cascade [6] which is the most widely used reconciliation protocol. Of course, there are other solutions like using Turbo Codes [7], LDPC codes [8, 9], but in this article we stay focused to Cascade. Using error-correcting codes to achieve best results is an option only at first glance, since any number of errors above the detection threshold of the code will go undetected.

Cascade begins with the random permutation of the key with the objective to evenly disperse errors throughout the key. The permuted key is divided into equal blocks of k_i bits, and Cascade continues to run iteratively in the given number of iterations. After the each iteration, permutations are performed again and the block size is doubled: $k_i = 2*k_{i-1}$. The number of iterations i is increased to the value for which the length of block k_i can be used to split the original key into two parts ($k_i < \frac{n}{2}$).

For each block, Alice and Bob will exchange the results of the parity test and perform a binary search to find and correct errors. Instead of going through all the iterations continuously, the Cascade protocol investigates errors in pairs of iterations. The process is recursive and no bits are discarded during the first iteration. It means that for any error corrected in the second iteration there must be at least one matching error contained in the same block in the previous iteration since neither error was found or corrected in that iteration. For this reason, for each correction made in any iteration after the first one, a binary search is rerun on the block containing the bit corrected in all previous iterations, in order to identify any potential matching errors. For any new error detected, it follows that another error in a previous iteration was masked, thus the process is repeated so that the error detection and correction process cascades through all previous iterations. This process is illustrated in Figure 1, where the following notation is used: e_i represent identified errors, e_m represent masked errors, and e_c represent errors that have already been corrected.

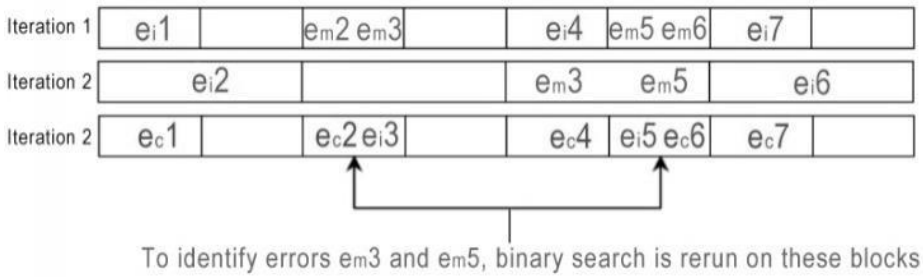

Figure 1. Error detection process in cascade protocol

Obviously, the length of the initial block k_1 is a critical parameter and should depend on the estimated error rate. An empirical result in the original paper [6] indicates the optimal value k_1 as 0.73/p, where p is the estimated error rate (QBER). Authors in [10] tried to reach the theoretical limit of protocol efficiency, and from these results, it is obvious that four iterations are sufficient for successful key reconciliation, as it was suggested in the original paper [6]. However, since the initial block length depends on the estimated error rate, it is necessary to perform all the iterations. The idea of usage dynamical initial block-size with Bayesian Networks and Cox Processes is explained in [11(p49)], but due to the simplicity we stay focused on initial fixed block size.

Let us go back to the parity check results. If the parity of a block disagrees between Alice and Bob, they perform a binary search on that block with the aim of identifying the single bit error. The binary search consists of dividing the block in half and comparing the parity check results for the divided block until the error is located. This means a maximum of $1 + \lceil \log_2 k_i \rceil$ parity bits are exchanged for each block with an error bit since $\lceil \log_2 k_i \rceil$ is the maximum number of times block k_i can be divided, and one parity bit is exchanged for blocks without errors. In order to minimize the amount of information gained by Eve, it is advisable to discard the last bit of each block and sub-block for which the parity bit was exchanged. Now if we define L_i as the maximum number of leaked bits, and k_i as the length of the block in the i^{th} iteration, it is clear that:

$$\sum L_i = \sum_i \left(\sum_{\substack{initially \\ even\ blocks}} 1 + \sum_{\substack{initially \\ odd\ blocks}} \left(1 + \lceil \log_2 k_i \rceil\right) + \sum_{\substack{other \\ errors\ corrected}} \lceil \log_2 k_i \rceil \right) \quad (3)$$

According to the results from [12], this can be shorted to:

$$L = \sum L_i = \sum_i \left(\frac{n}{k_i} + \sum_{\substack{errors \\ corrected}} \lceil \log_2 k_i \rceil \right) \quad (4)$$

where $k_i = 2k_{i-1}$, $k_i < \frac{n}{2}$ and n is the length of the initial key. Now it is clear that the number of leaked bits depends on the initial block size and error rate. From the results presented in [10], we can conclude that the majority of errors are corrected in the first two iterations.

Table 1. Percentage of corrected errors per iteration

Iteration	I	II	III	IV
Percentage of corrected errors	54.5223%	45.3478%	0.4517%	0.002%

We mark the length of the key after the key reconciliation phase as F.

PRIVACY AMPLIFICATION

Alice and Bob finally have an identical key without errors, but Eve may have gained significant knowledge of the key due to information leakage in the quantum transmission phase which is unavoidable. In order to reduce this leakage, Alice and Bob should delete some of the bits of the final key and strengthen their privacy. So, the final key is shortened even more. The number of rejected bits during the privacy amplification process is defined in Equation (5) [13], where S is the minimal number of bits that need to be discarded and *n* is the length of the key (*B*).

$$\frac{n \cdot 2^{-S}}{\log 2} < 1 \quad (5)$$

We mark the length of the key after this phase as *P*.

AUTHENTICATION

Due to intended level of security of QKD which is unconditional (Information-theoretical) security in the presence of an adversary with unlimited computer power and memory, in QKD no restriction's is put on adversary's computational power and storage capability. Taking into account that QKD uses two channels, quantum, and public channel, it is necessary to protect communication from Eve's influence. Public channel without authentication is like any other channel susceptible to a man-in-the-middle attack. In QKD, there are two types of authentication: immediate[iii] authentication and delayed authentication. Immediate authentication implies the authentication of messages immediately after they are received while the delay authentication implies the authentication for all messages exchanged during the session together to be done at the end of the session. There are variations in the details, but all QKD protocols contain authentication. In this article, we follow the approach from [11] where authentication is performed two times. The first time, before error correction phase, where Alice and Bob authenticate the outcome of the measurement. This authentication is necessary to prevent intercept/resend attack [11]. Finally, the authentication is done at the end of the session in order to verify that the key is indeed identical on both sides.

Author in [14] divided authentication schemes into two categories: Information-theoretically secure (ITS) and computationally secure message authentication schemes. Then author performed a comparative analysis of Wegman-

[iii] Often named as „continuous authentication". More details about the continuous authentication can be found in [17]

Carter, Sinson, den Boer, Bierbrauer et al., Krawczyk and a novel authentication scheme. In his document, author showed that Wegman-Carter [15] authentication which is based on ASU_2 (Almost Strong $Universal_2$) hashing is very well suited for authentication in QKD. To perform authentication it is necessary to sacrifice a certain part of the key and an upper bound for the key needed for authentication is defined with by the following equation [14]:

$$4 \times ((b + \log_2 \log_2 a) \times \log_2 a) \tag{6}$$

where *a* is the length of the message which needs to be authenticated and *b* is the length of authentication tag.
Finally, it means that we need to exchange one authentication message to verify measurement on quantum channel where the length of the message which needs to be authenticated is $\log_2 Q$ –bits, and it is necessary to verify key of length P. The amount of key which need to be sacrificed for authentication is shown in (7)

$$k_{auth} = 4 \times \left((b + \log_2 \log_2 \log_2 Q) \times \log_2 \log_2 Q\right) + 4 \times ((b + \log_2 \log_2 P) \times \log_2 P) \tag{7}$$

Now we can compare the length of the key from each of previous steps:

$$Q > B > R > F > P > A \tag{8}$$

Finally, Equation 8 shows that the length *Q* of the original key must be significantly longer than the key *A* after being reduced in all phases explained above.

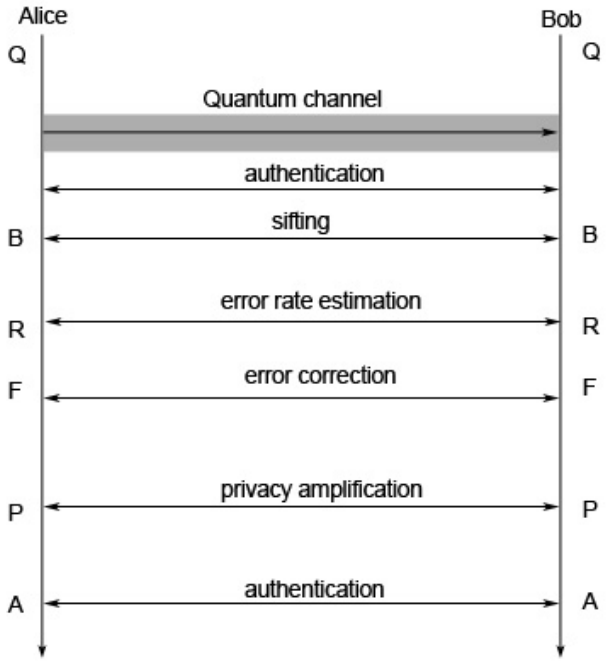

Figure 2. The sequence of operations in B92 protocol

CALCULATIONS

The length of the final key A can be calculated using Equation 9:

$$A = 0.25 \times Q - \frac{\log \left\lceil \frac{0.25 \times Q}{\log 2} \right\rceil}{\log 2} - \hat{S} \times Q - L - k_{auth} \tag{9}$$

Where
$\hat{S}$ – is the percentage of raw key (Q) used for calculating QBER
L - is the number of bits leaked during the key reconciliation phase
A - is the length of the final key

From Equation 9 it is easy to see that the noise in the quantum channel is included only in the calculation of the number of leaked bits L during the error reconciliation phase. Also, the parameter $\hat{S}$ has a significant impact on the final length of the key, but with the increase of $\hat{S}$ the possibility that Eve gain information about the key decreases.

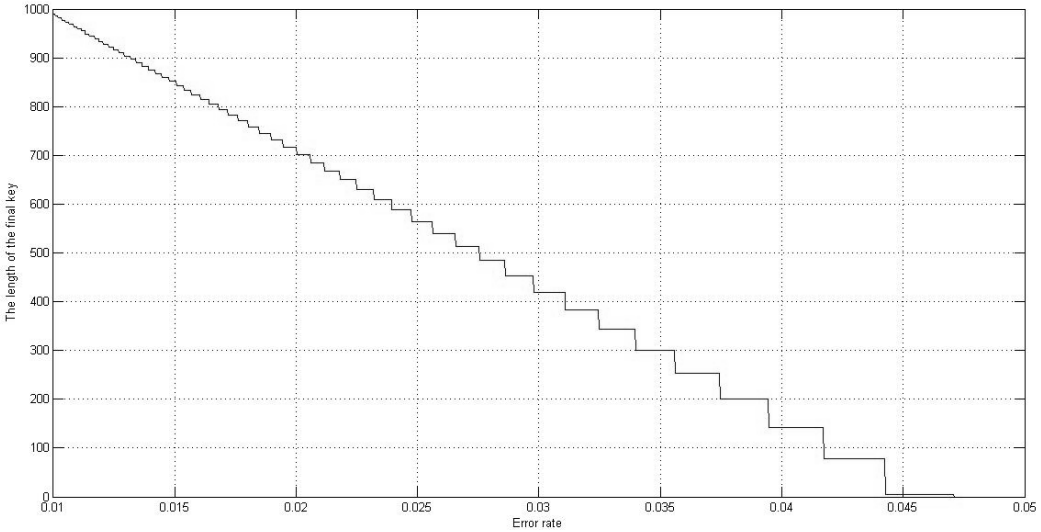

Figure 3. The length of the final key depending on the error rate; $\hat{S}$ = 0.01 (8.16%), Q=10 000

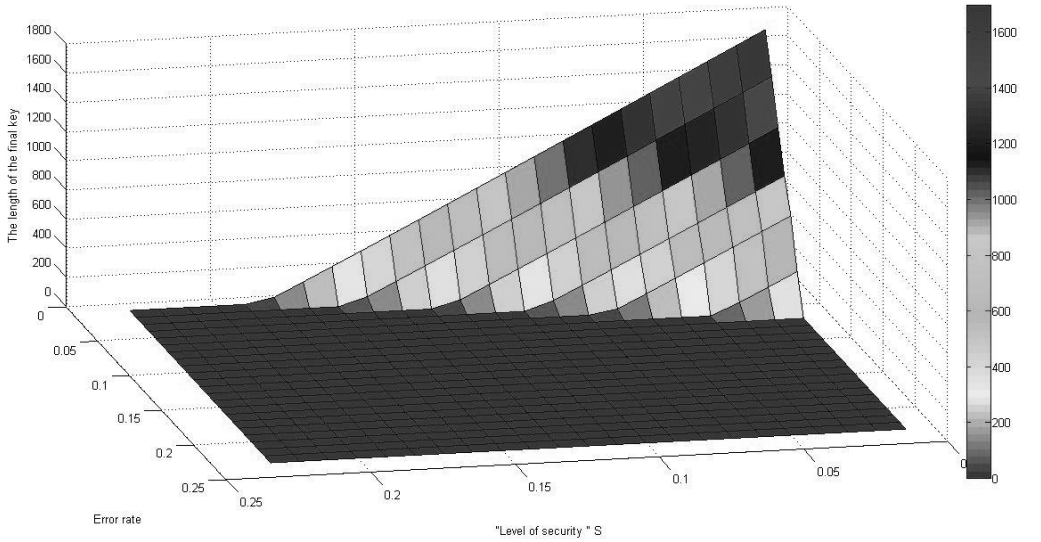

Figure 4. The length of the final key depending on the parameter $\hat{S}$ and error rate; Q=10 000

CONCLUSION

In this article we present the Equation (9) which is used to calculate the length of the final key A based on the length of the raw key Q, error rate in quantum channel and on security parameter $\hat{S}$. From Equation (9), it is clear that the length of the final key increases with the parameter error rate (p) and the "level of security" $\hat{S}$. In [16] and [11] authors defined the upper bound for tolerated quantum bit error rate p_{max} of 12.9% for BB84 protocol. However, we showed empirically that this error rate for B92 protocol is not acceptable. For minimal value of parameter $\hat{S}_{min}$= 8.16% the maximal error rate is 4.73% as it is shown in Figure 3. This value is the direct consequence of twice smaller number of photons that Bob detects in relation to the BB84 protocol.

From Figure 4, we can conclude that the length of the final key depends much more on error rate in the quantum channel. Parameter $\hat{S}$ influences much less on reducing the key length compared to error rate. During the measurements, the length of the authentication tag was 32 bits, which is the common length for authentication tag in practice.

It is important to underline that parameter $\hat{S}$ defines the amount of key which is needed to estimate QBER in the channel. Also, it is worth noting that the influence of eavesdropping is not included in the Equation (9) since the entire QKD process will be repeated if the estimated QBER is higher than maximally tolerated QBER.

ACKNOWLEDGEMENT

This research was funded by the grant of Technology Agency of the Czech Republic TF01000091 and partially was supported by the project SGS reg. no. SP2015/82 conducted at VSB-Technical University of Ostrava, Czech Republic.

REFERENCES

[1] Wiesner, S., "Conjugate Coding," ACM Sigact News 15, 78–88 (1983).

[2] Bennett, C. H.., Brassard, G., "Quantum Cryptography: Public Key Distribution and Coin Tossing," Proceedings of IEEE International Conference on Computers, Systems and Signal Processing 175(150), 8 (1984).

[3] Bennett, C. H., "Quantum Cryptography Using Any Two Nonorthogonal States," Physical Review Letters 68(21), 3121–3124 (1992).

[4] Lomonaco, S. J., "A Quick Glance at Quantum Cryptography," Cryptologia 23(1), 1–41 (1999).

[5] Niemiec, M.., Pach, A. R., "The Measure of Security in Quantum Cryptography," 2012 IEEE Global Communications Conference (GLOBECOM), 967–972, Ieee (2012).

[6] Brassard, G.., Louis, S., "Secret-Key Reconciliation by Public Discussion," Advances in Cryptology - EUROCRYPT93 765, 410–423 (1994).

[7] Nguyen, K., Van Assche, G.., Cerf, N. J., "Side-Information Coding with Turbo Codes and its Application to Quantum Key Distribution," Proceedings of the IEEE, 3 (2004).

[8] Elkouss, D., Martinez-Mateo, J.., Martin, V., "Information Reconciliation for Quantum Key Distribution," 14 (2010).

[9] Elkouss, D., Leverrier, A., AlÍeaume, R.., Boutros, J. J., "Efficient Reconciliation Protocol for Discrete-Variable Quantum Key Distribution," IEEE International Symposium on Information Theory - Proceedings, 1879–1883 (2009).

[10] Sugimoto, T.., Yamazaki, K., "A Study on Secret Key Reconciliation Protocol," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E83-A(10) (2000).

[11] Kollmitzer, C.., Pivk, M., [Applied Quantum Cryptography], Springer Science & Business Media (2010).

[12] RUTH II-YUNG, N., "A Probabilistic Analysis of Binary and Cascade," math.uchicago.edu (2013).

[13] Niemiec, M., "Design , Construction and Verification of a High-Level Security Protocol Allowing to Apply the Quantum Cryptography in Communication Networks," PhD Thesis, AGH University of Science and Technology, Krakow, Poland (2011).

[14] Abidin, A., "Authentication in Quantum Key Distribution : Security Proof and Universal Hash Functions," PhD Thesis, Linköping University (2013).

[15] Carter, J. L.., Wegman, M. N., "Universal classes of hash functions," Journal Of Computer And System Sciences 18(2), 143–154 (1979).

[16] Smith, G., Renes, J.., Smolin, J., "Structured Codes Improve the Bennett-Brassard-84 Quantum Key Rate," Quantum Physics, Physical Review Letters 100(17), 170502 (2008).

[17] Gilbert, G.., Hamrick, M., "Practical quantum cryptography: A comprehensive analysis (part one)" (2000).