UDC 621.3(045)

DOI: 10.20535/2523-4455.mea.201253

Optimal Low Density Parity Check Matrices to Correct Quantum Key Errors for QKD

B. O. Bilash, ORCID 0000-0002-1341-1920
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine

***Abstract*—In this paper, the parity-check matrices that can be used in low density parity check (LDPC) based error correction method for quantum key distribution are analyzed. The quantum key distribution system has inevitable errors in sifted key that must be corrected by an error correction algorithm to create a secure key. In this analysis, 1000-bit sifted keys are divided into 50 parts. The algorithm creates 50 syndromes corresponding to each part by multiplying 10 × 20 bit parity-check matrices. The algorithm sends the generated syndrome to the other side, which also divides the sifted key into 50 parts, creates a syndrome from each part, and compares with the received syndrome. If the syndromes are different, these sifted key parts are discarded. However, there may be situations where different parts may have the same syndromes. Therefore, it is necessary to find such an optimal matrix that removes the probability of getting the same syndromes at different parts of the sifted key.**

Key words — QKD; LDPC; error correction; parity-check matrix; post-processing

1. INTRODUCTION

A. Quantum key distribution.

Quantum key distribution (QKD) is a system that can securely share an identical key between two distant parties, Alice and Bob [1]. Unlike modern classical cryptographic protocols, such as RSA (Rivest–Shamir–Adleman), which is based on the practical difficulty of the factorization of the product of two large prime numbers [2], QKD protocol is based on the quantum mechanics.

Although the first BB84 protocol [3] for QKD was proposed in 1984, it is being actively researched. There are some subjects for research works on the QKD such as chip-scale system [4], long-distance communication [5], high secure key rate QKD [6]–[8], and efficient post-processing [9].

The main task of the post-processing is error correction to share an identical secure key with Alice and Bob. Therefore, it is necessary to explain where the errors occur. The unreliable quantum channel is named because photons may cause noise to change the photon's characteristics. There may also be errors when accepting photons by Bob and misreading the state of the photon. The single-photon detecter, which is an extremely sensitive component for detecting a single photon, has inevitably has some noises, such as Dark count, After pulse, also there is Cross talk from the other channel [10]. Theoretically, the quantum key is not safe when the quantum bit error rate (QBER) is more than 11%. Usually, in commercial QKD system, the QBER of the system is under control below 5% which must be corrected. That is why they use error correction at the post-processing stage. It should also be noted that at the post-processing stage and at subsequent stages, the exchange of information between Alice and Bob takes place via a classical channel, which with a high degree of probability can be considered as almost perfectly reliability. Therefore, the task is to correct the errors that occurred during the phase of the photon exchange.

B. Analysis of existing error correction methods.

There are some error correction methods to correct quantum bit error, such as Cascade, Winnow, Low-Density Parity-Check codes (LDPC). Those ideas are adopted from the classical error correction methods.

In the Cascade [11] protocol, in each pass, Alice and Bob agree on a random permutation that applies to their bits.

Winnow [12], like Cascade, breaks binary strings to match them into blocks, but instead of bug fixing using iterative binary bug fixing is based on Hamming code.

But these protocols work poorly over long distances, also with long-distance messages. It is necessary to use a protocol that would contain the check bits together with the main message at one time during transportation. On the contrary, the LDPC protocol has an advantage in the case of long distance communication [13]. Nowadays, the computing power and electronics have improved, so the LDPC protocol has a lot of attention and developments on the LDPC code [14], [15].

Figure 1 is the overall procedure of the QKD. At the error correction step, Alice and Bob have sifted keys, but they are slightly different from each other due to the background noises of the QKD system. The purpose of this phase is to reconcile the sifted keys so that they are the same and then pass them on to the next stage of privacy amplification. The main problem at this stage is that when transferring the sifted keys between Alice and Bob, it is necessary to protect them so that Eve could not

DOI: 10.20535/2523-4455.mea.201253

get the sifted key. To avoid this, you can use hashing protocols to protect the information being transmitted. But this approach is not rational because of increasing computing resources and the processing time for the hash function. That is why there is another approach for correcting errors and protecting information from Eve.

Consider the known methods for error correction. In [16], researchers propose a full-cycle creating QKD system. According to their error-correction step, one side of the fresh key encodes a syndrome and sends it to the other side. In this case syndrome is a hash-function. We cannot convert syndrome to original message. And if Eve has syndrome, she cannot do nothing with this syndrome. Using it, we are sure that when error correction step, Eve will not know the information about the original bits between Alice and Bob. The other side decodes the syndrome and compares the results with its own. If the results do not match, this key is discarded and a new one is sent. We assume under certain conditions even fresh keys are different, they can have the same syndromes. In this work, we will find optimal matrices, that let to us to do not have cases, where we will have the same syndromes.

2. *Common syndromes.*

In the classical LDPC applying, first, the parity-check matrix H is generated, in which the "1" are uniformly and very rarely located, and all other positions are "0". Such a matrix is not systematic. By using Gaussian elimination it is necessary to transform it into a systematic form $[-P^T \mid I_{n-k}]$, where I_{n-k} is a prime identity matrix, P^T — binary matrix (in binary codes $-P = P$). From matrix H we creating generate matrix $G = [I_k \mid P]$. Usually, one side, Alice, has an output message m of size k that is converted into a codeword c of length n by a matrix G. Therefore, the matrix G has dimensions $k \times n$. The k/n ratio is called the relative code transmission rate (or just code rate) [17]. Typically, this rate is 1/2, 2/3, 3/4 and so on. The codeword is sent to the other side, Bob, through the noise channel. Bob accepts a vector r, which may differ from the codeword by some number of bits. In a simple example, the resulting vector differs by one bit. Bob, using the matrix multiplication of matrix H and the resulting vector r, obtains a vector s called a syndrome: s = H*r. If, as a result, the syndrome has all zeros, then the resulting vector r has no errors and is equal to the codeword. Bob then decodes the vector into a message m that is equal to Alice's message. If the vector r, in the simplest case, has one error, then the syndrome s will coincide with the column of the matrix H, whose number will be the number of the error bit in the vector r.

In QKD systems, LDPC codes cannot be used as they would in classic applications. You cannot just send a sifted key because eavesdropper (Eve) can find out about the sifted key and then there is no point in creating a secure key. Therefore, it is necessary to come up with such a method of correcting the errors in Alice and Bob's sifted keys so that Eve cannot find out about them. In [16] it is suggested to create and exchange syndromes between the sides. This idea needs to be explained in more detail.

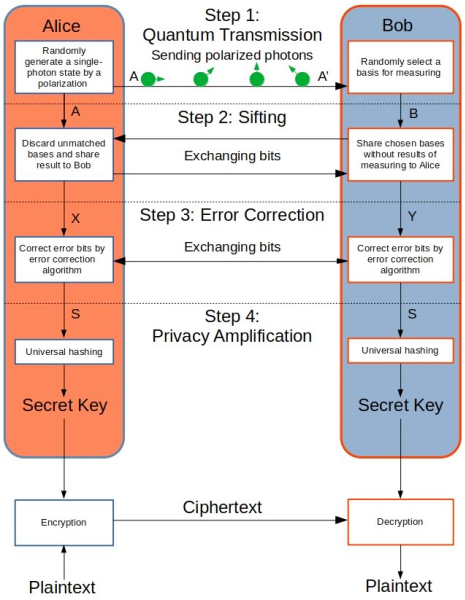

Figure 1. Overall procedure for the QKD

In Figure 1, in the first step, Alice encodes her information in the polarization of single-photon states and sends it to Bob. Bob detects single-photon states with the selected base and records the measurement result into classical bits. After that, Bob sends the chosen bases to Alice, who in turn discards the classical bits whose bases did not match with the chosen Bob [3]. As a result, at the beginning of step 3, Alice and Bob have sifted keys of the same length, but which differ by a certain percentage of bits. This is called the quantum bit error rate (QBER).

Drawing on the analogy of the classic error correction method, QBER is additive white Gaussian noise (AWGN).

In the case of QKD, the two sides already have sifted keys, X and Y, which are not the same and need to be fixed. Alice cannot send her sifted key to Bob, because Eve immediately finds out about it. Therefore, it is suggested to exchange syndromes. A syndrome is created by multiplying the sifted key by matrix H: $s = HX^T$. The proposed method does not require the creation of a matrix G. Unlike the classic case, the syndrome is unknown in advance (but can also be all zeros). It is not necessary to correct the sifted key X with the resulting s syndrome. This syndrome is sent to Bob. Eve, having received the syndrome has nothing to do with it. Bob creates his syndrome ŝ by multiplying his sifted key by the matrix H: $\hat{s} = HY^T$. The main idea is that if parts of the sifted keys are the same - the created syndromes will also be the same. If parts of the sifted keys are different, the syndromes should be different and these parts of sifted keys will be discarded.

In our research, it was decided to use a sifted key with a size of 1000 bits. Using LDPC matrix with 1000 + "parity bits" is not rational because it takes a lot of hardware resources. Instead, we propose to divide message for small parts and to create syndromes from each part independently for other parts. We used the standard matrix size at which the coding rate in the classical method is 1/2 and is one of common. To our

DOI: 10.20535/2523-4455.mea.201253

research we decided to investigate not big matrices. There are proposed to use 5 × 10, 10 × 20, 20 × 40 size matrices. In these cases, we divide our 1000-bit message for 100, 50, 25 parts with 10, 20, 40 bits in each part respectively.

We chose 10 × 20 matrix as optimal. The reasons for this choice will be explained later. In this case, the length of a part of the sifted key is 20 bits, and the created syndrome is 10 bits. Then the possible combinations of the sifted key may be 2^{20}, and the possible combinations of syndrome 2^{10}. In this case, one syndrome will respond to 1024 different messages. Therefore, when the messages are different, there may be situations where the syndromes will be the same. Then the error correction of the part will be failed. The solution may be to increase the length of the syndrome or to find a special H matrix, which, under certain conditions, will not cause common syndromes for different messages.

3. *Shared messages and their dependency on QBER.*

To analyze the error correction algorithm, proposed in [16], we arbitrarily selected messages of 1000 bits length. After that, we randomly changed the message bits depending on QBER. The simulation program was written in C language. The algorithm for adding error bits, for example, for QBER = 5% is as follows: 1) we generate a random number from 0 to 999 for each bit of the sifted key (1000 bits total), 2) if this number is less than 50 (in our case it is equivalent to 5% for QBER) - change this bit; if more - we save this bit. This does not mean that it will change exactly 50 bits in the message, but in Figure 2 it will be similar to a Poisson distribution with a mathematical expectation of 50 bits. For each QBER from 0.1% to 25% in 0.1% increments (1 to 250 bits, 1-bit increments), we repeated this procedure 100 times. Figure 2 lets us know, that we use Poison distribution to create errors in messages.

As a result, we have two messages: the original message, which is the sifted key of Alice, X, and changed to a certain number of bits equal to QBER, the message that is the sifted key of Bob, Y.

Next, we divide each sifted key from Alice and Bob into 50 parts, 20 bits each, and compare these parts. Then count the number of parts that are the same. Ideally, there will be 50 (all parts of one sifted key are equal to all parts of another sifted key).

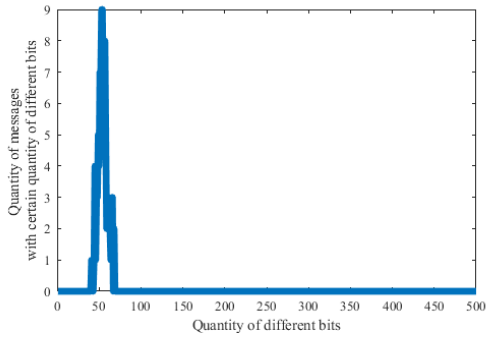

Figure 2. Changed bit distribution for 5% QBER

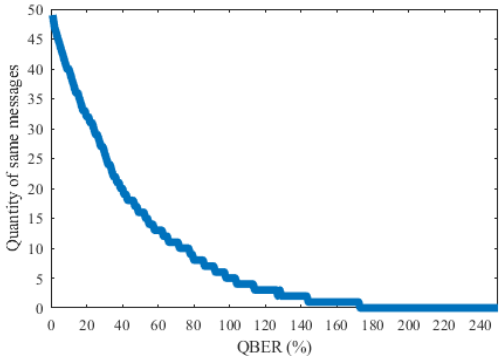

Figure 3. Identical parts of the message depending on QBER

Figure 3 shows the average number of messages that will be the same for different QBER. According to the graph, one can see that at a very small value of QBER most parts of the sifted key are the same. Every time, Alice and Bob exchange 50 parts of original message. If messagesare absolutely same, all 50 parts are also same. In this case, 0 messages will be discarded. But in real situation, if messages are different, some parts of messages and their syndromes are also different. For example, at QBER = 5%, on average only 16 parts from 50 are same between Alice and Bob. Other 34 parts have at least 1 different bits, and they will be discarded.

Consider other examples, 5 × 10, 20 × 40 size matrices. We added to messages errors and divided for 100 and 24 parts respectively. Results on Figure 4.

On Figure 4, first graph shows results for 10 × 20 matrix, same with Figure 3, but it is not on average. Second graph shows results for 20 × 40 matrix. Third graph shows results for 10 × 20 matrix.

Obviously, the best solution is to divide the message into 100 parts. Then there is one error in the message will be divided between two messages, one of which will be rejected and the other will be saved. The number of possible message combinations will be $2^{10} = 1024$, and the number of possible syndromes will be $2^5 = 32$. And each syndrome will correspond to 32 possible messages.

Then Eva will be able to more easily determine the possible combinations of codewords, which is dangerous, by the method of selection. On the other hand, if the length of the message is 40 bits, there will be 2^{40} possible combinations and 2^{20} possible syndromes. One syndrome will have more than one million possible messages, which will be a problem for Eva to find a possible codeword. However, with a small QBER, if there is only one error in the message - the whole message is rejected, it is not rational. From this point of view, the size of the messages chosen from the very beginning is the most successful compromise between the difficulty of Eva finding the original message and the number of rejected messages in the presence of an error. And a matrix of size 10 × 20 is the most optimal.

DOI: 10.20535/2523-4455.mea.201253

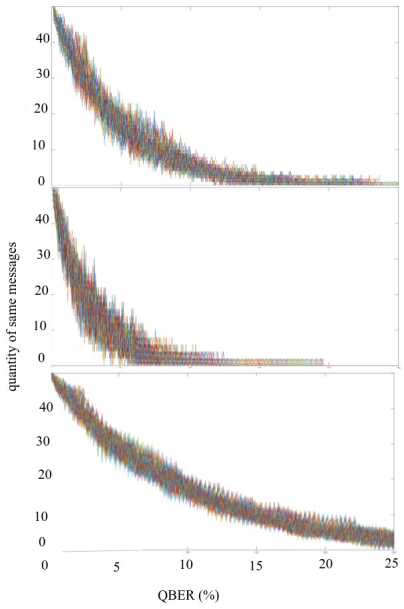

Figure 4. Identical parts of the message depending on QBER for 50, 25, 100 parts of message.

4. *Investigation of conflicting messages.*

As stated above, even if parts of sifted keys are different, there may be situations where syndromes are the same.

At QBER = 5%, we expect our sifted keys to be 50 bits apart. Of course, there may be a situation where these 50 bits are stocked at three parts, and then each part will contain at least 10 changed bits. But a more realistic situation is when these bits are distributed evenly between all parts. Then one bit will be changed in each part.

We created a new syndrome from the part of the sifted key and the matrix H, which is the same for all parts of the message. The matrix was created using the method suggested by David Mackay and Radford Neal in [14], [15]. Radford Neal has created open-source software [18]. We have integrated Neal's code to generate the H matrix into our code.

Since the created H matrix has dimensions of 10 × 20 bits, it is enough for checking all combinations of 20-bit messages. Therefore, the need to investigate all combinations of errors of the original key of 1000 bits in length makes no sense. It is enough to investigate all the combinations of a part of the message that has only 20 bits.

Since the H matrix must be sparse, the number of "1" in the matrix should at least not exceed the number of "0". Therefore, we have created 10 matrices using Neal's software, 5 matrices for "evencol" and 5 matrices for "evenboth" methods. In each matrix, the number of "1" in each column is from 1 to 5. The seed value of the random function is 1.

We used a message in which all 20 bits are zeros and changed the bits in every possible combination. First, we changed one bit in each possible position, then two, three, etc., to consider all possible combinations of the modified message. They then created the syndromes from these messages and compared them with the original message's syndrome.

We calculated the number of common syndromes varying a certain number of bits. The results showed that with the "evenboth" method and number of 1 in each column is 4, which is optimal H matrix for applying the QKD system because QKD operates properly QBER is under 5%. It lets to us to use optimal matrix that removes the probability of getting the same syndromes at different parts of the sifted key. In figure 5 we can see, that if we use this optimal matrix, we will not have same syndromes of messages are different when QBER is under 16%. In graph, if quantity is 50, all syndromes are different if messages are different. Only in two cases we have same syndromes if messages are different. The X-axis and Y-axis are changed bits from 0 to 20 and the number of common syndromes from the original and changed message, respectively.

According to the results, when the number of changed bits is less than 4, we do not have common syndromes in different messages. As mentioned above, errors are distributed in the sifted keys evenly, and therefore at QBER = 5%, from 1000 bits in the middle will change 50 bits. We have 50 parts of message, so in every part usually will change 1 bit. Our matrix will not have syndromes when change under 3 bits (15% from 20-bit parts of message). It is very rarely probability, that will change 4 or more bits, if QBER is under 5%. Result of same syndromes distribution to our matrix if on Figure 6.

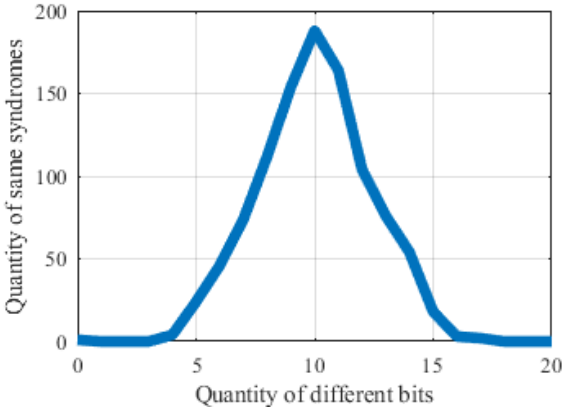

Figure 5. Distribution of the number of same syndromes versus the number of changed bits

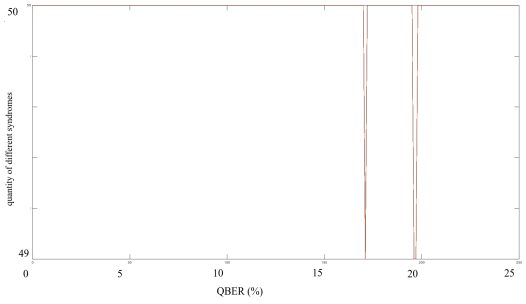

Figure 6. Error correction step

DOI: 10.20535/2523-4455.mea.201253

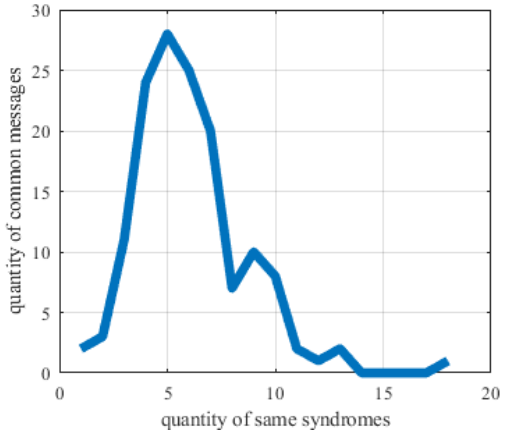

Figure 7. Number of common syndromes at 4 changed bits

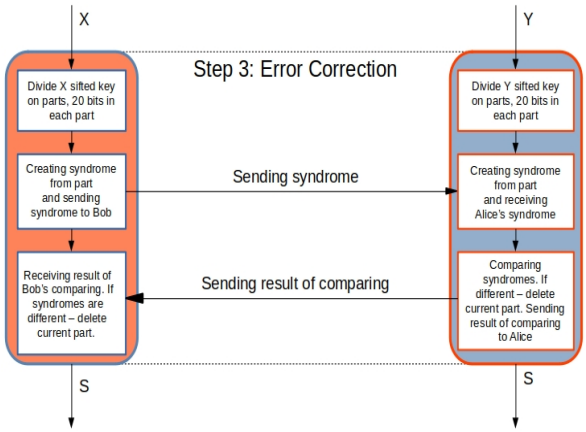

Figure 8. Error correction step

5. *Seed research for Neal's matrices.*

These results, as mentioned above, were obtained when the seed value for the generation of the matrix was 1. We conducted studies and created 1000 different matrices where the seed value was from 1 to 1000. Of these, 144 matrices did not have common syndromes, if they were only changed up to three bits in each message. So, theoretically, we can use those matrices to create syndromes to apply the QKD system.

Among these syndromes, when changing 4 bits into messages, there may be a common number of syndromes. For seed = 1, the number of common syndromes is 4. The total result for the 144 matrices found is in Figure 7.As you can see from the graph, the total number of common syndromes is 4-6, but there are matrices when the number of common syndromes is 1 which is enough for QKD condition.

6. *Results.*

The complete error correction process based on [16] is depicted in Figure 8.

Conclusion and discussion.

The main purpose was to find optimal matrices that can be used in the error correction method proposed in [16]. As can be seen from Figure 3, this method is not rational, but under certain conditions, it is easier to reject most of the messages than to correct them. It is not possible to use a message length equal to the length of the message because Eve can easily find out about the message by converting the syndrome back into a message. Therefore, the classic model is chosen when the length of the syndrome is twice less than the length of the message. If the length of the message is 20 bits and the length of the syndrome is 10 bits, then 1024 different messages will be needed for one syndrome, so it has robustness against Eve's attack. In addition, such messages in our case are 50 pieces, so the probability of finding the right message increases exponentially. But in this case, common syndromes may occur with different reports. You can increase the number of bits in a syndrome, or find a matrix in which the number of common syndromes under certain. The optimal matrix for the QKD system generates different syndromes under three erroneous bits. The generated 10 × 20 bit Neal's matrix, with 4 of "1" in each column, showed the best result in creating message syndromes to share an identical sifted key between Alice and Bob.

The process of discarding messages is not efficient enough, therefore the object of future research is to develop an algorithm for correcting errors in parts of the sifted key in which Alice's and Bob's syndromes are not matched.

References.

[1] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, no. 1, pp. 78–88, Jan. 1983, **DOI:** 10.1145/1008908.1008920.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, **DOI:** 10.1145/359340.359342.

[3] C. H. Bennett and G. Brassard, "BB84highest.pdf," *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*. pp. 174–179, 1984.

[4] P. Sibson *et al.*, "Chip-based quantum key distribution," *Nat. Commun.*, vol. 8, no. May 2016, 2017, **DOI:** 10.1038/ncomms13984.

[5] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, "Overcoming the rate–distance limit of quantum key distribution without quantum repeaters," *Nature*, vol. 557, no. 7705, pp. 400–403, May 2018, **DOI:** 10.1038/s41586-018-0066-6.

[6] Z. Yuan *et al.*, "10-Mb/s Quantum Key Distribution," *J. Light. Technol.*, vol. 36, no. 16, pp. 3427–3433, 2018, **DOI:** 10.1109/JLT.2018.2843136.

[7] H.-K. Lo, M. Curty, and B. Qi, "Measurement-Device-Independent Quantum Key Distribution," *Phys. Rev. Lett.*, vol. 108, no. 13, p. 130503, Mar. 2012, **DOI:** 10.1103/PhysRevLett.108.130503.

[8] C. H. Park *et al.*, "Practical plug-and-play measurement-device-independent quantum key distribution with polarization division multiplexing," *IEEE Access*, vol. 6, pp. 58587–58593, 2018, **DOI:** 10.1109/ACCESS.2018.2874028.

[9] B. K. Park, M. K. Woo, Y.-S. Kim, Y.-W. Cho, S. Moon, and S.-W. Han, "User-independent optical path length compensation scheme with sub-nanosecond timing resolution for a 1 × N quantum key distribution network system," *Photonics Res.*, vol. 8, no. 3, p. 296, Mar. 2020, **DOI:** 10.1364/PRJ.377101.

[10] T. A. Eriksson *et al.*, "Crosstalk Impact on Continuous Variable Quantum Key Distribution in Multicore Fiber Transmission," *IEEE Photonics Technol. Lett.*, vol. 31, no. 6, pp. 467–470, 2019, **DOI:** 10.1109/LPT.2019.2898458.

[11] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 765 LNCS, pp. 410–423, 1994, **DOI:** 10.1007/3-540-48285-7_35.

[12] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A - At. Mol. Opt. Phys.*, vol. 67, no. 5, p. 8, 2003, **DOI:** 10.1103/PhysRevA.67.052303.

[13] Gallager, “Low density parity check codes,” 1963.

[14] D. J. C. Mackay and R. M. Neal, “Good codes based on very sparse matrices,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995, vol. 1025, pp. 100–111, **DOI:** 10.1007/3-540-60693-9_13.

[15] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999, **DOI:** 10.1109/18.748992.

[16] N. Walenta *et al.*, “A fast and versatile quantum key distribution system with hardware key distillation and wavelength multiplexing,” *New J. Phys.*, vol. 16, 2014, **DOI:** 10.1088/1367-2630/16/1/013047.

[17] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge University Press, 2003.

[18] Radford M. Neal, “Software for Low Density Parity Check Codes.” 2012.

Надійшла до редакції 22 квітня 2020 року

DOI: 10.20535/2523-4455.mea.201253

УДК 621.3(045)

Оптимальні LDPC матриці для виявлення помилок в квантових бітах у системах QKD

Білаш Б. О., ORCID 0000-0002-1341-1920

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського" Київ, Україна

***Анотація*—В даній роботі проаналізовано матриці перевірки на парність, які можуть бути використані в методі виправлення помилок на основі low density parity check (LDPC) матриць в системах квантового розподілу ключів. Система квантового розподілу ключів має неминучі помилки в просіяному ключі, які повинні бути виправлені алгоритмом виправлення помилок для створення захищеного ключа. У цьому аналізі 1000-бітні просіяні ключі розділяються на 50 частин, по 20 біт в кожній частині. Алгоритм створює 50 синдромів, по 10 біт в синдромі, що відповідають кожній частині, за допомогою перемноження матриць перевірки на парність розміром 10 × 20. Матриці перевірки на парність створюються алгоритмом, запропонованим Девідом Маккеєм та Редфордом Нілом. Процес створення синдрому складається з матричного перемноження 20-бітної частини просіяного ключа на матрицю перевірки на парність. Алгоритм посилає сформований синдром другій стороні. Під час передачі підслуховувач може перехопити синдром, але він не може дізнатись точне повідомлення з синдрому, навіть якщо він володіє матрицею перевірки на парність теж. Друга сторона також ділить її просіяний ключ на 50 частин, створює синдром з кожної частини і порівнює з отриманим синдромом. Якщо синдроми різні, ці частини просіяних ключів відкидають. Однак, через те, що довжина повідомлень складає 20 біт, а довжина синдромів 10 біт, можуть виникати ситуації, коли різні частини просіяного ключа матимуть однакові синдроми. Для даного випадку кожні 1024 повідомлень будуть мати один спільний синдром. Тому необхідно знайти таку оптимальну матрицю, яка знімає ймовірність отримання однакових синдромів від різних частин просіяного ключа.**

Ключові слова — QKD; LDPC; корекція помилок; матриця перевірки на парність; пост-обробка