Journal of Information Security and Applications 78 (2023) 103581

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Analysis of multiple overlapping paths algorithms for secure key exchange in large-scale quantum networks

Mateusz Stępniak *, Jakub Mielczarek

Institute of Theoretical Physics, Jagiellonian University, Łojasiewicza 11, 30-348 Cracow, Poland

ARTICLE INFO

Keywords:
Quantum cryptography
Quantum key distribution
QKD network
Trusted repeater
Secure key agreement
Secret sharing

ABSTRACT

Quantum networks open the way to an unprecedented level of communication security. However, due to physical limitations on the distances of quantum links, current implementations of quantum networks are unavoidably equipped with trusted nodes. Consequently, the quantum key distribution can be performed only on the links. Due to this, some new authentication and key exchange schemes must be considered to fully benefit from the unconditional security of the links. One such approach uses Multiple Non-Overlapping Paths (MNOPs) for key exchange to mitigate the risk of an attack on a trusted node. The scope of the article is to perform a security analysis of this scheme for the case of both uncorrelated attacks and correlated attacks with finite resources. Furthermore, our analysis is extended to the case of Multiple Overlapping Paths (MOPs). We prove that introducing overlapping paths allows one to increase the security of the protocol, compared to the non-overlapping case with the same number of additional links added. This result may find application in optimising architectures of large-scale (hybrid) quantum networks.

1. Introduction

Public-key cryptography discovered in the 1970s of the last century provided a long-sought solution to the problem of secret key exchange [1]. The ingenious breakthrough allowed for the promotion of ideas such as secure communication on the Internet. The security of public-key cryptography is based on the high computational complexity of the problem used, such as factorisation of composite numbers.

When public-key cryptography has already been implemented on a global scale, it has been realised that the complexity of the problems used can be reduced if quantum computing resources are used [2]. This raised concerns about the security of public-key cryptography with respect to hypothetical quantum attacks. However, the solution to the problem already existed (theoretically) and relied not on computational complexity-based security but on information-theoretic security in short ITS. It is a type of security guaranteed by mathematical properties of information theory, such as Shannon entropy and mutual information. A limiting case of ITS is the *unconditional security* satisfied by the one-time pad (OTP) cipher, which uses a random key of the same length as the message to encrypt and decrypt it. The information-theoretic security is stronger than computational security, which relies on the computational hardness of certain problems, such as factoring large numbers.

ITS can be obtained within a use of the quantum key distribution (QKD) [3]. Quantum key distribution (QKD) is a method of securely exchanging encryption keys between two parties using quantum physics. It allows them to create and share a secret key that can be used to encrypt and decrypt messages. QKD works by sending polarised photons over a fibre optic cable and measuring their properties. It is provably secure because any attempt to eavesdrop on the key exchange will disturb the quantum state of the photons and be detected by the communicating parties.

Although quantum key distribution (QKD) is theoretically attractive, there are many technical difficulties, challenges, and open problems that need to be solved [4]. These challenges have precluded the wide implementation of QKD for decades, since its theoretical introduction in the 1980s. However, the situation has improved significantly in the last few years and QKD solutions are blossoming. One significant obstacle remains: the distance over which QKD can be performed. This is due to the suppression of photons in the optical medium, limiting practical QKD over ground-based optical fibre links to distances not longer than approximately 100 km [5].

One potential but costly and challenging solution to this problem is to use space. Due to the much weaker suppression of photons in air and cosmic vacuum, QKD can be performed at much longer distances [5]. Another potential solution is given by quantum repeaters, but the

* Corresponding author.
E-mail addresses: mateusz.stepniak@student.uj.edu.pl, mat.stepniak.algo@gmail.com (M. Stępniak).

https://doi.org/10.1016/j.jisa.2023.103581

Available online 29 September 2023

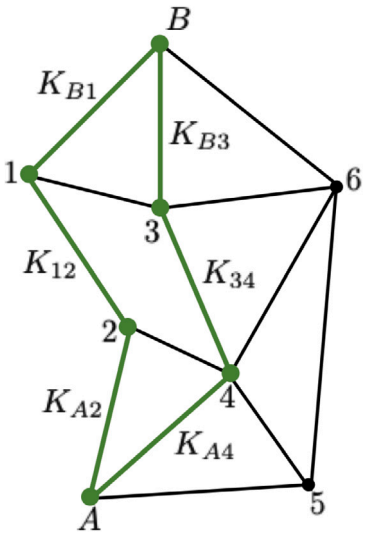

Fig. 1. Pictorial representation of the exchange of a secret key $K = (K_1, K_2)$ shared in two paths connecting the communicating nodes A (transmitter) and B (receiver). Here, the part K_1 is exchanged via the nodes 1 and 2, while the part K_2 is exchanged via the nodes 3 and 4. In the first step, the partial keys K_1 and K_2 are sent from the node A to the nodes 2 and 4 respectively. In both cases, the OTP encryption with the QKD keys (K_{A2} and K_{A4}) are used. The procedure is then continued, via the nodes 1 and 3, to the end node B.

technology is not mature enough to be implemented in the present realisations of the QKD solutions.

As a consequence, current implementations of large-scale quantum networks, which are systems that enable secure quantum communication among multiple nodes over long distances, must use classical trusted nodes. Beside problem with link range, these networks face many challenges, such as preserving quantum coherence, and ensuring interoperability and scalability. They comprise both ground-based and complementary satellite segments, each playing a crucial role in its overall operation. Examples of experimental realisations of such networks are the Tokio network [6], Beijing-Shanghai network [7], and Madrid network [8].

However, the existence of the classical nodes in these large-scale quantum networks raises security concerns. Although the QKD link has been shown to be ITS-safe, nodes can become a source of information leakage. The purpose of this article is to present a scheme that will significantly improve the security of these hybrid QKD networks.

In particular, at this point QKD algorithms are also potentially vulnerable to the man-in-the-middle (MITM) attacks. Usually, this problem is resolved by applying the authentication of the classical nodes. Since the idea of employing QKD is to eliminate non-ITS protocols, the authentication of QKD nodes must be performed with the use of the Wegman-Carter protocol, which has been proven to be of the ITS class. More information about QKD authentication can be found in Ref. [9]. Although the authentication problem can be successfully resolved in this way, this does not concern end-to-end encryption (E2EE).

From a theoretical point of view, using QKD combined with a one-time pad (OTP) guarantees end-to-end ITS communication. However, with actual limitations on the key exchange rate, this approach is too slow to be used instead of classical communication. An alternative to OTP is to use weaker (non-ITS) symmetric cryptography algorithms that are generally resistant to quantum attacks [10]. This seems to be the way the QKD network could be used. A completely different direction for preparing for quantum attacks, which does not rely on the QKD network, is post-quantum public-key cryptography, which has been intensively developed in recent years [11].

In this work, we consider the potential vulnerability in the quantum network based on trusted nodes with a Multiple Path Scheme [12]. This scheme is a way of distributing quantum keys over a hybrid quantum network with classical nodes that may have different levels of trustworthiness. The scheme involves sending secret keys along different paths, thereby increasing the security or efficiency of the communication network. We construct two models of possible attacks: the uncorrelated attack, in which each node has a certain probability of becoming compromised, and the correlated attack, in which the opponent owns certain resources that could be used to compromise the security of certain nodes.

Furthermore, based on [13], we extend the QKD multiple-path distribution protocol, assuming that the paths can overlap. Our analysis shows that to improve security, it is (under certain conditions) more optimal to add interlinks between disjoint paths, instead of adding a new path. A similar concept using overlapping paths has been presented in [14], where security is improved by introducing complete subgraphs ("cities"). While preparing this article, another work has appeared that addresses the security issue of overlapping multiple paths [15].

These works explore different approaches to improving the efficiency and security of quantum key distribution through the use of multiple paths. They build on the findings of earlier studies and offer new insights and solutions to challenges in this field.

2. The multiple paths protocol

One of the central concepts behind the design of a telecommunication network is redundancy. To marginalise the probability of a lack of connectivity between two nodes, there must be at least two alternative paths that connect an arbitrary two nodes. Here, we assume that the same must concern QKD networks, in particular the hybrid QKD networks under consideration. This approach has been proposed in [12] and recently explored in [16], in which the optimal flooding of key elements is considered.

Now, to help the reader better understand further material, we introduce problems concerning security and concepts of multipath protocol. Consider a simple uncorrelated attack scheme, in which the possibility of a successful attack on any node is p and n is the total number of intermediate nodes on the path (excluding communicating nodes/parties). The probability that at least one node has been successfully attacked is $P_1 = 1-(1-p)^n \approx np$, where the approximation is valid for $p \ll \frac{1}{n}$. So roughly, the probability of an attack on the network grows with the number of nodes. If the network is a hybrid QKD network, this would be equivalent to leaking a secret key exchanged via the attacked node.

Following this simple model, we find the probability that at least two nodes have been attacked: $P_2 \approx (np)^2$. Therefore, under the condition $p \ll \frac{1}{n}$, the probability of a successful simultaneous attack on at least two nodes is quadratically lower than in the previous case.

Following the above observations, let us consider a scenario in which a secret key K is composed of two parts K_1 and K_2 of equal length. For example, if the key K is devoted to be applied in the symmetric AES-256 algorithm, both parts K_1 and K_2 are 256 bits long. Now, we require that knowing one of the keys gives us zero knowledge of the key K. This requirement can be easily satisfied by the One-Time Pad (OTP) applied to the two parts K_1 and K_2, so that:

$$K = (K_1, K_2) = K_1 \oplus K_2, \tag{1}$$

where $\oplus$ is the XOR operation (addition modulo two).

Now, let us suppose that K_1 and K_2 are two bit strings that are distributed using two different paths in the QKD network. The two paths connect two parties (nodes A and B), between which the key K is exchanged (see Fig. 1). For every two adjoint nodes in the network (i, j), a secret key $K_{ij} = K_{ji}$ is established through QKD. Then, in a *hop-by-hop* approach, if node i wants to send a secret message M to node j, the OTP encryption is used by evaluating the cipher $C = M \oplus K_{12}$. The classical ciphertext C is transmitted to the subsequent node via

the classical (untrusted) channel. Then, by evaluating $M = C \oplus K_{12}$, the ciphertext is decrypted at the node j.

The protocol introduced in this section quadratically improves the security of secret key exchange in a hybrid QKD network. However, this is achieved by the cost of doubling the number of keys exchanged. Therefore, we reduce the performance of the system by a factor of two. However, the quadratic improvement by the linear cost seems to be a beneficial solution.

The decomposition of the secret key into two parts is an example of secret sharing. It is worth emphasising that the idea can be generalised by splitting the secret key into three or more constituents. In this case, Eq. (1) generalises to:

$$K = (K_1, K_2, \dots, K_N) = K_1 \oplus K_2 \oplus \cdots \oplus K_N, \tag{2}$$

where N is the number of different paths. Realisation in the case with $N > 2$ allows for a further reduction in the probability of attack at $\sim (np)^N$. However, this is due to the cost of both the much more complex topology of the QKD network and its lower performance. In the next sections, we perform a detailed analysis of two types of attack on trusted nodes and in Section 4 we present the concept in which key exchange can be done with paths that cross each other, along with the discussion of performance of this new protocol.

3. Security considerations on multiple non-overlapping paths

We consider two models of possible attack, first for an uncorrelated attack that simulates leaking and publishing information (secret key) steaming from random failure of certain nodes, and second for a correlated attack where the party is assumed to have certain resources that can be used to take over some nodes and gain information. We model the QKD network using an unweighted graph, where two communicating nodes can always be connected by a certain number of disjoint paths. This is justified as a desired property of a real telecommunication network [17].

3.1. Uncorrelated attacks

3.1.1. General formulation

Given a graph with two distinguished nodes named A and B (Alice and Bob), the ith node is marked with a certain number p_i representing the probability that during the protocol the ith node will be hacked and publicly reveal secret keys ($p_i = 0$ represent complete trust, while $p_i = 1$ represent fully corrupted node). A and B can be connected with some disjoint paths, and to compromise the security of the protocol, at least one node on each path must be untrusted. The task is to find a system of disjoint paths between A and B that minimise the probability of hacking communication.

For a pre-defined system, calculating the probability is an easy task. Let $\mathcal{R}$ be the family of all paths in the solution (where the path is considered to be a set of intermediate nodes, *i.e.* excluding the A and B nodes). Then, the probability of hacking is given by:

$$P = \prod_{\mathcal{R}_j \in \mathcal{R}} \left(1 - \prod_{i \in \mathcal{R}_j} (1 - p_i)\right). \tag{3}$$

Here, each node is identified with its label i.

This task poses an algorithmic challenge, and to the best of our knowledge, there is no standard effective method to solve this in this form. Instead, we could search for a strategy that guarantees a certain threshold of security level and is flexible within this limit. As a benefit, this may also allow us to adjust the algorithm to network traffic. In the next paragraph, we present an approach to the problem with respect to the considerations mentioned above.

3.1.2. Simplified problem

Here, we assume that each node has the same probability p, which is small enough so that we can use the approximation $1 - (1-p)^n \approx np$. The approximation is satisfied under the assumption that $np \ll 1$ and is further analysed in Section 4.2.4. According to the model above, we can simply take $p := \max_i(p_i)$. If we consider $N := |\mathcal{R}|$ paths each containing $n_j := |\mathcal{R}_j|$ (intermediate) nodes, then the probability of hacking is equal:

$$P = (n_1 p)(n_2 p)\dots(n_N p) \le p^N \left(\frac{\sum_{1 \le j \le N} n_j}{N}\right)^N = \left(\overline{n}p\right)^N, \tag{4}$$

where $\overline{n}$ denotes the average path length (in the sense of the number of intermediate nodes), and we use a well-known inequality:

$$\sqrt[N]{a_1 a_2 \dots a_N} \le \frac{a_1 + a_2 + \cdots + a_N}{N}. \tag{5}$$

Note that the equality is satisfied for $a_1 = a_2 = \cdots = a_N$, which in the case under consideration corresponds to paths of equal length. In a real QKD network, we may expect that the lengths of the paths are similar, and in this case the given upper bound could be a good approximation.

3.1.3. Solution

Now we can slightly reformulate the problem so that we do not minimise P but $P' := (\overline{n}p)^N$ instead. This depends on two factors: the number of paths and the average length of these paths. In many analyses, equally length paths are considered, and therefore it is always optimal to use as many paths as possible, but it turns out that it may not be desired for an arbitrary network. The analysis of an educational example is provided in Appendix A. For a fixed N solution, which is given and can be obtained by the minimum-cost flow algorithm (with unit capacities and certain transformation of the graph), there also exist other simpler algorithms like Suurballe's algorithm (vertex disjoint path version) (see [18]). We may assume that in the realistic case N will not exceed 10. Therefore, an efficient algorithm could be obtained by checking each possible number of paths separately.

One last remark is about the practical aspect of the obtained solution. If we accept loss in security level (but within threshold bound), we can try to add traffic management within the algorithm simply by manipulating weights of edges. Undesired routes will be less likely to be chosen. However, in this article we do not develop this concept further; it is left for future work.

3.1.4. Multiple communicating parties

The problem arises when we have more than two communicating parties, and we do not allow a path to share a link (for quantum networks, the effectiveness of links is the main restriction). This is an extensively studied problem called k-EDP (k-edge-disjoint path problem) [19].

Definition 1. Consider the following well-known problem, which is called the k-disjoint paths problem (k-DPP). For a given graph G and a set of k pairs of terminals in G, the objective is to find k vertex-disjoint paths connecting given pairs of terminals or to conclude that such paths do not exist.

It is proven that this problem is NP-complete [20]. Therefore, finding many paths for each of the pairs k is "at least of class NP", as this problem can be reduced to k-DPP by adding an appropriate number of paths between distinguished pairs of terminals. However, if k is fixed, there are polynomial solutions for k-DPP and even for the shortest k -DPP [21,22], so we can hope to find a solution while dividing the network into clusters.

3.2. Correlated attacks with finite resources

3.2.1. General formulation

As in the previous section, let us first formulate a general problem:

Consider a graph with a distinguished pair of nodes (A, B). We assume that there exists a system of (disjoint) paths connecting A, B, and the adversary, knowing this system and having some resources that can be used to take control of nodes and extract keys. The following assumptions are made:

- *The opponent has complete knowledge of the arrangement of the system.*
- *Hacking one node on the path makes this path untrusted.*
- *Communication is hacked if each path is untrusted.*
- *The probability p_i that the ith node becomes untrusted depends on amount of allotted resources $r_i \geq 0$ and is given with proper continuous function $p_i = f_i(r_i)$ specified for this node. Because $f_i(r_i)$ has an interpretation of probability, it takes values from the range $[0, 1]$.*
- *The resources are bounded, that is, $\sum_i r_i \leq R$.*
- *For each node $f_i(0) = 0$.*

We seek tactics (system of paths) that minimise the probability of hacking communication.

As before, solving the problem in this form poses a challenge, and even for a fixed system, calculating the minimal probability of hacking (corresponding to the optimal redistribution of resources) is difficult due to the continuous character of variables and unknown functions. Therefore, again, we need simplification.

3.2.2. Simplified problem

We first make the following observation:

Lemma 1. *Without loss of generality, we can assume that each function f_i is not decreasing, and the condition $\sum_i r_i = R$ is used.*

Proof. We do not need to use all resources, so if it is optimal to use x_2 resources for a certain node and there exists a x_1, such that $x_1 < x_2$ and $f(x_1) > f(x_2)$, then it is optimal for the adversary to use x_1. The adversary will obtain the same result using the alternative function $f(x) = \max_{y \leq x}(f(y))$. The second part of the lemma is straightforward. □

In real communication networks, we can assume that each node does not differ much in construction, and thus their characterisation will have much in common. At the same time, one shall not allow the adversary to easily take control of the node (other than the mean $f(R) \ll 1$), so the arguments of the function could be considered small, which allows us to expand the function f in series and consider its linear approximation. Alternatively, for a given function $f(x)$, we can construct a new function $g(x)$ that is linear up to a certain point, then constant (equal 1) and satisfying $f(x) \leq g(x)$.

We summarise this discussion with the following additional assumptions for the problem.

- The function f is the same for all nodes, *i.e.* $\forall_i f_i = f$.
- The function f is not decreasing, and the opponent always uses all available resources.
- The function f is expressed in the following form:

$$f(x) = \begin{cases} \alpha x, & x \leq \frac{1}{\alpha} \\ 1, & x \geq \frac{1}{\alpha} \end{cases}.$$

With this simplification, it turns out, that a sensible analysis can be made. We first develop an optimal adversary strategy for a single path of length n. To solve this problem, we use Lagrange multiplayer method. The function we want to maximise (probability of hacking) is:

$$P_{\text{sp}}(r_1, r_2 ..., r_n) = 1 - (1 - f(r_1))(1 - f(r_2))...(1 - f(r_n)), \tag{6}$$

with constrain $G(r_1, \dots, r_n) = \sum_i r_i - R = 0$.

As a result, we see that among the candidates for the global extremum (points $(r_1, r_2, \dots, r_n)$) some of $r_i = 0$ and the rest are equal to each other. Therefore, the set of extreme values is $\{P_k | k \in 1, \dots, n\}$, where $P_k = 1 - \left(1 - \alpha \frac{R}{k}\right)^k$ and since this represents the Euler sequence that is decreasing, we obtain the global maximum for $k = 1$, which corresponds to placing all available resources on a single node.

We summarise it in the following theorem:

Theorem 1. *Given a single path, the optimal strategy for an adversary is to attack a single node, and the probability of hacking is αR.*

From this we obtain an important conclusion about the situation with many paths.

Corollary 1. *Given N disjoint paths, the optimal strategy for an adversary is to attack only one node on each path.*

Proof. Let us assume on the contrary that in optimal strategy for the opponent, there exists a path on which two or more nodes are attacked. If we relocate resources from these nodes to a single node on this path, we will have a higher probability of hacking this path and, therefore, obtain a better strategy since the probability of hacking the system is the product of probabilities for individual paths. □

If we have N paths and r_j are resources used to hack the jth path (at a single node), then the probability of hacking the protocol is:

$$P(r_1, r_2, \dots, r_N) = (\alpha r_1)(\alpha r_2)...(\alpha r_N) \leq \left(\frac{\alpha R}{N}\right)^N. \tag{7}$$

Here, we assumed that $R \ll \frac{1}{\alpha}$, and we used Eq. (5).

3.2.3. Solution

Choosing the hacking strategy, we can focus on minimising the term $\left(\frac{\alpha R}{N}\right)^N$. There is only one variable to control: the number of disjoint paths. As αR in this approximation is always less than 1 (otherwise, the approximation we used fails), we are interested in increasing N. The maximal number of disjoint paths between pairs A and B is equal to the minimal size of the vertex cut of that pair, thanks to Menger's theorem.

Through the vertex cut, there is a notion that has no unique definition in the literature, and we restate it here:

Definition 2. A-B vertex cut is a set of vertex that does not contain $A - B$ so that after the removal of this set from the graph, there is no path between A and B. Later in the article, we will refer to it as cut, while the default vertex A and B will be sender and receiver (Alice and Bob). We say that the vertex cut is **minimal** if there is no cut with a smaller order.

We summarise our conclusions with the following theorem:

Theorem 2. *To improve security against correlated attacks for users A and B, the desired strategy is to increase the order of the minimal vertex cut $A - B$, that is, the number of disjoint paths.*

Finding the order of the minimal vertex cut is a problem equivalent to (after a simple transformation of the graph) solving *the max-flow problem*.

4. Multiple overlapping paths scheme

We have performed an analysis of the security of multiple-path scheme models under the assumption that all paths are disjoint. We now present a Multiple Overlapping Paths scheme (MOPs), where additionally to a system of disjoint of paths, the interpath links exist. Such an extension can always be made without loss of security, with only a slight modification of the well-known *hop-by-hop* protocol. Unlike

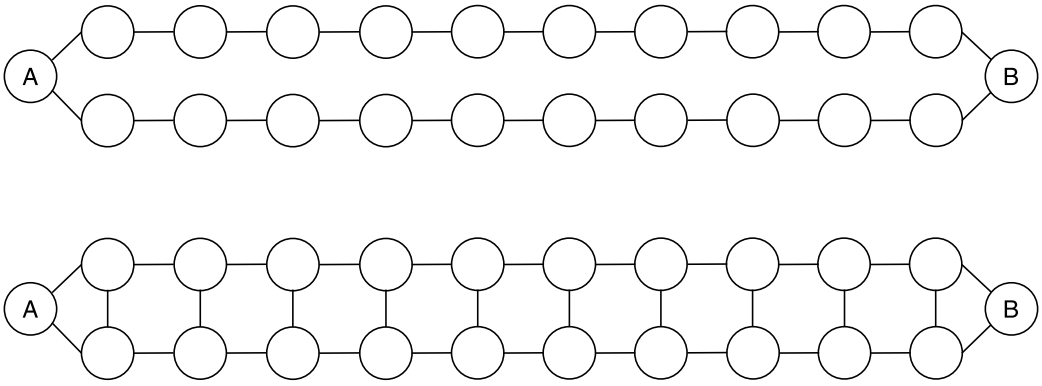

Fig. 2. MNOP network (above) with 2 disjoint paths and with additional interlinks (below).

MNOPs, where each intermediate node has exactly two links, now it can have more. A similar problem was previously analysed in Ref. [14]. However, to improve security, the total number of links is increased.

In this article, we follow an alternative idea, increasing security with the use of interlinks, but without changing the number of links. Roughly, we can say that we completely remove one path and use its resources (links) to make interconnections between the rest.

Definition 3. In the MOPs communication scheme, each node in the network, except Alice and Bob, sends the XOR result of all the keys from the neighbour connections to Bob via an unencrypted (but authenticated) channel (available to Eve). The shared key will be XOR of all these messages received for Bob and for Alice XOR of all subkeys Alice shares with the intermediate nodes.

This idea was originally published in [13], where a detailed analysis is included. Here, we recall only the most important conclusions. For the case without interlinks between paths, this will work as in the classic scheme, with the difference that the message is transmitted to Bob instead of to the next node on the path, but in Appendix B we show that MOPs can be modified so that it mimics the hop-by-hop method, which is important for the practical use and efficiency of the network.

4.1. Security against correlated attacks

Property 1. *In MOPs connection is secure only if there exists a path without any node controlled by adversary. Therefore, to hack a communication between A and B, the adversary must control the cut of the $A-B$ vertex.*

There is no reward in using MOPs against correlated attacks. It can be shown that adding interlinks to a system of disjoint paths does not increase the order of minimal $A-B$ vertex cut (for example, because the number of nodes connected to Alice does not change, and these nodes form a vertex cut). However, it can change the number of such cuts and, therefore, turns out to be useful against an uncorrelated attack. In general, to calculate the probability that the protocol is compromised, one must know the trustworthiness of all nodes and calculate the probability that at least one of the $A-B$ cuts (not necessarily minimal) becomes untrusted. This can be a really challenging problem in general. We managed to perform an analytical analysis on a certain special type of 'rectangular-grid-graph" that could model the real QKD network. The model is discussed in the following paragraphs.

4.2. Security against uncorrelated attack

4.2.1. Intuitive approach

We begin with an intuitive assertion that adding interlinks instead of a new path could perform better than MNOPs. If the probability of compromising the security of a node (and therefore leaking the key) is sufficiently small, we can take into account only the smallest (minimal) $A-B$ vertex cut. The number of such combinations of nodes in the classical scheme is

$$\prod_{1\le j\le N} n_j \approx (\overline{n})^N, \tag{8}$$

where n_j is the number of intermediate nodes in the jth path and N is the number of paths. We will improve security by reducing the number of $A-B$ vertex cuts in the graph by adding interlinks between paths. An example is shown in Fig. 2.

Let us assume that the number of intermediate nodes between A and B is n (for every path) and we have initially 2 disjoint paths. The number of 2-cuts (cuts of size 2) is $\approx 3n$, for the MNOP scheme, it would be n^2. If p is sufficiently small, we can state that only minimal cuts influence the probability of hacking. Then, the probability of compromising the security of the MNOP scheme is $(np)^2$, while for MOPs:

$$\underbrace{\binom{2n}{2}p^2(1-p)^{2n-2}}_{\text{probability of hacking pairs}} \times \underbrace{\frac{3n}{\binom{2n}{2}}}_{\text{allowed pairs}} \approx 3np^2. \tag{9}$$

The formula stems from a binomial distribution (we may control 2 nodes) and the probability that these two nodes form the desired hacking cut. This can be a significant advantage when n is large. However, we use additional links that, in turn, could be used to make another path. We now pose the question: "Can adding interlinks, instead of a new path, be a desired strategy?". It appears that in some cases, especially for large n, this can be true. Comparing the graph with interlinks and the MNOP scheme with one additional path, we obtain probability, respectively $(np)^3$ and $(3np^2)$. The ratio of probabilities is,

$$\eta := \frac{P_{\text{MOP}}}{P_{\text{MNOP}}} \approx \frac{3np^2}{(np)^3} = \frac{3}{pn^2}. \tag{10}$$

If $\eta < 1$, it is optimal to use the proposed strategy. But this poses a condition on p, that is,

$$p > \frac{3}{n^2}. \tag{11}$$

At the same time, we have assumed that p is "sufficiently" small (because we neglected the influence of non-minimal cuts). In fact, it must

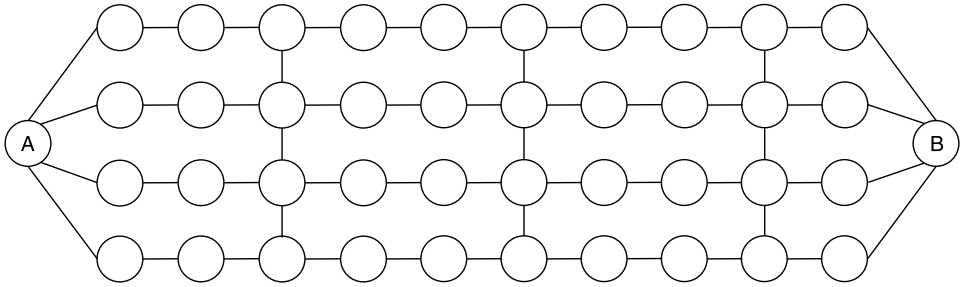

Fig. 3. A network with interlinks placement in MOP 4-scheme. The configuration is an alternative to 5 disjoint paths (MNOP) scheme.

at least meet the condition discussed in Section 2, *i.e.* $p \ll \frac{1}{n}$. Now, if n is sufficiently large, these conditions do not lead to contradiction. In the next chapters, we formalise and generalise the above considerations.

4.2.2. Adding a single link

After the discussion in the previous section, we could easily notice an important property, namely that adding just one link can reduce the probability of hacking by a factor of $\approx 1/2$. This approximation becomes more accurate as n increases. Consider an MNOP network with two disjoint paths as in Fig. 2, if we add a single intermediate link somewhere in the middle of the network and perform similar considerations as in Section 4.2.1, we obtain the result that the number of 2-cuts is $\approx 2\left(\frac{n}{2}\right)^2 = \frac{n^2}{2}$. Consequently, for a network with l paths, we can add $l-1$ links and reduce the probability of hacking by a factor $\left(\frac{1}{2}\right)^{l-1}$. This can be an important property, as by using only a few links, quite a good profit is obtained. Unfortunately, this effect does not stack: just from the example for two paths presented in Fig. 2 we see that adding n interlinks gives as probability reduction by factor $\frac{3}{n}$ not $\left(\frac{1}{2}\right)^n$.

4.2.3. Formal consideration

Now, we want to compare two situations, the MNOP scheme with the $l+1$ disjoint path and the strategy presented in Section 4.2.1 (MOPs) where on the behalf of interlinks we remove one path. For the new strategy, Alice and Bob are connected through l disjoint paths, each containing n intermediate nodes and each vertex having the same trust level $(1-p)$. We introduce the vertices numbering system: A, B for two communicating vertexes, g_{ij}, $i \in \{1,\dots,l\}$, $j \in \{1,\dots,n\}$ for intermediate nodes where i and j denote the row and column numbers, respectively. Adding interlinks can be achieved using different strategies. Here, we develop a strategy called *MOPs l-scheme*, which is probably not optimal but provides the possibility of analytical analysis. In the MOPs l-scheme, we connect all vertically adjoined nodes in every $(l-1)$-th column (connecting takes $l-1$ edges).

Formally, a new graph is obtained by adding a set of edges:

$$\mathcal{E} = \{g_{ij} \leftrightarrow g_{(i+1)j} | j \equiv 0 \pmod{l-1} \ \wedge \ i < l\}. \tag{12}$$

It can be easily seen that the number of created interlinks is at most n. The example is presented in Fig. 3.

In the MNOP scheme, the probability of hacking is $(1-(1-p)^n)^l \approx (np)^l$. We will keep this approximation, but we will do so later in Section 4.2.4, we perform an analytical analysis and estimate its error. To calculate the probability of hacking the l-scheme, we divide the sample probability space into events H_k, $0 \le k \le nl$, where each event means that exactly k of nl nodes become corrupted. This is a complete and pairwise disjoint set of events. Therefore, according to the law of total probability and binomial distribution, the probability of information leakage in the MOP l-scheme is:

$$P(H) = \sum_{k=0}^{nl} \binom{nl}{k} p^k (1-p)^{nl-k} P(H|H_k), \tag{13}$$

where H is an event related to hacking a $A-B$ vertex cut. Of course, $P(H|H_k) = 0$ if $k < l$. Therefore, we must calculate $P(H|H_k)$ for $k \ge l$. This is a rather difficult task. Instead, we perform an estimate that provides a useful upper bound on $P(H)$. Let us denote the number of minimal cuts for the l-scheme of length l by

$$c(l,n) := P(H|H_l)\binom{nl}{l}. \tag{14}$$

Obviously, $c(l,n) = n$ if $l = 1$, and for $l \ge 2$ we have the following theorem:

Theorem 3. *Define:*

$$\alpha(l) = \frac{2^{-l}\left(\left(1+l+\sqrt{l^2+6l-7}\right)^l - \left(1+l-\sqrt{l^2+6l-7}\right)^l\right)}{\sqrt{l^2+6l-7}}, \tag{15}$$

for $l \ge 2$ *and* $\alpha(1) = 1$. *Then*

$$\alpha(l) \times (n-2l^2) \le c(l,n) \le \alpha(l) \times n. \tag{16}$$

The term $\alpha(l)$ gives the asymptotic average number of minimal cuts per column as $n \to \infty$. For real schemes, it serves as an upper bound.

Proof. The vertex cut requires hacking at least one vertex in each row, and each vertex belongs to at least one cut (each column forms a cut), and if we settle on one vertex, we can easily find all cuts containing this vertex. Let g_{ij} be the settled vertex, and

$$1 + l < j < n - l, \tag{17}$$

we can formulate a procedure that determines all vertices of the next (or previous) row that can be used to form the cut:

- If degree $g_{ij} > 2$, then the possible vertices are $g_{(i+1)(j+r)}$, $r \in \{-l+1, -l+2, \dots, l-2, l-1\}$.
- If degree $g_{ij} = 2$, then the possible vertices are $g_{(i+1)m}$, $m \in \{a \dots b\}$, where a and b are determined by the conditions: $deg(g_{ia}) > 2$, $deg(g_{ib}) > 2$, $b - a = l - 1$, $a < j < b$.

So, we can say that each vertex "produces" certain new vertices in each row. For vertices that do not meet (17), which can be intuitively described as "boundary vertices", the above procedure must be slightly modified but will result in a smaller number of "produced vertices". Then, we calculate $\alpha(l)$ with the following recursion:

```
f(n,dg2): %n-number of iteration,
dg2-boolean variable true if degree >2
  if n==1 return 1
  if dg2==True return 3*f(n+1,1)+2(l-2)*f(n+1,0)
  if dg2==False return 2*f(n+1,1)+(l-2)*f(n+1,0)
alpha(l):
  return (f(l,1)+(l-2)*f(l,0))/(l-1)
```

This can be alternatively rewritten as a pair of related sequences and expressed in the following form with matrix multiplication:

$$\alpha(l) = \frac{1}{l-1}\begin{pmatrix}1 & l-2\end{pmatrix}\begin{pmatrix}3 & 2(l-2)\\ 2 & (l-2)\end{pmatrix}^{l-1}\begin{pmatrix}1\\1\end{pmatrix}. \tag{18}$$

The matrix in this formula can be diagonalised, leading to Eq. (15). The expression in $\alpha(l)$ is overestimated because we neglected the "boundary corrections". Because the ranges of r and m are sometimes smaller (especially close to the sides of the graph), the estimation improves as n increases. An underestimation is obtained if we do not count all vertex cuts that contain "boundary vertices". □

Now we have analytic estimation for the first non-zero term in Eq. (13), where we neglect the "boundary corrections". In fact, it can also be calculated numerically in time $O(nl)$ using dynamic programming. The algorithm is presented in Appendix C. Therefore, in the numerical calculation (for hypothetical testing of the algorithm), we will use the exact value of $\alpha(n,l)$, although for analysis it is convenient to use the upper bound as in Theorem 3. The difference is significant when n has a similar order as l.

Calculating the higher terms in Eq. (13) poses a problem. However, it turns out that if $nlp < 1$ it can be approximated by geometric series. To do that, we first derive an important observation:

Lemma 2. *Consider,*

$$\beta_{n,l}(k) := \binom{nl}{k} P(H|H_k), \tag{19}$$

which is the number of cuts of size k in the MOPs l-scheme graph. The following inequality holds.

$$\beta_{n,l}(k) \le \beta_{n,l}(k-1)nl \text{ for } k-1 \ge l. \tag{20}$$

Therefore:

$$\beta_{n,l}(k) \le \alpha(l)n(nl)^{k-l}. \tag{21}$$

Lemma 2 is obtained by numerical analysis, not analytical analysis. We do not have a proper proof of its validity, but for practical use, we only need to confirm its correctness with the given n and l. This could be done numerically with brute force, generating all possible vertex sets and checking if it is cut. In fact, due to the computational complexity of this method, it can only be performed on small graphs. We perform such calculations with all possible schemes with "graph size" < 30 *i.e.* if $nl < 30$. See more in Appendix D. An alternative method that could support these results is its statistical testing, *i.e.* that generates a sample consisting of random nodes and calculates the cut ratio $A-B$, generated this way versus the sample size. An important observation worth stressing is that the problem of calculating the number of cuts in such a graph (and calculating the probability that there exists a path between A and B) looks similar to the percolation problem. We did not follow this lead, but it could be useful if we tried to prove Lemma 2.

Using Lemma 2, we can now obtain the following.

Theorem 4. *If p is small enough, so it satisfies $nlp \le r < 1$, then:*

$$P(H) \le \frac{1}{1-r}\alpha(l)np^l. \tag{22}$$

Proof. First, from Eq. (21), we derive

$$P(H|H_k) = \frac{\beta_n l(k)}{\binom{nl}{k}} \le \frac{\alpha(l)n(nl)^{(k-l)}}{\binom{nl}{k}}. \tag{23}$$

Then, Eq. (13) can be estimated using a geometric progression formula with the initial term $\alpha(l)np^l$ and the common ratio nlp. □

It will be useful to know how much our algorithm is better than the MNOPs. Thus, for a given network, we introduce the efficiency coefficient η, which approximates the ratio of probability of hacking of our algorithm versus the MNOP scheme:

$$\frac{P(H)}{(np)^{l+1}} \le \frac{\frac{1}{1-r}\alpha(l)}{pn^l} := \eta. \tag{24}$$

As described above in Section 2, we used the approximation $1-(1-p)^n \approx np$ for the classic scheme, which is not necessarily desirable, as is generally $1-(1-p)^n < np$. We refer to this concern in Section 4.2.4, in which we show that due to this fact we shall introduce a numerical correction for η. However, for "reasonable" graphs, the corrected efficiency is in the worst cases only about 1.7 times greater than the one predicted by Eq. (24). Therefore, it has a really small impact, and we omit it. The sufficient condition for the l-scheme to perform better than the classic scheme is:

$$\eta < 1. \tag{25}$$

Consequently, the risk of hacking is at least η times smaller (if both schemes use the same number of links). Eq. (25) allows us to study the performance of the algorithm, and we see that for the constant p we can obtain, asymptotically for very large graphs, very low values of η. However, as the trust of the nodes increases (so p becomes smaller), the effectiveness of the algorithm decreases. At the same time, we need to maintain the assumed condition $nlp < r < 1$. This may raise concerns about the upper bounds on p. However, for "realistic networks" we can assume that $n < 100, l < 10, r < 1/2$ ($\frac{1}{1-r} < 2$) which, in the worst case, gives $p < 1/2000$, which is not very restrictive. In Section 4.3 we present a numerical analysis of Eq. (25), which shows exactly in which ranges of n and p our algorithm is useful.

4.2.4. Analysis of approximation in MNOPs scheme

In Sections 3 and 4 the approximation $1-(1-p)^n \approx np$ has been used, which works for $np \ll 1$. Here, we analyse the validity of this assumption and make a correction for Eq. (25). First, let us note that just for the classic scheme we shall demand that $np < 1$. If this does not hold (so $p > \frac{1}{n}$), then the probability of hacking a single path is:

$$P = 1-(1-p)^n > 1-\left(1-\frac{1}{n}\right)^n > 1-\frac{1}{e} \approx 0.63, \tag{26}$$

which is unacceptable for any real network called "secure". This can also substantiate the reality of the assumption made in Theorem 4 that $nlp < 1$. We define the parameter μ

$$\mu := \frac{1}{np} \ge 1, \tag{27}$$

which is a certain constant fixed for a network (path), this divides all networks models (paths) into certain classes, with a fixed value of μ. Analysing the validity of the approximation $1-(1-p)^n \approx np$ as a function of μ gives us useful results, namely the lower bound for the probability of hacking. For a given class (characterised by μ), consider the following.

$$\upsilon_n(\mu) = \frac{1-(1-p)^n-np}{np} = \frac{1-(1-\frac{1}{\mu n})^n-\frac{1}{\mu}}{\frac{1}{\mu}} = \mu\left(1-\left(1-\frac{1}{\mu n}\right)^n\right)-1. \tag{28}$$

Clearly,

$$P = 1-(1-p)^n = (1+\upsilon_n(\mu))np.$$

We have the following two properties:

$$\upsilon_n(\mu) > \upsilon_{n+1}(\mu), \tag{29}$$

and

$$\upsilon(\mu) := \lim_{n\to\infty}\upsilon_n = -1+\mu(1-e^{-1/\mu}). \tag{30}$$

Let $\gamma(\mu) := 1+\upsilon(\mu) = \mu(1-e^{-1/\mu}) < 1$. Then,

$$1-(1-p)^n \ge \gamma(\mu)np. \tag{31}$$

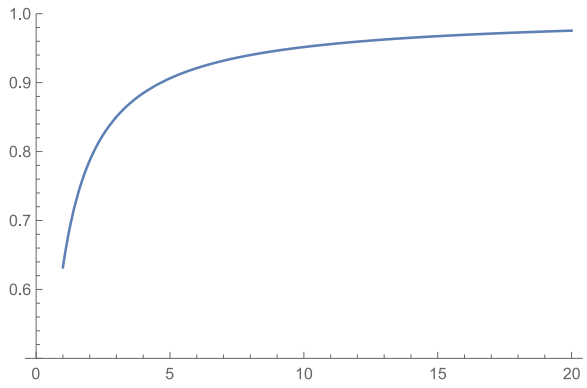

Fig. 4. Plot of the function $\gamma(\mu)$.

Therefore, putting $\gamma(\mu)np$, instead of np, in Eq. (25), we get a more restrictive condition on η (which refers to the situation where the classic scheme works better than assumed):

$$\frac{P(H)}{(1-(1-p)^n)^{l+1}} \leq \frac{P(H)}{(np)^{l+1}} \frac{1}{\gamma(\mu)^{l+1}} \leq \eta \frac{1}{\gamma(\mu)^{l+1}} < 1. \tag{32}$$

The function $\gamma(\mu)$ is presented in Fig. 4. We conclude that typically we shall deal with systems with parameter $\mu \geq 10$, this is done by posing arbitrarily but reasonable condition ($P < 1/10$) and analysis as in Eq. (26). We assume that, in general, $l \leq 10$, otherwise communication will be very expensive in resources. For such an assumption, we can estimate the maximal value of correction, that is, $\frac{1}{\gamma(10)^{11}} \approx 1.72$. This section does not change our conclusions significantly, but is required for the completeness of the analysis.

4.3. Numerical analysis

For a given graph and parties communicating with the l-scheme, parameters like n and l can be settled or easily estimated, but trust (equivalently probability of hacking) of intermediate nodes is very unclear to define and measure in reality. It can be connected with various events, such as a random failure of the network, corrupted labour, hackers, or it could change over time. Therefore, it is rather impossible to declare its value at the stage of theoretical considerations. But still we expect that this trust (and, respectively, probability of hacking p) must be in reasonable range, for example $p \approx 0.1$ or $p \approx 10^{-27}$ are certainly not, of course, for security purposes p should be at least as possible. Another problem is to find this range, but we can analyse the conditions imposed on p steaming from the algorithm structure of the l-scheme and our considerations, namely $nlp < r < 1$ and $\eta < 1$, depending on its parameters n, l, r. Those two conditions determine the possible range of p (first serves to establish the upper bound and second for the lower bound) in which the MOPs l-scheme will work. The maximal and minimal values determined for p dependent on l and p with fixed parameter $r = 1/10$ are presented in Figs. 5–7.

Taking plans regarding the future quantum network spanned *e.g.* across Europe, we suppose that its size will refer to the average number of intermediate nodes on one path $n < 40$ (since the expected size of the QKD link is about 100 km) and the maximum number of disjoint paths $l < 6$. Therefore, the applicability of the MOP l-scheme is under question and depends on the real value of p.

5. Conclusions and future works

In this article, we have analysed two scenarios of attack that can be performed on trusted nodes in the hybrid QKD network. First, describe the situation in which each trusted node could be compromised with a certain probability, p, and then describe a correlated attack on a network with finite resources. For which case, the risk of hacking is greater depends on individual parameters of the network and attacking party, which are difficult to predict in reference to the real world. However, we can infer that with growing network size, the second scenario is less vulnerable to attacks.

Next, we have described the scheme of communication in the QKD network extending the multiple-path scheme by the possibility of crossing communicating paths, the MOPs scheme. This scheme uses the same amount of resources (QKD links) and can perform better under certain conditions, as analysed in Section 4.3. The graphic visualisation of the most restrictive constraint (*i.e.* on the minimal value of p) is presented in Fig. 8. Unfortunately, these constraints are relevant for distant users in large-scale networks only.

The concept presented in Section 4.2.2 is worth highlighting. It does not present a groundbreaking idea but introduces an interesting and inexpensive improvement to the QKD network. This work develops a new way of thinking about QKD multiple-path algorithms in hybrid networks with trusted nodes. The proposed algorithm is not optimal. Considering different topologies of interlinks could perform better, but the simple model under investigation enabled analytical analysis. For this moment, it is difficult to judge whether the presented concept will be useful in practice, yet it opens a new path for the future development of QKD networks.

In the literature, different approaches to improving the efficiency and security of quantum key distribution through the use of multiple paths are explored. Here, we base on the findings of earlier studies and offer new insights and solutions to challenges in this field. Specifically, we extended the QKD multiple-path distribution protocol, assuming that the paths can overlap. Our analysis shows that to improve security, it is (under certain conditions) more optimal to add interlinks between disjoint paths instead of adding a new path. A similar concept using overlapping paths has been presented in [14], where security is improved by introducing complete subgraphs (“cities”). However, in that article, no indication of an advantage over the MNOPs was made. Furthermore, while preparing this article, another work appeared that addressed the security issue of overlapping multiple paths [15]. In this article, the potential vulnerability in quantum networks based on trusted nodes with multiple paths is considered, to which our results are complementary. Specifically, in that article, a different architecture of a hybrid network was considered, which enabled to obtain certain analytical results, which led to conclusions on security being in qualitative agreement with the results obtained here. However, the results of [15] do not indicate the possible advantage of MOPs over MNOPs, which has been shown here.

We constructed two models of possible attacks: uncorrelated attack (in which each node has a certain probability of becoming compromised), and correlated attack (in which the opponent owns certain resources that could be used to compromise the security of certain nodes). Furthermore, based on [13], the attacks have been extended to the QKD multiple-path distribution protocol, assuming that the paths can overlap. Our analysis shows that to improve security, it is (under certain conditions) more optimal to add interlinks between disjoint paths instead of adding a new path. We analysed existing multiple-overlapping path algorithms for key exchange and identified their weaknesses in terms of security and efficiency. Furthermore, we proposed a new algorithm that addresses these weaknesses by using overlapping paths. The proposed algorithm is shown to be more secure and efficient than existing algorithms, making it potentially useful in case of the large-scale quantum networks.

The research on Multiple Overlapping Paths algorithms for Secure Key Exchange in Large-Scale Quantum Networks holds potential future prospects in the realm of secure communication network development. As quantum technology progresses and quantum networks expand, the demand for secure key exchange intensifies. This work offers a

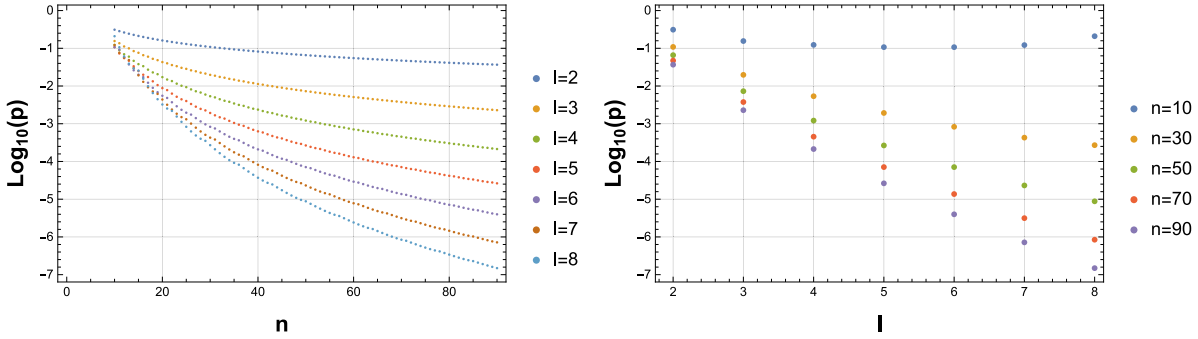

Fig. 5. Minimum value of p (expressed in logarithm of base 10) in dependence on l and $r = 1/10$, satisfying condition $\eta < 1$.

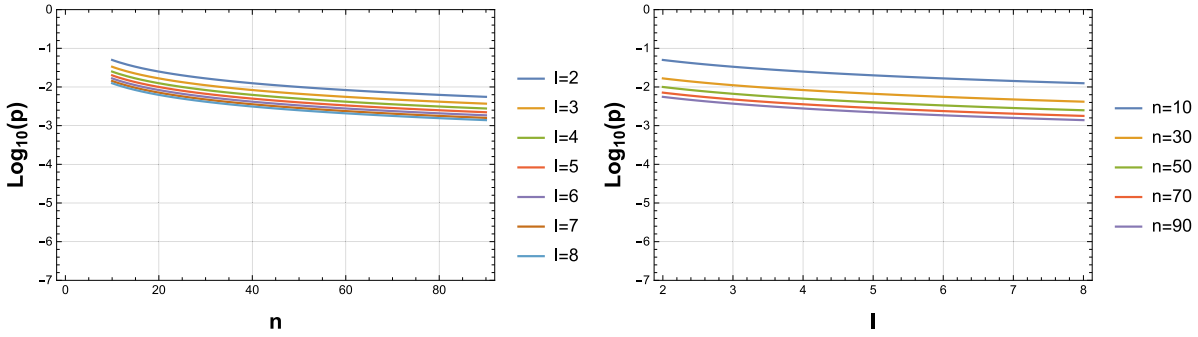

Fig. 6. Maximal value of p (expressed in logarithm of base 10) in dependence of l and n and, satisfying condition $nlp < r = 1/10$.

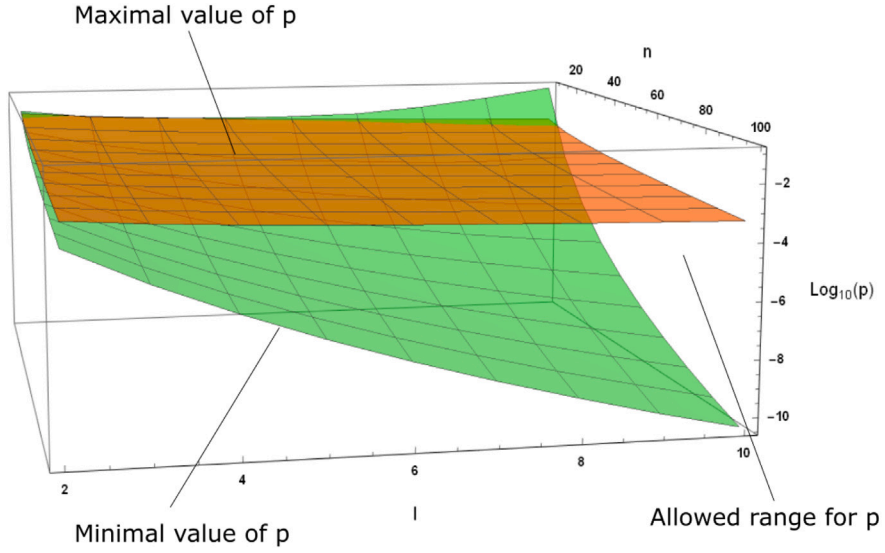

Fig. 7. Range of p which fulfil conditions $\eta < 1$ and $nlp < 1/10$, expressed with logarithm of p in dependence of different l and n.

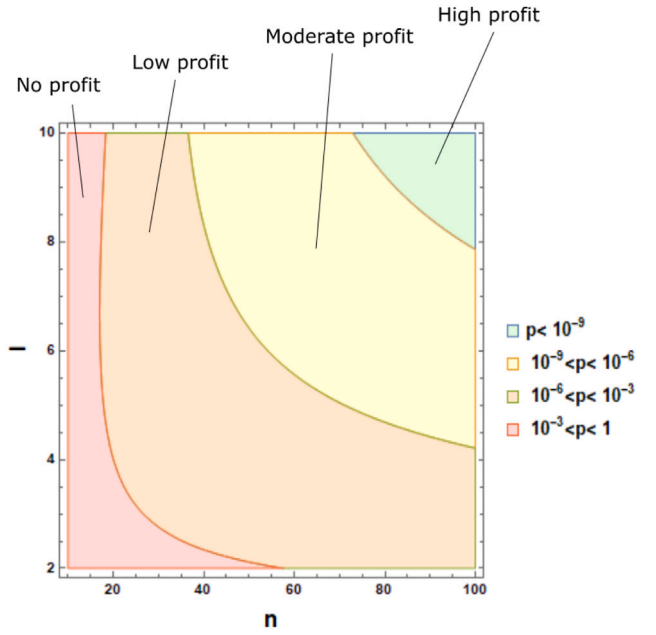

Fig. 8. Regions of the allowed value of p as a function of l (number of paths) and n (number of nodes in a path), under the condition $\gamma < 1$. The condition guarantees that the probability of hacking the MOP scheme is smaller than for the MNOP scheme. The colours represent the degree of usefulness of the algorithm, with green as the most useful and red as not useful. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

potential solution by employing multiple overlapping paths to enhance security and reliability in quantum communication networks, especially in large-scale situations where single-path schemes might fall short.

In quantum networks, multiple path strategies consume numerous quantum node resources [23], and our work could help address this issue. Other efficiency-improving algorithms, such as Quantum Dynamic Routing [24], can be combined with our approach for further research.

Regarding future prospects and potential applications, the proposed algorithm could be employed to bolster the security of quantum key distribution networks in various contexts. For instance, it could secure communication between government agencies or financial institutions that necessitate high-security levels. Moreover, the algorithm could enhance secure communication between individuals who require privacy, such as journalists or activists, even in situations where intermediate nodes are not entirely trustworthy.

To ensure the comprehensive evaluation of the proposed algorithm, it is highly recommended that future research concentrates on simulating its functionality across diverse scenarios. It is important to note that while the proposed algorithm surpasses existing alternatives in terms of security and efficiency, it is not optimal. The selected architecture for overlapping paths has been dictated by the simplicity of theoretical analysis, and better strategies may exist.

CRediT authorship contribution statement

Mateusz Stępniak: Conceptualization, Methodology, Investigation, Software, Formal analysis, Visualization, Writing – original draft. **Jakub Mielczarek:** Conceptualization, Investigation, Validation, Writing – original draft, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

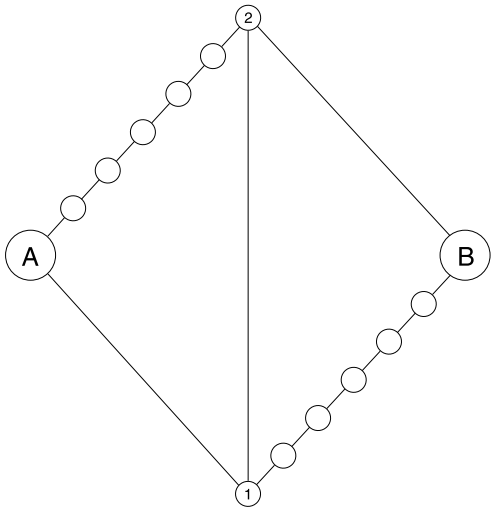

Fig. 9. Example of network for which single path protocol is more optimal than multi-path one.

Data availability

Data will be made available on request.

Acknowledgements

The research has been supported under "The Excellence Initiative - Research University" programme at the Jagiellonian University in Kraków.

Appendix A

Here, we present an example where for the uncorrelated attack (even with uniform probability p for each vertex), it is not always optimal to use as many paths as possible. We adopt the assumptions made in the discussion of this type of attack in Section 3.1. Consider the network presented in Fig. 9, and let the number of intermediate nodes on the path $A-2$ and $1-B$ (a path that goes from A to 2 and 1 to B but omits 1 and 2, respectively) be n, and the probability that any (intermediate) node becomes compromised is p. If we use one or two paths, then the probability of hacking is, respectively:

$$P_1 = 2p, \tag{33}$$

$$P_2 = ((n+1)p)^2 . \tag{34}$$

It is possible to satisfy the inequality $P_1 < P_2$, which leads to the condition:

$$\frac{2}{(n+1)^2} < p. \tag{35}$$

But for n large enough, we can find p small enough so Eq. (35) do not contradict the condition assumed before, that is, $np \ll 1$. Therefore, even for the simplified version of the correlated attack model, it is not always optimal to use as many paths as possible.

Appendix B

In the MNOP scheme, each intermediate node has a connection to two other nodes and passes a secret key from one node to another in a hop-by-hop fashion using a one-time pad. In the MOPs scheme,

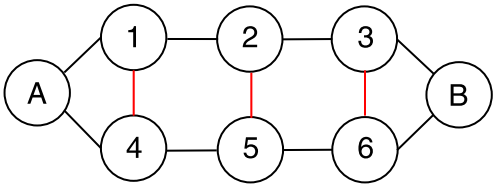

Fig. 10. Example of graph for classic and NDMS scheme, red links represent additional interlinks in NDMS scheme. In NDMS, the following messages will be created: $M_{12} = K_{A1} \bigoplus K_{12} \bigoplus K_{14}$, $M_{23} = M_{12} \bigoplus K_{23} \bigoplus K_{25} \bigoplus K_{12}$, $M_{3B} = M_{23} \bigoplus K_{3B} \bigoplus K_{36} \bigoplus K_{23}$, $M_{45} = K_{A4} \bigoplus K_{45} \bigoplus K_{14}$, $M_{56} = M_{45} \bigoplus K_{56} \bigoplus K_{25} \bigoplus K_{45}$, $M_{6B} = M_{56} \bigoplus K_{6B} \bigoplus K_{36} \bigoplus K_{56}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

each node sends a classical message to Bob, which may be not efficient (since it generates a lot of traffic in the classic network), and we now show a certain alternative that reduces impact on key rate if only we can decompose network (vertex used in actual communication) on system of disjoint paths that cover all vertexes. If we cannot meet this condition, we can mark some vertex that will still send a message directly to Bob and will not be counted as part of the network and then try to make decomposition.

Lemma 3. *Alternatively to the hop-by-hop scheme, we can use the following procedure: On path $A - 1 - 2...n - B$ vertex number 1 take the XOR of key's K_{A1} and K_{12} and send it to the next node. If the node $i \in 2...n$ receives the message M_{i-1} from the previous node, it takes XOR of his shared keys and the message and create new message, i.e. $M_i = M_{i-1} \bigoplus K_{i-1,i} \bigoplus K_{i,i+1}$. The difference is that instead of decoding and encoding operations, we do it in a single step.*

Lemma 4. *Alternatively, to the MOPs method, we can use the following procedure. Having a graph, we find a system of disjoint paths that cover all vertices. This system will determine the next node for each vertex. When communicating, each node uses the procedure described in Lemma 3 with a difference, we additionally XOR all interlink keys. Consequently, the message sent is: $M_i = M_{i-1} \prod_{(j,i)} \bigoplus K_{i,j}$, where (j,i) indicates the adjointness of the nodes j and i.*

One can check that this extension of the MOPs scheme omits unwanted procedure in which each node sends message to Bob and at the same time keeps benefits of utilisation of interlinks. A useful example is presented in Fig. 10.

Appendix C

Here, we present a dynamical programming algorithm to calculate the exact value of $c(l,n)$, the greatest discrepancy in the result appears when n is close to l:

```
algorithm alpha is
  input: n and l parameters of graph
  output: number of minimal cuts separating A and B

 create two dimensional array tab[l,n]
 for j in 1.. n :
   tab[1,j]=1

 for each row i in 2...n:
   for each cell in a given row tab[i,j], j in 1,n :
     calculate range of possible k (a,b) such that
     nodes(i,j) and (i-1,k)
     can be in vertex cut for k in a...b :
     tab[i,j]+=tab[i-1,k]

 return sum of tab[l,i]
```

Appendix D

Here, we present the numerical calculation performed to substantiate Lemma 2 of Section 4.2.3. For each graph, each subset of size k can be generated and checked to see whether it separates A and B (if A and B are in disjoint components). The numerical values of the function $\beta_{n,l}(k)$ for small graphs are presented in Fig. 11.

From this analysis, we can deduce that the ratio $\frac{\beta_{n,l}(k+1)}{\beta_{n,l}(k)}$ is maximal for $k = l$. We also derive numerical observations suggesting that the inequality (20) "cannot be strengthened", by which we mean that:

$$\frac{\frac{\beta_{nl}(l+1)}{\beta_{nl}(l)}}{nl} \to 1, \tag{36}$$

as n grows. In Fig. 12 the numerical results obtained for $l = 2$ are shown. For $l > 3$, the above thesis seems to hold, but collecting data for many points is too time-consuming, and therefore we do not present more results.

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 7 & 18 & 15 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 44 & 68 & 56 & 28 & 8 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 13 & 82 & 194 & 250 & 210 & 120 & 45 & 10 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 16 & 132 & 429 & 772 & 922 & 792 & 495 & 220 & 66 & 12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 19 & 194 & 809 & 1900 & 2979 & 3430 & 3003 & 2002 & 1001 & 364 & 91 & 14 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 22 & 268 & 1370 & 4008 & 7862 & 11412 & 12868 & 11440 & 8008 & 4368 & 1820 & 560 & 120 & 16 & 1 & 0 & 0 & 0 & 0 \\
0 & 25 & 354 & 2148 & 7566 & 17956 & 31626 & 43726 & 48618 & 43758 & 31824 & 18564 & 8568 & 3060 & 816 & 153 & 18 & 1 & 0 & 0 \\
0 & 28 & 452 & 3179 & 13140 & 36790 & 76568 & 125712 & 167924 & 184754 & 167960 & 125970 & 77520 & 38760 & 15504 & 4845 & 1140 & 190 & 20 & 1
\end{pmatrix}$$

$$\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 8 & 12 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 27 & 81 & 108 & 81 & 36 & 9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 38 & 221 & 561 & 814 & 764 & 492 & 220 & 66 & 12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 61 & 502 & 1858 & 4049 & 5923 & 6262 & 4971 & 3000 & 1365 & 455 & 105 & 15 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 72 & 814 & 4189 & 12912 & 26918 & 40872 & 47479 & 43466 & 31780 & 18561 & 8568 & 3060 & 816 & 153 & 18 & 1 & 0 & 0 & 0 \\
0 & 0 & 95 & 1339 & 8700 & 34221 & 92075 & 182619 & 280963 & 346925 & 350892 & 293543 & 203440 & 116277 & 54264 & 20349 & 5985 & 1330 & 210 & 21 & 1
\end{pmatrix}$$

$$\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 16 & 32 & 24 & 8 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 81 & 324 & 594 & 648 & 459 & 216 & 66 & 12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 256 & 1536 & 4480 & 8320 & 10896 & 10560 & 7744 & 4320 & 1816 & 560 & 120 & 16 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 313 & 3128 & 14724 & 43436 & 90044 & 139558 & 167926 & 160576 & 123638 & 77018 & 38694 & 15500 & 4845 & 1140 & 190 & 20 & 1
\end{pmatrix}$$

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 32 & 80 & 80 & 40 & 10 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 243 & 1215 & 2835 & 4050 & 3915 & 2673 & 1305 & 450 & 105 & 15 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1024 & 7680 & 28160 & 66560 & 112960 & 145376 & 146240 & 116880 & 74720 & 38160 & 15424 & 4840 & 1140 & 190 & 20 & 1
\end{pmatrix}$$

Fig. 11. Values of function $\beta_{n,l}(k)$. Column number (starting from 1) depict value of k, and row number value of n, and l is constant within a given matrix, respectively $l = 2, 3, 4, 5$. Through computation, time is exponential in the size of the system (which equals nl), calculations were made for graphs of a maximal size of about 30.

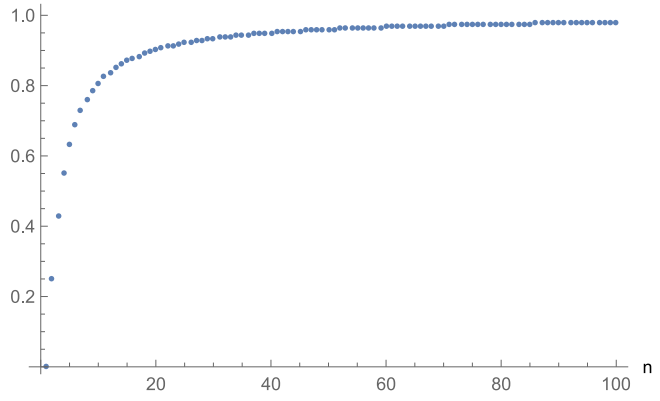

Fig. 12. Function $\frac{\beta_{nl}(l+1)}{\beta_{nl}(l)} \frac{1}{nl}$ with $l = 2$ in dependence of n.

References

[1] Diffie W, Hellman M. New directions in cryptography. IEEE Trans Inform Theory 1976;22(6):644–54.

[2] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. 1994, p. 124–34.

[3] Ekert A. Quantum cryptography based on Bell's theorem.

[4] Cacciapuoti Angela Sara, Caleffi Marcello, Tafuri Francesco, Cataliotti Francesco Saverio, Gherardini Stefano, Bianchi Giuseppe. Quantum internet: Networking challenges in distributed quantum computing. IEEE Netw 2020;34(1):137–43.

[5] Bedington R, Arrazola JM, Ling A. Progress in satellite quantum key distribution. Quantum Inf 2017;3(1):1–13.

[6] Sasaki M, et al. Field test of quantum key distribution in the Tokyo QKD network. Opt Express 2011;19(11):10387–409.

[7] Mehic M, Niemiec M, Rass S, et al. Quantum key distribution: a networking perspective. ACM Comput Surv 2020;53(5):1–41.

[8] Lopez DR, Martin V, Lopez V, et al. Demonstration of software defined network services utilizing quantum key distribution fully integrated with standard telecommunication network. Quantum Rep 2020;2(3):453–8.

[9] Stebila D, Mosca M, Lütkenhaus N. The case for quantum key distribution. In: Quantum communication and quantum networking. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 283–96.

[10] Bernstein DJ. Introduction to post-quantum cryptography. In: Post-quantum cryptography. Springerecuy; 2009, p. 1–14.

[11] Chen L, Chen L, Jordan S, et al. Report on post-quantum cryptography, vol. 12. US Department of Commerce, National Institute of Standards and Technology; 2016.

[12] Salvail L, Peev M, Diamanti E, Alléaume R. Security of trusted repeater quantum key distribution networks. J Comput Secur 2010;18:61–87.

[13] Zhou H, Lv K, Huang L, Ma X. Security assessment and key management in a quantum network. 2019, arXiv, abs/1907.08963.

[14] Beals TR, Sanders BC. Distributed relay protocol for probabilistic information-theoretic security in a randomly-compromised network. In: Safavi-Naini Reihaneh, editor. Information theoretic security. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008, p. 29–39.

[15] Gaidash A, Miroshnichenko G, Kozubov A. Quantum network security dependent on connection density between trusted nodes. 2022, arXiv, abs/2202.09162.

[16] Solomons Naomi R, Fletcher Alasdair I, Aktas Djeylan, Venkatachalam Natarajan, Wengerowsky Sören, Lončarić Martin, et al. Scalable authentication and optimal flooding in a quantum network. PRX Quantum 2022;3:020311.

[17] Chip E. Building the quantum network. New J Phys 2002;4:46.

[18] Suurballe JW. Disjoint paths in a network. Networks 1974;4(2):125–45.

[19] Eilam-Tzoreff T. The disjoint shortest paths problem, 85 (2). 1998, p. 113–38.

[20] Karp RM. On the computational complexity of combinatorial problems. Networks 1975;5(1):45–68.

[21] Lochet W. A polynomial time algorithm for the k-disjoint shortest paths problem. 2020.

[22] Kawarabayashi K, Kobayashi Y, Reed B. The disjoint paths problem in quadratic time. J Combin Theory Ser B 2012;102(2):424–35.

[23] Tsai Chia-Wei, Yang Chun-Wei, Lin Jason, Chang Yao-Chung, Chang Ruay-Shiung. Quantum key distribution networks: Challenges and future research issues in security. Appl Sci 2021;11(9).

[24] Amer Omar, Krawec Walter O, Manfredi Victoria U, Wang Bing. Dynamic routing for quantum key distribution networks. 2022.