Towards the World-Wide Quantum Network

Quoc-Cuong Le [1] and Patrick Bellot [1] and Akim Demaille [2]

[(1)] Institut TELECOM, Telecom ParisTech, Paris, France,
[(2)] EPITA Research and Developpement Laboratory (LRDE), Paris, France

Abstract. QKD networks are of much interest due to their capacity of providing extremely high security keys to network participants. Most QKD network studies so far focus on trusted models where all the network nodes are assumed to be perfectly secured. This restricts QKD networks to be small. In this paper, we first develop a novel model dedicated to large-scale QKD networks, some of whose nodes could be eavesdropped secretely. Then, we investigate the key transmission problem in the new model by an approach based on percolation theory and stochastic routing. Analyses show that under computable conditions large-scale QKD networks could protect secret keys with an extremely high probability. Simulations validate our results.

1 Introduction

The problem of transmitting a secret key from an origin to a destination over the network was considered for a long time. The current solution in most Internet applications is using Public Key Infrastructure (PKI). PKI relies on plausible but unproven assumptions about the computation power of eavesdroppers and the non-existence of effective algorithms for certain mathematical hard problems. As a result, PKI cannot meet the highest security level, also called *unconditional security*. Quantum Key Distribution (QKD) technology is a prominent alternative [1]. It was proven that QKD can provide unconditional security [2, 3, 4]. It is successfully implemented in realistic applications [5, 6, 7, 8]. However, QKD only supports point-to-point connections and intrinsically causes serious limits on throughput and range [5, 9]. A long-distance QKD transmission needs intermediate nodes to relay the key. In realistic scenarios, some of these nodes could be eavesdropped without the others knowing it. In consequence, the security of key will be compromised. Larger networks are more vulnerable.

This paper studies a partially compromised QKD network model that allows any member pair establishing securely a common key with almost certainty. The contributions are (i) a model of partially compromised QKD networks, (ii) the use of percolation theory techniques to find where almost-certainty can be achieved, (iii) stochastic routing proposals capable of achieving a given secrecy level.

The remainder is organized as follows. Section 2 introduces the QKD network's context and proposes a novel model of the world-wide QKD network. Section 3 presents related works. Section 4 seeks for the necessary condition to

achieve a given high secrecy of key transmissions. Section 5 presents our adaptive stochastic routing algorithms and analyzes their performances. We conclude in Section 6. The proofs of the theorems are given in Appendix.

2 A proposal for the world-wide quantum network

Preliminary QKD networks present two types of links: classical and QKD. Classical links are easy to implement, capable of providing high-speed but low-confidentiality communications. By contrast, QKD links aim at *unconditional security*. This causes undesirable limits of rate and range [5, 9]. The ultimate goal of QKD networks is unconditional security. QKD networks rather sustain QKD's restrictions to reach this goal. As such, there is no need to consider classical links in studying QKD network prototypes. In the following we will simply write links instead of QKD links.

The feasibly-implemented model of QKD networks so far is the trusted network model. Its representers are SECOQC and DARPA networks [10, 6, 11, 12]. This model assumes that all the network nodes are perfectly secured. This assumption is too strong in large-scale scenarios. Actually, eavesdroppers can ingeniously attack a proportion of nodes without leaving any trace in large-scale networks. Consequently, security may be compromised.

Restricted by a modest length of link, QKD networks don't present many choices of topology. Meshed topology would suit QKD networks [12]. Besides, distributed architecture is considered to be good. This paper follows these ideas. However we focus on the world-wide quantum network that is very different from small-scale quantum networks like DARPA and SECOQC. For simplicity, we choose the 4-connected grid topology. Nodes are represented by squares. Links have no representation because they have no effect on security analysis (see Fig. 1).

In QKD networks, intermediate nodes are vulnerable. Attacks are either detectable or undetectable. In principle, if an attack is detectable then we can find solutions to fix it. Undetectable attacks are very dangerous. We cannot detect them until great damage has been done. We take into account such attacks. Assume that each node sustains a probability p_e being eavesdropped without knowledge of the others. For simplicity, we focus only on cases where p_e is the same for all the nodes. Note that p_e should be small unless eavesdropper resources are much larger than those of legitimate users.

Modeling the world-wide QKD network problem Consider a 4-connected grid lattice network (see Fig. 1). The network is large enough so that we can ignore its borders. Nodes are represented by squares. Each node is connected with its four neighbors. Links however are not represented since they do not affect the security analysis. In graph theory our network is described as follows. Network is the set of vertices $V = Z^2$. A vertex is *safe* if it is not eavesdropped. Otherwise, it is called *unsafe*. Each vertex is eavesdropped *without any trace* with probability $p_e \in [0, 1]$. As mentioned above, we focus only on the cases

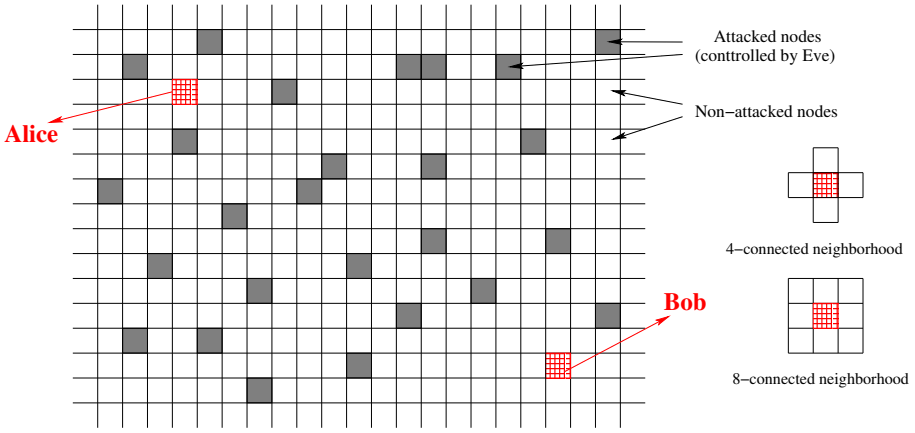

Fig. 1. Two dimensional lattice network.

where p_e is the same for all the vertices. The probability that a vertex is safe is $p_s = 1 - p_e$.

Alice and Bob are represented by vertices v_A and v_B. Alice wants to convey a secret key K to Bob. We study the secrecy probability Σ that K is not revealed to the eavesdropper Eve. If v_A and v_B are adjacent then K certainly is safe, i.e. $\Sigma = 1$. Otherwise, K must pass over l intermediate vertices $v_1, v_2, .., v_l$ whose task is to relay K. The sequence $\pi = v_A, v_1, v_2, .., v_l, v_B$ is a path from v_A to v_B. A path is *safe* if all its nodes are safe.

We define the *length* of path as the number of intermediate vertices. Since K is transmitted in π, we have $\Sigma(K) = p_s^l$. This implies that Σ is dramatically decreased with respect to (w.r.t) the length l. We focus on a simple way to improve Σ: sending a number of sub-keys $K_1, K_2, .., K_N$ by different paths $\pi_1, \pi_2, .., \pi_N$. K is computed by a bitwise XOR operation over $K_1, K_2, .., K_N$. As such, K is safe unless Eve intercepts *all* $\pi_1, \pi_2, .., \pi_N$. If the graph presents safe paths then with a larger N, K is more likley to be safe. The following questions are basic:

1. When are all the safe vertices almost certainly connected? In other words, find the condition on p_s such that $\forall \Delta \in [0, 1] : \Sigma_\infty = \lim_{N\to\infty}(\Sigma) \geq 1 - \Delta$.
2. Assume that $\Sigma_\infty \geq 1 - \Delta$. Given a pair of vertices (v_A, v_B), consider a set of N paths $\pi_1, \pi_2, .., \pi_N$ from v_A to v_B generated by a proposed routing algorithm. Let $\lambda(N)$ be the secrecy probability of the final key if N sub-keys are sent by $\pi_1, \pi_2, .., \pi_N$. Find N_0 such that for any small $\epsilon \geq \Delta$, $\epsilon \in [0, 1]$, we have: $\forall N \geq N_0 : \lambda(N) \geq 1 - \epsilon$.

3 Related work

Percolation theory This theory investigates the transition phase from the non-existence to the existence of the giant wetted cluster when we pour water at

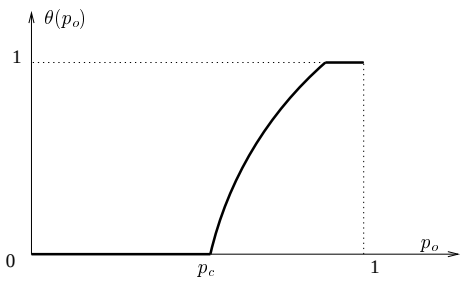

Fig. 2. The percolation probability $\theta(p_o)$.

the center of a graph [13,14,15]. The 2-dimensional site percolation model can be roughly described as follows. Let $G = (V, E)$ be a graph with vertices set V and edges set E. Vertices and edges are either *open* or *closed*. In the open status, they allow water to pass through and water make them become wetted. Otherwise, they do not allow the passage of water. All the edges are open. Each vertex is open with *open probability* $p_o \in [0,1]$. Let $\theta(p)$ be the *percolation probability* that measures the proportion of wetted vertices to open vertices. Fig. 2 roughly shows the behavior of θ w.r.t p_o. The value p_c, the *critical probability*, is the minimum p_o such that $\theta(p_o) > 0$.

The 2-dimensional site percolation's framework is similar to our network model's one. The open probability p_o and the safe probability p_s play an equivalent role. If we set $p_s = p_o$ and assume that v_A sends to v_B an infinite set of sub-keys $K_1, K_2, ..$ by an infinite set of different paths $\pi_1, \pi_2, ..$, then the secrecy probability Σ of the final key K is identical to the probability existing a safe path between v_A and v_B. However this probability is equivalent to the probability θ that almost open vertices belong to the infinite open cluster. We can apply to Σ two important properties of θ [15]:

1. θ is a non-decreasing and continuous function in the right of p_c (see Fig. 2).
2. The number of infinite open clusters is either 0 or 1 for $\theta = 0$ or $\theta > 0$, respectively.

Stochastic routing algorithms Traditional routing algorithms, such as those used on the Internet, are mostly deterministic. Tailored to be efficient, they are guessable, which is not a good property for our purpose. By contrast, stochastic routing algorithms seem to be better. The basic idea is sending randomly a packet to one of possible routes, not necessarily the "best" one. When the message holder forwards a packet, the choice of next-hop is random, following a next-hop probability distribution. The main challenge is how to determine the best next-hop probabilities that optimize a given specific goal. Previous works on stochastic routing [16,17,18] focus on performance metrics (latency, throughput, acceptance rate, etc.) which are not of major importance to QKD networks whose priority is security. Besides, the 4-connected grid topology also makes previous

optimizations on stochastic routing useless. We need to build our own stochastic routing algorithms.

4 Condition on p_s for $\Sigma \geq 1 - \Delta$

Safe connectivity function Two vertices v_A and v_B are safely connected if there exists a safe path between them. In the percolation literature, $\Sigma_\infty(v_A, v_B)$ can be interpreted as the connectivity function $\tau(v_A, v_B)$. We can use the following approximation from [13]:

$$\Sigma_\infty(v_A, v_B) = \tau(v_A, v_B) \sim \theta^2 \tag{1}$$

Given a non-negative small value Δ, we must find out the critical p_c such that $\forall p_s : p_c \leq p_s \leq 1$, we have $\Sigma_\infty \geq 1 - \Delta$. Here, we propose a heuristic method and use simulations to validate our method.

It is well known that the critical probability for the 2-dimensional lattice percolation is about 0.6. From this value to 1, the percolation probability θ is greater than zero, non-decreasingly and continuously tends to 1. Let ξ be the probability that a given vertex is encircled by unsafe vertices, we have $\theta = 1 - \xi$. From Approximation 1 we can derive the condition on ξ w.r.t a given Δ as follows:

$$\xi \leq 1 - \sqrt{1 - \Delta}$$

Our task now turns into studying ξ in the region close to 0. The trivial case where the given vertex is encircled by its four unsafe neighbors gives the lower bound of ξ, or:

$$\xi \geq (1 - p_s)^4 \quad \text{(equality i.i.f } p_s = 1) \tag{2}$$

If we set $p_s = 0.8$ then from (2) we have $\xi > 1.6 \times 10^{-3}$. It is small enough to temporarily set $p_c = 0.8$ in order to incrementally study ξ in its low-value region.

We first study ξ in the one-dimensional case. To distinguish ξ in the one-dimensional and two-dimensional cases we denote by $\xi^{(1)}$ and $\xi^{(2)}$, respectively. We measure $\xi^{(1)}$ for a given radius r (see Fig. 3.A).

$$\begin{aligned} \xi^{(1)} = &\big(\Pr(\text{At least one unsafe vertex in the left})\big) \times \\ &\big(\Pr(\text{At least one unsafe vertex in the right})\big) = (1 - p_s^r)^2 \end{aligned} \tag{3}$$

We now extend to $\xi^{(2)}$ from $\xi^{(1)}$. Assume that we are focusing on the vertex O in the two-dimensional lattice. Let $R(r)$ be the set of vertices of distance r from O. We study unsafe circuits inside $R(r)$. Denote by (see Fig. 3.C and Fig. 3.B):

- $G(r)$: the event that there are unsafe circuits that encircle the vertex O and do not exceed $R(r)$.
- $G_{LR}(r)$: the event that there are unsafe vertices at both the left and the right of the vertex O. These unsafe vertices are inside the radius r from O.

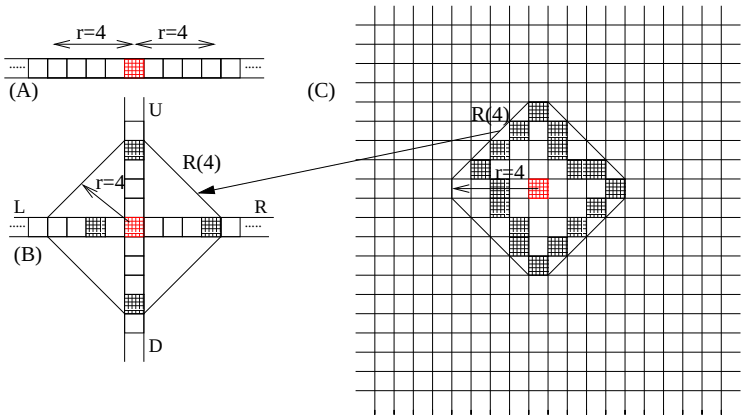

Fig. 3. Unsafe circuits in the one-dimensional and two-dimensional cases.

– $G_{UD(r)}$: the event that are unsafe vertices both above and below the vertex O. These unsafe vertices are inside the radius r from O.

Obviously, $\Pr\big(G(r)\big) \leq \Pr\big(G_{LR}(r)\big) \times \Pr\big(G_{UD}(r)\big)$. That means

$$\xi(r) = \xi^{(2)}(r) \leq \big(\xi^{(1)}(r)\big)^2 \tag{4}$$

By applying (3) to (4), we have:

$$\xi(r) \leq (1 - p_s^r)^4 \tag{5}$$

Based on $G(r)$ we define the event $G(r_1, r_2)$ is an event that there is no unsafe circuit inside the inferior $R(r_1)$ but there is an unsafe circuit inside the exterior $R(r_2)$. Let $\xi(r_1, r_2)$ be the probability that the event $G(r_1, r_2)$ appears. We have:

$$\xi(r_2) = \xi(r_1, r_2) + \xi(r_1)$$

Let r_2 tend to infinity and set $r_1 = r$, we have:

$$\xi = \xi(\infty) = \xi(r) + \xi(r, \infty) \tag{6}$$

The upper bound of ξ is estimated by applying (5) to (6):

$$\xi = \xi(\infty) \leq (1 - p_s^r)^4 + \xi(r, \infty) \tag{7}$$

If a circuit belongs to the set $G(r, \infty)$ then its length must be equal or greater than $2r$. As such, the minimum degree of p_e in the function $\xi(r, \infty)$ is $2r$ or $\xi(r, \infty) = O\big(p_e^{2r}\big) = O\big((1 - p_s)^{2r}\big)$.

We consider the ratio between ξ and $(1 - p_s)^{2r}$. From (2),

$$\lim_{r \to \infty} \frac{\xi}{(1 - p_s)^{2r}} \geq \lim_{r \to \infty} \frac{(1 - p_s)^4}{(1 - p_s)^{2r}} = \infty$$

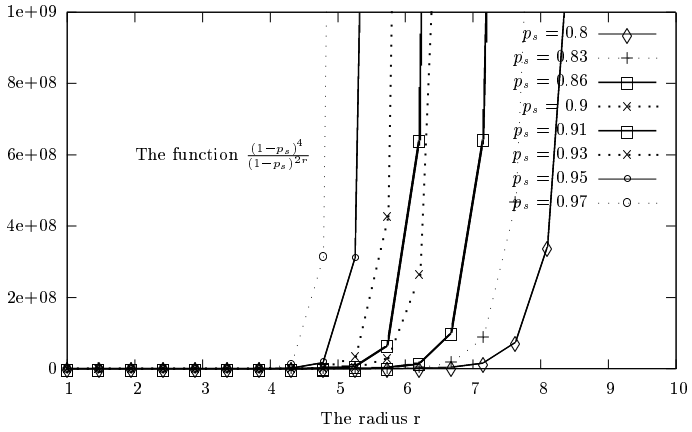

Fig. 4. The ratio between $(1-p_s)^4$ and $(1-p_s)^{2r}$

This is to say $\xi >> (1-p_s)^{2r} \sim \xi(r,\infty)$, or $\xi >> \xi(r,\infty)$ as $r \to \infty$. Fig. 4 shows the ratio between two quantities $(1-p_s)^4$ and $(1-p_s)^{2r}$ with values of p_s in $[0.8:1]$. We realize that in order to get a great ratio about 10^8, we can choose $r = 8$ for $p_s \in [0.8:0.9]$ and $r = 6$ for $p_s \in [0.9:1]$. With these choices of r, we can ignore $\xi(r,\infty)$ in the formula of the upper bound of ξ. We derive from (7) to the following approximation:

$$\xi \leq \begin{cases} (1-p_s^8)^4, & \text{if } 0.8 \leq p_s < 0.9 \\ (1-p_s^6)^4, & \text{if } 0.9 \leq p_s \leq 1 \end{cases}$$

Simulations We first determined the possible size of the world-wide quantum network according to our proposed model. The Earth's surface is 510,065,600 square kilometers. The optimal length of QKD links so far is believed to be approximately 40 km long [11]. Thus, the network size is approximatively of 600×600.

Simulation was done in the 2-dimensional grid lattice 600×600. For each experiment, we randomly generated an untrusted network w.r.t a given p_s. Then, we used the spreading algorithm to find the greatest connected safe cluster. We calculate the probability ξ_{si} that a safe vertex does not belong to the greatest safe cluster as follows:

$$\xi_{si} = 1 - \frac{\text{The number of nodes belonging to the greatest safe cluster.}}{\text{The number of all the safe nodes.}}$$

We executed 10^4 experiments for each p_s. Table 4 shows theoretic values and simulation results. We realize that as p_s increases the mean of ξ_{si} gets closer to its lower bound, and both tends to 0. For $p_s \in [0.8:0.9]$, the upper bound of ξ is important in comparison with $p_e = 1 - p_s$. This implies that the probability that the final key is eavesdropped in its transmission is greater than that of the

p_s	ξ_{lb}	$E(\xi_{si})$	ξ_{ub}
0.8	1.6×10^{-3}	2.14×10^{-3}	4.79×10^{-1}
0.83	8.35×10^{-4}	1.03×10^{-3}	3.6×10^{-1}
0.86	3.84×10^{-4}	4.47×10^{-4}	2.4×10^{-1}
0.9	1×10^{-4}	1.12×10^{-4}	4.82×10^{-2}
0.93	2.4×10^{-5}	2.7×10^{-5}	1.55×10^{-2}
0.95	6.25×10^{-6}	7×10^{-6}	4.92×10^{-3}
0.97	8.1×10^{-7}	1×10^{-6}	7.78×10^{-4}

Table 1. Lower bound ξ_{lb}, mean of simulations $E(\xi_{si})$ and upper bound ξ_{ub}.

final key being eavesdropped at the transmitter. This is out of our interest. By contrast, for $p_s \in [0.93 : 1]$ the upper probability of ξ is approximate or less than the probability of this vertex itself being unsafe. This seems more interesting. Table 4 also suggests that $\xi \sim \xi_{lb} = (1 - p_s)^4$ for $p_s \in [0.93 : 1]$.

5 Applying stochastic routing algorithms

5.1 Some proposed routing algorithms

An adaptive drunkard's routing algorithm (ADRA) In the classic drunkard's walk problem, the next-hop probability distribution is unbiased. We propose an adaptive drunkard's routing algorithm, named ADRA, that is biased. The idea is to give a bigger chance for the vertex that is closer to the destination vertex. Assume that the vertex v_A wants to send a message to the vertex v_B. The vertex v_A computes next-hop probabilities for its neighbors. This computation is based on the coordinate correlations between neighbors and v_B. The higher probability is given to the vertex that is closer to v_B. Then the vertex v_A randomly chooses one of its neighbors to forward the message, but according to the probability distribution that has been computed. Anyone that subsequently receives the message would do the same thing and the chain of communication would continue to reach to v_B.

A constant-length stochastic routing algorithm (l-SRA) The *length* of a path is the number of the vertices belonging to the path. A vertex may be counted as many times as the path runs through this vertex. The *distance* between two vertices is the length of the shortest path between these vertices.

Our *constant-length stochastic routing algorithm*, called l-SRA(l) or l-SRA for short, is a stochastic routing algorithm that takes a value l as input and tries to transmit a message by a random path of length l.

Assume that there are some different paths $\pi_1, \ldots, \pi_m$ that hold $l_{(\pi_1)} = \ldots = l_{(\pi_m)} = l$. Note that in the 4-connected grid lattice, it must $l = d + 2 \times k, k \geq 0$. When sending a message l-SRA will choose randomly a path π_i among $\pi_1, \ldots, \pi_m$ according to a probability distribution that holds two following conditions:

1. $\forall i, 1 \leq i \leq m : 0 \leq \Pr(l\text{-SRA(l) takes } \pi_i) \leq 1$
2. $$\sum_{i=1}^{m} \Pr(l\text{-SRA(l) takes } \pi_i) = 1 \quad (8)$$

Theorem 1. *The probability that l-SRA(l) chooses successfully a safe path to send one message depends only on the safe probability p and the length l, not on the distance d between Alice and Bob:*

$$\Pr(1, p, d, l\text{-}SRA(l)) = p^l$$

A parameterized-length stochastic routing algorithm (k-SRA) This algorithm takes an input parameter $k > 1$, and tries to transmit the message by a path of length $l \leq k \times d$. We call this algorithm k-SRA(k) or k-SRA for short. It is built based on l-SRA. The idea is as follows. When k-SRA(k) receives the input $k > 1$, it considers the paths of length $l \leq k \times d$. Note that the difference between the length and the distance cannot be an odd number. Therefore, the possible lengths are $d, (d+2), \ldots, (d + 2 \times \lfloor \frac{(k-1)\times d}{2} \rfloor)$. When sending a message k-SRA(k) chooses randomly for l a value among $d, (d+2), \ldots, (d+2\times \lfloor \frac{(k-1)\times d}{2} \rfloor)$ according to the uniform distribution, i.e:

$$\forall i, 0 \leq i \leq u = \lfloor \frac{(k-1)\times d}{2} \rfloor : \Pr\left((d + 2 \times i) \text{ is taken for } l\right) = \frac{1}{(k+1)\times d}$$

Once l was chosen, k-SRA uses l-SRA to send the message. This implies that the message will take a random path that has the length l.

Theorem 2. *The probability that k-SRA(k) chooses successfully a safe path to send one message depends on the safe probability p, the input parameter k, and also the distance d between Alice and Bob:*

$$\lambda = \Pr(1, p, d, k\text{-}SRA(k)) = \frac{p^d \times (1 - p^{2\times(u+1)})}{(u+1)\times(1-p^2)} \quad (9)$$

5.2 Our proposed routing algorithms in some attack strategies

We consider two attack strategies of Eve:

1. Dynamic attack: To catch a set of N messages Eve frequently re-chooses nodes being attacked.
2. Static attack: Eve keeps her choice of the nodes being attacked until all N messages have been sent.

Because the algorithm ADRA is based on random walk, it does not give rigorous mathematical results. Its performance is estimated by experimental statistics. The algorithm l-SRA is not a real routing solution. This algorithm only executes one sub-task of the algorithm k-SRA. The algorithm k-SRA presents some rigorous bounds.

Theorem 3. *If Eve executes a dynamic attack, then the probability that there is at least one safe path in N routings of k-SRA(k) depends on N, the safe probability p, the input parameter k, and the distance d between Alice and Bob:*

$$\Pr(N, p, d, k\text{-}SRA(k)) = 1 - (1 - \lambda)^N$$

Where λ is evaluated in (9).

We have a lemma derived directly from the theorem 3.

Lemma 1. *If Eve executes a dynamic attack, given ϵ and k-SRA(k), then we have the threshold N_0 responding to the second question stated in Section 2:*

$$N_0 = \frac{\lg(\epsilon)}{1 - \lg(\lambda)}$$

Where λ is evaluated in (9).

Theorem 4. *If Eve executes a static attack, then the upper bound of the probability that there is at least one safe path in N routings of k-SRA(k) depends on N, the safe probability p, the input parameter k, and the distance d between Alice and Bob:*

$$\Pr(N, p, d, k\text{-}SRA(k)) \leq 1 - (1 - \lambda)^N$$

Where λ is evaluated in (9). The equality is possible when $N \leq 4$.

We have a lemma derived directly from the theorem 4.

Lemma 2. *If Eve executes a static attack, given ϵ and k-SRA(k), we have the threshold N_0 responding to the second question stated in Section 2:*

$$N_0 \geq \frac{\lg(\epsilon)}{1 - \lg(\lambda)}$$

Where λ is evaluated in (9). The equality is possible when $N_0 \leq 4$.

5.3 Simulations

ADRA's simulations The next-hop probabilities computation can vary to result in many ADRA's variants. Here we reused the next-hop probabilities computation presented in [19]. Then, we ran simulations in the lattice 600×600, in varying the safety probability $p_s \in [0.93 : 1]$ and the distance d_{AB} between Alice and Bob [19, 20]. For each p_s, we generated a network with randomly spread eave-droppers. For each distance d_{AB}, we generated 400 (Alice, Bob) pairs. For each such pair, we ran 400 experiments. In each one we generated stochastic routes from Alice to Bob until we find a safe one (i.e., a route with no Eve). For each 400 experiments we gathered the largest number of messages that were needed. To avoid sending an infinite number of messages, we set the maximum effort to 10^4 messages.

Table 5.3 presents simulation results. This suggests that there exists a threshold of the number of sending messages above which we can be almost certain that there exists at least one safe message.

	p_s								p_s						
d	0.99	0.98	0.97	0.96	0.95	0.94	0.93	d	0.99	0.98	0.97	0.96	0.95	0.94	0.93
1	8	12	12	22	14	12	14	10	149	169	340	1267	3731	1267	2854
2	44	105	122	68	82	425	106	20	127	338	829	9300	×	×	×
3	87	51	273	99	122	233	439	30	315	1987	2908	×	×	×	×
4	95	171	160	408	244	1125	476	40	386	4111	×	×	×	×	×
5	66	61	186	917	286	967	2149	50	437	×	×	×	×	×	×
6	34	397	356	377	644	583	921	60	656	×	×	×	×	×	×
7	43	194	155	395	625	420	2102	70	1911	×	×	×	×	×	×
8	72	1645	224	414	936	773	1663	80	3117	×	×	×	×	×	×
9	53	185	477	386	585	717	2794	90	7039	×	×	×	×	×	×
10	149	169	340	1267	3731	1267	2854	100	4117	×	×	×	×	×	×
								110	×	×	×	×	×	×	×

Table 2. Worst cases's experiment results. Symbol × stands for more than 10,000.

***k*-SRA's simulations** Simulations were implemented in the lattice 600×600. We ran 10^4 experiments. The table 5.3 shows the lower bounds, the simulation values, and the upper bounds for the case of $k = 2$ and $d = 10$ with $p_s = 0,93; 0.95; 0.97; 0.99$. Note that the lower bound holds if N messages have taken the only possible path. The convergence of the experimental results to their upper bound is significant. We realize that the secrecy probability of the final key is a non-decreasing function. As the number of sent messages increases, this probability converges to its upper bound. Moreover, both tend to 1 as $N \to \infty$.

6 Conclusions

We investigated constraints of quantum networks, in particularly, the ineluctable probability that some nodes are compromised. Given the distance between source and destination, we proposed routing algorithms and estimated the number of pieces that the message must be divided into with respect to the distance and the compromising probability distribution imposed over nodes. The principle result of our work is that it opens another door allowing to investigate QKD networks using percolation theory and stochastic routing.

A lot of work remains to be done in the future. For example, we need to take into account key authentication to complete our key exchange scheme. The eavesdropping distribution was uniform in this paper. More complex probability distributions seem more interesting. Studying other topologies will be of significance, grids are only the first step. We also aim at finding rigorous and tight formulas. Besides, we must improve our stochastic routing proposals, e.g. hiding routing information as onion routing. We attach importance to throughput and computational overhead in practice. We plan to carry out a cost estimation with respect to today's QKD technology.

	$p_s = 0.93$				$p_s = 0.97$		
N	$\lambda_{lb}(\%)$	$\lambda_{si}(\%)$	$\lambda_{ub}(\%)$	N	$\lambda_{lb}(\%)$	$\lambda_{si}(\%)$	$\lambda_{ub}(\%)$
1	34.71	42.54	34.71	1	63.66	69.99	63.66
10	34.71	80.57	98.59	10	63.66	93.84	98.59
100	34.71	95.36	100	100	63.66	98.94	100
1000	34.71	99.52	100	1000	63.66	99.94	100
10000	34.71	99.96	100	10000	63.66	100	100
	$p_s = 0.95$				$p_s = 0.99$		
N	$\lambda_{lb}(\%)$	$\lambda_{si}(\%)$	$\lambda_{ub}(\%)$	N	$\lambda_{lb}(\%)$	$\lambda_{si}(\%)$	$\lambda_{ub}(\%)$
1	47.04	54.31	47.04	1	86.05	88.75	86.05
10	47.04	87.96	98.59	10	86.05	98.40	100
100	47.04	97.59	100	100	86.05	99.81	100
1000	47.04	99.84	100	1000	86.05	99.99	100
10000	47.04	100	100	10000	86.05	100	100

Table 3. Lower bound, experimental results, upper bound of the key secrecy for $p_s = 0.93; 0.95; 0.97; 0.99$. λ_{si} is the percentage in 10^4 experiments done.

Appendix

Proof of theorem 1 $\Pr\left(1, p, d, l\text{-SRA(l)}\right)$

$$= \sum_{i=1}^{k} \Big(\Pr\left(l\text{-SRA(l) takes } \pi_i\right) \times \Pr(\pi_i \text{ is safe}) \Big) = \sum_{i=1}^{k} \Big(\Pr\left(l\text{-SRA(l) takes } \pi_i\right) \times p^l \Big)$$

$$= \Big(\sum_{i=1}^{k} \Pr\left(l\text{-SRA(l) takes } \pi_i\right) \Big) \times p^l = p^l \quad (\text{ from (8)})$$

Proof of theorem 2 $\lambda = \Pr\left(1, p, d, k\text{-SRA(k)}\right)$

$$= \sum_{l=d,..,d+2u} \Big(\Pr\left(k\text{-SRA(k) takes } l\right) \times \Pr\left(l\text{-SRA}(l) \text{ takes a safe path}\right) \Big)$$

$$= \sum_{l=d,..,d+2u} \left(\frac{1}{(u+1)} \times \Big(\Pr\left(1, p, d, l\text{-SRA(l)}\right) \Big) \right)$$

$$= \frac{1}{(u+1)} \times \left(\sum_{l=d,..,d+2u} \Big(\Pr\left(1, p, d, l\text{-SRA(l)}\right) \Big) \right)$$

$$= \frac{1}{(u+1)} \times \left(\sum_{l=d,..,d+2u} p^{(l)} \right) \quad = \quad \frac{p^d \times (1 - p^{2(u+1)})}{(u+1) \times (1 - p^2)}$$

Proof of theorem 3 It is a memoryless system. From (9),

$$\begin{aligned}
&\Pr(\text{All the N trials are failed}) = \big(1 - \Pr(\text{A trial is successful})\big)^N = (1-\lambda)^N \\
&\rightarrow \Pr(N, p, d, k\text{-SRA(k)}) = \Pr(\text{At least one of N trials is successful}) \\
&\quad = 1 - \Pr(\text{All the N trials are failed}) = 1 - (1-\lambda)^N
\end{aligned}$$

Proof of theorem 4 We must take into account the path dependence of N paths taken by N messages sent. The probability that k-SRA(k) takes an unsafe path for each trial is:

$$\begin{aligned}
\overline{\Pr(1, p, d, k\text{-SRA(k)})} = \sum_{d \le l \le k \times d} \Big(&\Pr\big(k\text{-SRA(k) takes } l\big) \times \\
&\Pr\big(l\text{-SRA}(l) \text{ takes an unsafe path}\big)\Big) = 1 - \lambda
\end{aligned} \tag{10}$$

The probability of N messages being intercepted is:

$$\begin{aligned}
\overline{\Pr(N, p, d, k\text{-SRA(k)})} = \sum_{\substack{d \le l_1 \le k \times d \\ \cdots \\ d \le l_N \le k \times d)}} &\Bigg(\Pr\big(k\text{-SRA(k) takes } (l_1, \ldots, l_N)\big) \times \\
&\Bigg(\sum_{\substack{l_{\pi_1} = l_1, \\ \cdots \\ l_{\pi_N} = l_N}} \Big(\Pr(l\text{-SRA takes } \pi_1 \ldots \pi_N) \times \big(\Pr(\pi_1 \ldots \pi_N \text{ are failed})\big)\Big)\Bigg)\Bigg)
\end{aligned} \tag{11}$$

For a given path set $(\pi_1, \ldots, \pi_N)$, we can prove the following inequality:

$$\Pr(\pi_1, \ldots, \pi_N \text{ are failed}) \ge \prod_{i=1}^{N} \Pr(\pi_i \text{ is failed}) \tag{12}$$

Where the equality holds i.i.f $\pi_1, \ldots, \pi_N$ are independent.

We first prove with $N = 2$. Assume that π_1, π_2 have the length l_1, l_2 respectively, and have l common nodes $(0 \le l \le \min(l_1, l_2))$. We have:

$$\begin{aligned}
\Pr(\pi_1, \pi_2 \text{ are failed}) &= p^l \times (1 - p^{(l_1 - l)}) \times (1 - p^{(l_2 - l)}) + (1 - p^l) \\
&= (1 - p^{(l_1)}) \times (1 - p^{(l_2)}) + (p^{(l_1 + l_2 - l)} - p^{(l_1 + l_2)}) \\
&\ge (1 - p^{(l_1)}) \times (1 - p^{(l_2)}) = \Pr(\pi_1 \text{ is failed}) \times \Pr(\pi_2 \text{ is failed})
\end{aligned}$$

Inequality (12) was proven with $N = 2$. We iterate this to obtain (12) for $\forall N$. Note that the equality holds iff $\pi_1 \ldots \pi_N$ are separated. In the square 4-connected lattice there are maximum 4 separated paths between Alice and Bob. Thus, if $N > 4$, the equality for (12) cannot appear. By applying (12) to (11), we have:

$$\overline{\Pr(N,p,d,k\text{-SRA(k)})} > \sum_{\substack{d\le l_1\le k\times d\\ \cdots\\ d\le l_N\le k\times d)}} \Big(\Big(\prod_{i=1}^{N}\Pr\left(k\text{-SRA(k) takes } l_i\right)\Big)\times$$
$$\Big(\sum_{\substack{l_{\pi_1}=l_1,\\ \cdots\\ l_{\pi_N}=l_N}}\big(\prod_{i=1}^{N}\Pr(l\text{-SRA takes }\pi_i)\big)\times\big(\prod_{i=1}^{N}\Pr(\pi_i\text{ is failed})\big)\Big)\Big)$$
$$=\sum_{\substack{d\le l_1\le k\times d\\ \cdots\\ d\le l_N\le k\times d)}}\Big(\Big(\prod_{i=1}^{N}\Pr\left(k\text{-SRA(k) takes } l_i\right)\Big)\times$$
$$\Big(\prod_{l_j=l_1}^{l_N}\Big(\sum_{l_{\pi_i}=l_j}\big(\Pr(l\text{-SRA takes }\pi_i)\times\Pr(\pi_i\text{ is failed})\big)\Big)\Big)\Big)$$
$$=\sum_{\substack{d\le l_1\le k\times d\\ \cdots\\ d\le l_N\le k\times d)}}\Big(\prod_{i=1}^{N}\Pr\left(k\text{-SRA(k) takes } l_i\right)\times\prod_{l_j=l_1}^{l_N}\Pr\left(l\text{-SRA}(l_j)\text{ takes an unsafe path}\right)\Big)$$
$$=\sum_{\substack{d\le l_1\le k\times d\\ \cdots\\ d\le l_N\le k\times d)}}\Big(\Big(\prod_{i=1}^{N}\Pr\left(k\text{-SRA(k) takes } l_i\right)\times\Pr\left(l\text{-SRA}(l_j)\text{ takes an unsafe path}\right)\Big)\Big)$$
$$=\prod_{i=1}^{N}\Big(\Big(\sum_{d\le l_i\le k\times d}\Pr\left(k\text{-SRA(k) takes } l_i\right)\times\Pr\left(l\text{-SRA}(l_i)\text{ takes an unsafe path}\right)\Big)\Big)$$
$$=\prod_{i=1}^{N}\Big(\Pr\left(k\text{-SRA(k) takes an unsafe path}\right)\Big)=(1-\lambda)^N\ \text{(from (10))}$$

Thus,

$$\Pr(N,p,d,k\text{-SRA(k)})=1-\overline{\Pr(N,p,d,k\text{-SRA(k)})}=1-(1-\lambda)^N$$

Acknowledgements We thank Steve Frank and Daniela Becker for their proof-reading. All the mistakes are ours.

References

1. Bennett, C., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proc. of IEEE Int. Conf. on Computers, Systems, and Signal Processing, Bangalore, India (December 1984) 175–179
2. Mayer, D.: Unconditional security in quantum cryptography. Journal of the ACM **48** (May 2001) 351–406

3. Lo, H.K.: A simple proof of the unconditional security of quantum key distribution. Journal of Physics A **34** (September 2001) 6957–6967
4. Chau, H.: Practical scheme to share a secret key through an up to 27.6% bit error rate quantum channel. Phys. Rev. A **66** (December 2002) 060302
5. Elliott, C., Pearson, D., Troxel, G.: Quantum cryptography in practice. In: Proc. of the Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, Karlsruhe, Germany (August 2003) 227–238
6. Elliott, C., Colvin, A., Pearson, D., Pikalo, O., Schlafer, J., Yeh, H.: Current status of the DARPA quantum network (March 2005) `http://arxiv.org/abs/quant-ph/0503058v2`.
7. Bellot, P., Gallion, P., Guilley, S., Danger, J.L.: The hqnet project (2006) `http://hqnet.enst.fr`.
8. Qing Xu., e Silva, M.B.C., Danger, J., Guilley, S., Gallion, P., Bellot, P., Mendieta, F.: Towards Quantum key distribution System using Homodyne Detection with Differential Time-multiplexed Reference. In: Proc. of the 5th Int. Conf. on Computer Sciences, Research Innovation and Vision for the Futur, Hanoi, Vietnam (March 2007) 158–165
9. Kimura, T., Nambu, Y., Hatanaka, T., Tomita, A., Kosaka, H., Nakamura, K.: Single-photon interference over 150-km transmission using silica-based integrated-optic interferometers for quantum cryptography criterion. Japanese Journal of Applied Physics. **43** (September 2004) L1217–L1219
10. Elliott, C.: Building the quantum network. New Journal of Physics **4** (July 2002) 46.1–46.12
11. Alléaume, R., Roueff, F., Maurhart, O., Luthenhaus, N.: Architecture, Security and Topology of a global Quantum key distribution Network. In: IEEE/LEOS Summer Topical Meeting on Quantum Communications in Telecom Networks, Quebec (July 2006)
12. Alléaume, R., al.: Secoqc white paper on quantum key distribution and cryptography (January 2007) `http://arxiv.org/abs/quant-ph/0701168`.
13. Grimmett, G.: Percolation. Second edn. Springer-Verlag (1999)
14. Hughes, B.D.: Random walks and random environments. Volume 1. Oxford University Press (1995)
15. Hughes, B.D.: Random walks and random environments. Volume 2. Oxford University Press (1995)
16. Bohacek, S., Hespanha, J.P., Lee, J., Lim, C., Obraczka, K.: Game theoretic stochastic routing for fault tolerance and security on computer networks. IEEE Transactions on Parallel and Distributed Systems **18** (September 2007) 1227–1240
17. Hespanha, J., Bohacek, S.: Preminilary results in routing games. In: American Control Conference. Volume 3., Arlington, Virginia, USA (June 2001) 1904–1909
18. Bohacek, S., Hespanha, J.P., Obraczka, K., Lee, J., Lim, C.: Enhancing security via stochastic routing. In: Proc. 11th Int. Conf. on Computer Communication and Networks, Miami, Florida, USA (October 2002) 58–62
19. Le, Q.C., Bellot, P., Demaille, A.: Stochastic Routing in Large Grid Shaped Quantum Networks. In: Proc. of the 5th Int. Conf. on Computer Sciences, Research Innovation and Vision for the Futur, Hanoi, Vietnam (March 2007) 166–174
20. Le, Q.C., Bellot, P., Demaille, A.: On the security of Quantum Networks: a proposal framework ans its capacity. In: Proc. of the Int. Conf. on New Technologies, Mobility, and Security, Paris, France (May 2007) 385–396