Math-Net.Ru

Общероссийский математический портал

D. A. Kronberg, Новые методы исправления ошибок в квантовой криптографии с использованием LDPC-кодов, *Матем. вопр. криптогр.*, 2017, том 8, выпуск 2, 77–86

DOI: <https://doi.org/10.4213/mvk225>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.172.240.79

26 апреля 2023 г., 11:55:38

New methods of error correction in quantum cryptography using low-density parity-check codes

D. A. Kronberg

Lomonosov Moscow State University, Moscow

Получено 18.III.2016

Abstract. The problem of error correction in quantum cryptography is considered, including the estimation of error rate. We show that low-density parity-check (LDPC) codes are appropriate for this problem, and propose some modifications to achieve better code performance, taking into account the special properties of quantum cryptography.

Keywords: quantum cryptography, error correction, coding theory, LPDC-code

Новые методы исправления ошибок в квантовой криптографии с использованием LDPC-кодов

Д. А. Кронберг

Московский государственный университет им. М. В. Ломоносова, Москва

Аннотация. В работе рассматривается задача коррекции ошибок в квантовой криптографии совместно с оценкой уровня ошибок. Показано, что для решения такой задачи подходят коды с малой плотностью проверок на четность (LDPC-коды). С учетом особенностей квантовой криптографии предлагаются некоторые модификации применения этих кодов, повышающие их эффективность.

Ключевые слова: квантовая криптография, коррекция ошибок, теория кодирования, LDPC-код

1. Introduction

The goal of quantum cryptography is to distribute a secret key between two distant participants, usually referred as Alice and Bob, where Alice sends information to Bob. The key secrecy relies on fundamental laws of quantum mechanics, which lead to two basic statements: first, one cannot make a copy of an arbitrary quantum state, and second, one cannot get full information from an ensemble of nonorthogonal quantum states. This means that every eavesdropping attempt (performed by the third participant, Eve) may be detected by the legitimate users, and Eve's information may be estimated by the channel error rate.

There are four stages of key distribution in quantum cryptography.

1. Sending information via quantum channel. Alice sends quantum states to Bob, and Eve can perform any actions with these states, such as measurement, separation by beam splitter, entangling with her particles etc. At the end of this stage Alice and Bob get raw key, which includes errors and is correlated with what Eve had obtained.
2. Error estimation. Eve's information depends on the protocol parameters and output error rate, and legitimate users can estimate this information. If error rate is too high, Eve can get all the information, and the protocol aborts.
3. Error correction in the raw key. After this stage the key has no errors, but still is correlated with Eve, with also some information leaked during error correction.
4. Privacy amplification. By applying 2-universal hash functions Alice and Bob can obtain a shorter string with arbitrary small Eve's information about it.

Here we focus our attention on the second and third stages, and the problems of error estimation and correction will be considered. As our goal is to obtain a secret key, the leakage of information should be as small as possible. Thus, the following should be taken into account for error correction in quantum cryptography:

1. There is no message coding, since the raw key appears after quantum stage and is random.
2. The leakage of the information should be small and its value will be used at the stage of privacy amplification.
3. The error rate depends on the eavesdropper as well, and is unknown for the legitimate users, thus they need to estimate error rate and be able to perform correction for not accurate error value.
4. The rate of raw key generation depends on equipment and cannot be very high.

Here we will show the solution of this problem by using low-density parity-check (LDPC) codes which were introduced by Gallager in 1962 and became more popular in 1990s.

2. Elements of linear codes theory

Let us now provide a brief overview of the linear codes theory (see [2, 3] for details). Linear code is the set of bit strings that form linear subspace of the Hamming space. This subspace may be described as the set of solutions of homogeneous system of linear equations, with the number of equations M smaller than the number of variables N . The $M \times N$ matrix H of this system is called the *check matrix* of the code. The codeword t is the word of length N that satisfies

$$Ht = 0 \bmod 2. \quad (1)$$

If the rank of H is equal to the number of rows M , then the number of codewords is 2^{N-M} , and every information message of length $N - M$ may be encoded. The *code rate* R is given by the relation

$$R = \frac{N - M}{N}. \quad (2)$$

Noisy channel adds some noise to the codeword, it may be described as the addition of noise vector n :

$$r = (t + n) \bmod 2.$$

The condition (1) for the noisy word r now does not hold:

$$Hr = Ht + Hn = Hn \bmod 2 = z.$$

The value z is called the *syndrome* of the error, and if one can find error vector n by the syndrome, the error may be corrected.

One can compare the performance of codes with some given check matrices of size $M \times N$ for different values of syndrome length M . The longer the syndrome is, the more errors may be corrected, and the smaller is the code rate, according to (2). Below we will consider error estimation and the choice of appropriate check matrix with the optimal value of M .

Finding the error vector by the syndrome is a nontrivial problem for long codewords, since the number of possible syndromes grows exponentially. LDPC codes are linear codes with sparse check matrix, the typical number of 1s in every column is usually smaller than 10. The sparseness of the check matrix makes it possible to apply fast iterative decoding algorithms, see [3, 4] for details.

3. Raw key error correction with linear codes

The raw key in quantum cryptography appears after measurement of quantum states and removal of inconclusive outcomes, thus it is considered to be random. Hence, Alice and Bob cannot perform encoding, and (1) is not satisfied. Let us here introduce a slightly different code definition: a code is the set of vectors $\{v_i\}$ which have the syndrome s for the matrix H :

$$s = Hv_i \quad \forall i.$$

Now, let us consider the error correction procedure for this situation. After the quantum stage (i. e. after the sending of quantum states, their measurement by Bob and discarding the positions with inconclusive outcomes) Alice and Bob have the words v^A and v^B respectively, both of length N . These words may be different, and we consider Alice's word as the correct one. So, we have to correct errors in the Bob's word, i. e. find the error vector n such that $v^B = v^A + n$ holds.

To do this, Alice computes the syndrome of her word

$$s^A = Hv^A, \quad (3)$$

sends this syndrome to Bob and therefore declares the set of words which have syndrome s^A as the set of codewords. Then Alice will have a codeword, as if an encoding was performed. Such a code is indeed not a linear one and is a shift of a linear code to the fixed vector v^A .

Bob also computes the syndrome s^B of his string v^B

$$s^B = Hv^B,$$

and one can see that, for error vector n and $v^B = v^A + n$

$$s^B = H(v^A + n) = Hv^A + Hn = s^A + Hn.$$

Hence, the error syndrome Hence, the error syndrome for error vector n is given by

$$s = Hn = (s^B - s^A) \bmod 2,$$

and Bob's task is again to find error vector n by the syndrome s , it is the same task as for ordinary error correction. Bob can do this after getting syndrome s^A from Alice.

Since the only thing that Alice sends to Bob is her syndrome (computed according to (3)), one can easily find the number of disclosed bits: it is the length of syndrome, i. e. the number of rows in the check matrix (if it has full rank). Every syndrome bit is a parity bit for the positions with 1s in the corresponding row of the check matrix.

Let us note that while classical error correction deals with code rate, which depends on the syndrome length, in error correction for quantum cryptography the more important thing is the leakage of information, which is equal to the syndrome length if all the rows of check matrix are linearly independent.

The raw key length was N , and this length is decreased due to the leakage. Hence, if the error correction was successful, the new key length after error correction is given by

$$l = N - M. \quad (4)$$

It is not the final key length, because it does not include a key compression after applying 2-universal hash function. But here we do not consider this stage of quantum cryptography, and the task of legitimate users is to make the key length after error correction (4) as long as possible.

4. Decoding with extra bits request

The priority in quantum cryptography is the disclosure of the smallest possible number of key bits during the error correction stage, thus the number of rows in the check matrix should be small; on the other hand too high code rate often leads to fails in decoding attempts, and one would like to repeat the decoding with larger number of check matrix rows. In such situations the following algorithm may be more efficient: in case of aborted decoding procedure Alice and Bob add one row to their check matrix, and Alice sends one extra syndrome bit. Then Bob performs one more decoding attempt and in case of a fail requests one more bit from Alice. This process should be repeated until the successful decoding or while there are extra rows for check matrix.

Here we use an important property of LDPC-codes that the sparseness of check matrix is almost the only condition for the efficiency of decoding algorithm. It means that in the case of fail one can try to repeat the procedure with a check matrix containing some extra rows, and the decoding procedure does not become more complex.

The key length after disclosure of M syndrome bits and forwarding of extra n_E bits in the case of successful decoding (the success probability is higher then success probability without extra bits request) is

$$l = N - M - n_E. \quad (5)$$

To illustrate this method, we have considered the check matrix of size 1200×1800 and applied the decoding procedure 500 times for every error value. At first, we tried to perform decoding with the check matrix containing only the first 900 rows, i. e. for the matrix of size 900×1800 , which is the part of the original matrix.

If the decoding was successful, the key length equals 900, otherwise we add one row to the matrix and perform the decoding attempt again etc. If the decoding was successful for some number of extra bits n_E , the key length according to (5) equals $900 - n_E$. Eventually, if the decoding failed for 1200×1800 matrix, the key length is zero.

Fig. 1 shows an average decoding rate, i.e. average value of $\frac{l}{N}$ for 500 experiments at each error rate. The situation of fixed check matrix of size 900×1800 is compared with extra bits request scenario. As one can see, the procedure with extra matrix rows yields higher decoding rate.

Fig. 1. The dependence of average decoding rate on the error probability with fixed check matrix (dashed line) and with extra bits request (solid line). Shannon bound is shown by the dotted line

5. Combination of LDPC- and repetition codes

Practical usage of LPDC-codes shows that they show good performance for the error rates up to 15%, and are not effective for higher error rates. At the same time the two-parametric phase-time coding quantum key distribution protocol ([5, 6]) may (under some conditions) guarantee secret key generation with bit error rate up to 50%, so one needs to perform error correction for high error rates as well.

Let us propose the method for error correction that combines LDPC-codes with repetition codes. If one needs to apply repetition code R_k to the word of length k , one can use the scheme described above: Alice first computes the syndrome of her word v_k^A with the repetition code check matrix, which is (for $k = 5$)

$$H_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The error syndrome for the repetition code of length k is given by $s_k^A = H_k v_k^A$.

Alice forwards this syndrome to Bob and, and he compares their syndromes, finding the most probable error vector. For the repetition codes, there are two possible error vectors, different in all positions. By his word v^B , Bob can compute the probability of each of the two error vectors n for the known error probability q : $p(n|q) = (1-q)^{1-w(n)} q^{w(n)}$, where $w(n)$ is the Hamming weight of error vector n .

This means that Bob has to choose the bit value, having their probabilities. It is exactly the problem for LDPC-codes. Thus, to combine repetition code with LDPC-code for the word of length kN one first applies N times repetition code R_k for each block of length k , and then applies LDPC-code with the number N of columns in the matrix to N values of bit probability. This solves the problem of high error rates. Let us note that repetition codes with even block length, which are impractical in traditional error correction, are also useful here, since the important thing for LDPC-code is the input bit probability, not the exact value.

6. Error estimation and code choice

As mentioned in the Introduction, at the beginning of error correction Alice and Bob do not know the error rate, and therefore they have to estimate it. A common way to do this is to look at the number of errors when Alice and Bob disclose some positions of their raw key.

Let the original raw key length be 2000 bits, and consider disclosure of 200 bits. We can create a number of LDPC-code check matrices with the number of columns equal to 1800 and different numbers of rows for different error values. Then the goal is to choose the appropriate check matrix (and repetition code block length, if it is used) by the number of errors to achieve the best key length. Let us do it in the following way: first we compute the error rate probability distribution, and then get the code with the best key length expectation for the given probability distribution.

For m disclosed bits with k errors, let us use the Bayes theorem to get the a posteriori probability density function of error rate q

$$p(q|k) = \frac{p(k|q)p(q)}{p(k)}, \quad (6)$$

here we need to know the a priori error probability density function $p(q)$, which may be obtained by the results of an experiment. We also assume the same error probability q for every position, then the conditional probability of k errors in m bits for the given value of error rate q is proportional to $p'(k|q) = C_m^k q^k (1-q)^{m-k}$, and the conditional probabilities $p(k|q)$ may be computed as normalized values of $p'(k|q)$. The more positions are disclosed, the less is the variance of error rate distribution (6), and the more accurate is the error rate estimate.

For every LDPC-code one can find the function of average decoding rate $r(q)$, depending on the error rate (two such functions are shown on Fig. 1). For small error rates the average decoding rate is equal to the code rate, and for high error values it is equal to zero. To choose the best code for given error probability one needs to find the code with highest expected average decoding rate, which is given by

$$\mathbf{E}r = \int_q r(q) p(q|k) dq$$

and depends on two parameters — disclosed positions number m and number of errors k . Thus, for every values of these parameters we can choose the optimal code.

Fig. 2 shows the probability distributions of $p(q|k)$ for two set of values of m and k and a uniform a priori error probability distribution.

Fig. 2. A posteriori error probability distribution density $p(q|k)$ if 10% errors observed in disclosed part of length $m = 50$ (solid line) and $m = 300$ (dashed line)

The optimal number of disclosed positions depends on a priori error distribution and is about 10% of the original raw key length.

Note that if a priori error distribution is not uniform and expected error value is low, a more powerful and suitable method of error estimation may be useful. Alice can disclose not the single bits of the raw key, but the parity bits of blocks, which length depends on the expected error rate. For example, for expected error rate close to 3% the blocks of length 10 can be useful, since parity bits give more precise information about low error, and since block parity bits can be later used in LDPC code, being included in check matrix, so one does not need to throw away the positions used for error estimation.

7. Conclusion

The problem of developing error correction module for quantum key distribution has been considered. The following characteristics of QKD system are important: first, because of low raw key generation rate the input sequence should be no longer than 2000 bits, and second, most frequent error rate values are between 3% and 6%. The existing key error protocols based on LDPC codes ([7–9]) show better results than method described here for higher sequence length (from 10^4 bits), which is not surprising due to Shannon theorem and the fact that good LDPC code should not have 4- and 6-cycles in its Tanner graph, the requirement that is easier to satisfy for large check matrix size. Error estimation is also easier for longer sequences.

Another common method for error correction in quantum cryptography is Cascade ([10–12]). In this method the sequence is divided into blocks, in each of them one error may be corrected on every iteration by the algorithm close to the Hamming code. Cascade method shows excellent performance for low error rate (below 2%), but is worse than LDPC codes for higher error values. Therefore the combination of Cascade and LDPC methods is of interest for future research.

Fig. 3 shows the computational results for uniform a priori error distribution compared with Shannon bound.

Fig. 3. Results of 400 computation for every error rate. The average decoding rate (solid line) and Shannon bound (dotted line) are shown

Acknowledgements

The author is grateful to S. N. Molotkov and A. N. Klimov for useful discussions. This research was supported by the Russian Science Foundation, project No.16-12-00015.

References

- [1] Gallager R., “Low-density parity-check codes”, *IRE Trans. Inf. Theory*, **IT-8** (1962), 21–28.
- [2] MacWilliams F. J., Sloane N. J. A., *The Theory of Error-Correcting Codes*, Amsterdam: North Holland, 1977, xx+762 pp.
- [3] MacKay D. J. C., *Information Theory, Inference and Learning Algorithms*, Cambridge: Cambridge Univ. Press, 2003, xii+628 pp.
- [4] Johnson S. J., *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*, Cambridge: Cambridge Univ. Press, 2009, 356 pp.
- [5] Molotkov S. N., “Cryptographic robustness of a quantum cryptography system using phase-time coding”, *Z. Eksper. Teor. Fiziki*, **133**:1 (2008), 5–24 (in Russian); англ. пер., *J. Exper. Theor. Physics*, **106**:1 (2008), 1–16.
- [6] Kronberg D. A., Molotkov S. N., “Security of a two-parameter quantum cryptography system using time-shifted states against photon-number splitting attacks”, *Z. Eksper. Teor. Fiziki*, **136**:4 (2009), 650–683 (in Russian); англ. пер., *J. Exper. Theor. Physics*, **109**:4 (2009), 557–584.
- [7] Elliott C., Colvin A., Pearson D., Pikalo O., Schlafer J., Yeh H., “Current status of the DARPA quantum network”. In: “*Defense and Security*”: Int. Society for Optics and Photonics, 2005, 138–149.
- [8] Elkouss D., Leverrier A., Alleaume R., Boutros J. J., “Efficient reconciliation protocol for discrete-variable quantum key distribution”. In: “*IEEE Int. Symp. Inf. Theory*”, 2009, 1879–1883.
- [9] Dixon A. R., Sato H., “High speed and adaptable error correction for megabit/s rate quantum key distribution”, *Scientific reports*, **4**:7275 (2014), <https://www.nature.com/articles/srep07275>.
- [10] Brassard G., Salvail L., “Secret-key reconciliation by public discussion”. In: “*EUROCRYPT'93*”, Lect. Notes Comput. Sci., **765**, 1993, 410–423.
- [11] Pedersen T. B., Toyran M., “High performance information reconciliation for QKD with CASCADE”, *Quant. Inf. and Comput.*, **15**:5-6 (2015), 419–434.
- [12] Martinez-Mateo J., Pacher C., Peev M., Ciurana A., Martin V., “Demystifying the information reconciliation protocol cascade”, *Quant. Inf. and Comput.*, **15**:5-6 (2015), 453–477.