Secure Key Exchange in Wireless Networks

László Czap Christina Fragouli
École Polytechnique Fédérale de Lausanne, Switzerland
Email: {laszlo.czap christina.fragouli}@epfl.ch

***Abstract*—We investigate the problem of exchanging a secret key within a group of wireless devices. In particular, we are interested in information theoretically secure key exchange schemes that enable honest nodes of a multi-hop network to establish a secret group key in the presence of an eavesdropping adversary. Similarly to [1], the scheme presented here makes use of erasures over the wireless channel. In essence, this work can be seen as an extension of [1] for multi-hop networks instead of a single-hop network. We extend the approach and investigate the performance of our proposed protocol.**

***Keywords*—secrecy, wireless networks, group keys**

I. Introduction

We consider information theoretically secure secret key exchange over the wireless channel within a group of nodes. In particular, a multi-hop wireless network is considered, where a set of connected nodes wish to set up a common group key securely in the presence of a passive eavesdropping adversary.

Compared to cryptographic alternatives, such a scheme has significance because it does not rely on computational assumptions, and provides guarantees against a computationally unbounded adversary as well. Moreover, we also believe that its computational complexity can be lower than that of cryptographic counterparts that heavily rely on public key operations in most cases.

We consider broadcast erasure channels and we take advantage of the fact that spatially separated users have independent channels. Packets sent over the wireless channel may or may not be correctly received by other devices in the vicinity and the probability of an erasure is different towards each candidate receiver depending on the node's location and the current status of the particular channel. As a result, receivers – including the adversary – receive correctly a different set of transmitted packets. This property enables to benefit from packets that are erased on the adversary's channel but correctly received by honest devices.

A key exchange scheme was built on this principle in [1] for the setting where both honest nodes and the adversary are able to overhear the broadcast communication of a selected source node who initiates the key exchange. Here, we aim to propose a scheme of the same vein that achieves group secrecy in a multi-hop network, where the underlying topology does not ensure that all nodes hear a selected source. We are going to describe and analyze the key exchange scheme in a specific circle topology as a canonical example and then argue to applicability in more general networks.

The rest of the paper is organized as follows. In Section II we describe the specific network we consider. In Section III we give an overview of the one-hop key exchange scheme and we present our multi-hop key exchange protocol in Section IV. Section V provides the analysis of the scheme. We summarize related work in Section VI. We discuss general applicability of our scheme and draw conclusions in Section VII.

II. System model

We assume a multi-hop wireless network where the goal of the nodes is to create a common group key that is perfectly secret from the adversary. We do not assume any common randomness available for them initially, however nodes can generate random bits independently of each other.

a) Topology: Nodes are deployed in a circle topology where nodes can communicate directly only with their left and right neighbor along the circle. For ease of description we assume an odd number of nodes, but this does not effect our conclusions. Hence, in the circle topology the number of nodes equals $2d+1$, where d is the maximal distance between two nodes in hops. See Figure 1 for illustration, when $d = 3$.

b) Channel: The wireless channel can be utilized either as a reliable broadcast channel or as a broadcast erasure channel. A packet sent over the erasure channel is either correctly received or provides no information to the receiver. To utilize the wireless channel as reliable nodes can apply an error correcting code to their packets. From a cost point of view, in our analysis we do not distinguish between the two modes of channel usage, we treat the total number of all wireless transmissions in the network as the communication cost of the scheme. For simplicity, we assume that the probability δ that a packet is erased on the channel is the same between honest nodes in the whole network and erasures are independent for every receiver. We also assume that erasure probabilities are known to participating nodes. Finally, when using the erasure channel, the feedback of correctly received packets play an important role. The size of this feedback is considered negligible compared to the size of data packets.

c) Adversary: We aim to deal with an adversary, Eve, who can eavesdrop on a selected link in the network. The adversary is static and listens on a single link, but honest nodes are not aware of the location of Eve. A packet sent over the erasure channel can be received by Eve with probability $1 - \delta_E$. The probability of erasure towards the adversary is also assumed to be independent from erasures towards any other nodes. Of course, a packet sent over the reliable channel can be correctly received by Eve as well.

d) Performance measure: We are going to investigate the cost of creating a unit size secret key in the group in terms of

978-1-61284-140-3/11/$26.00 ©2011 IEEE

the total number of wireless transmissions t in the network. We refer to this as the cost of a secret key or simply the cost of our scheme. This performance measure is inversely proportional to the secrecy rate if the communication rate of both channel modes are taken into account.

We will be interested in information theoretically secure keys. Namely, we require that the group key is independent of the information that the adversary may gain during the protocol run: $H(K) = H(K|A)$, where K is the group key and A is the knowledge of Eve.

III. Key exchange in a single-hop network

Our scheme crucially uses the following proposition that we repeat here for convenience [1]. If Alice has N packets and Eve has overheard an (unknown) subset of κ of them, then Alice can create $N - \kappa$ linear combinations of the N packets that are information theoretically secure from Eve.

There are m nodes in the network, one of them is the source node, who initiates the key exchange, and there are $m - 1$ receivers besides. The underlying topology is such that all nodes can receive the transmissions of the source and they are also able to send feedback to it. The channel is characterized exactly as we described in the previous section. The protocol has the following steps:

Single-hop protocol:

1) The source generates N random packets (x-packets) and sends them over the erasure channel. All nodes send a feedback to the sender of which packets were received correctly. Nodes are expected to receive $N(1 - \delta)$ x-packets, out of which $N\delta_E(1 - \delta)$ packets are not received by the adversary. These packets serve as a basis for the common key.
2) Based on the feedback of the receivers, the source can generate $N(1 - \delta^{(m-1)})\delta_E$ linear combinations of x-packets that are ensured to be secret from the adversary – this is the expected number of x-packets that was correctly received by at least one honest node, but not Eve. The coding scheme applied here relies on the fact that Eve could receive only a fraction of x-packets and does not depend on which x-packets are those. The packets generated this way are referred as y-packets. Every node is expected to be able to also generate $N(1 - \delta)\delta_E$ of the y-packets from the packets it received through the erasure channel.
3) The source generates linear combinations of y-packets (z-packets) such that all receivers can reconstruct all the y-packets from the y-packets that they already have and from the z-packets. z-packets are then sent by reliable transmissions, so Eve can eavesdrop all of them. Now, all nodes can reconstruct all y-packets.
4) Linear combinations of y-packets that are independent of z-packets are used as the shared key in the group. These packets – referred as k-packets – are perfectly secure from the eavesdropper.

As we can see, the scheme builds on the feedback that receivers give. The feedback needs to contain some indexes only, so the overhead of the feedback is considered negligible.

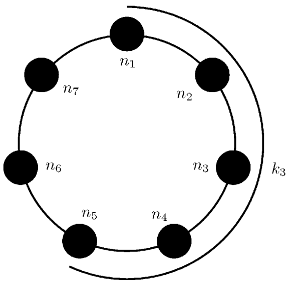

Figure 1. Example for $d = 3$, $k = 2$. Local keys are created by up to the k-hop neighbors of a node. As an example k_3 is shown.

IV. Key exchange in multi-hop networks

In the following, we describe an algorithm that achieves secrecy with a cost of $O(d)$ wireless transmissions per node.

A. Two special cases

To illustrate the difference between the multi-hop network and the single-hop case, let us first consider two special cases.

1) Case 1: The simplest possible setting is when $\delta_E = 1$ and $\delta = 0$, i.e., honest nodes communicate reliably through the erasure channel, while the adversary is not present at all. Here, the problem of key exchange reduces to the problem of one-to-all broadcast communication, because nodes can send generated keys to each other without revealing any information to the adversary. The problem of such broadcast communication is analyzed in [2], [3]. It is shown that the most efficient broadcast communication is achieved if all nodes have a packet to broadcast. Using this protocol the key exchange for this simple case is as follows :

Key exchange without the adversary:

1) Initially, every node generates a random key.
2) Nodes broadcast their keys to their two neighbors. This way, every node receives two new keys.
3) Next, nodes transmit the xor of the keys they just received. This allows both of their neighbor to decode one novel key.
4) They repeat the previous step always with the newly received keys until all nodes have received all keys. This requires d transmissions from each node.
5) As a result nodes share $2d + 1$ common keys.

From the optimality of this broadcast protocol, it follows that this process provides the lowest possible cost of any key exchange scheme, because the weakest possible adversary against the strongest possible honest nodes was assumed. Hence, the cost of a secret key with this method

$$\frac{t}{key\ size} = \frac{(2d+1)d}{2d+1} = d$$

serves as a basis for comparison. Note that in the one-hop case, for $\delta_E = 1$, the corresponding cost is 1.

2) Case 2: Now, we discuss another special case, when the adversary does not experience erasures at all, i.e. $\delta_E = 0$. This means that the eavesdropper overhears all communications that are sent through the observed link whether it is a reliable transmission or not. Clearly, in this case honest node can not gain by using the erasure channel, so we can restrict ourselves to the use of reliable transmissions only. Here, we can only make use of the limitation of the adversary. Namely, that Eve can not eavesdrop every communication, but a single link only. However, the location of the adversary is not known, so again we are looking for a broadcast scheme that minimizes the maximal information that is sent through any link of the network. The previously described broadcast protocol serves as a solution for this problem also. Thus, a possible key exchange protocol for this case is exactly the same as in the first case, except for the last step, because the adversary now successfully overhears some of the keys. During the broadcast protocol, the adversary overhears d transmissions from both of its neighbors, that means $2d$ independent combinations of the keys. This reduces the achievable key size to $(2d+1) - 2d = 1$. Nodes can produce one linear combination of the keys (e.g. the xor of them) as a group secret key. The cost of a secret key is now

$$\frac{t}{key\ size} = \frac{(2d+1)d}{1} = (2d+1)d.$$

In the one-hop network, no secrecy could be achieved if $\delta_E = 1$ resulting in infinite cost. Note that this case incorporates the worst possible case also, when $\delta = 1$. These two examples illustrate that moving from one-hop to the multi-hop have significant consequences. The cost of a secure key ranges within the interval $(1;\infty)$ for a one-hop network, while within $(d;(2d+1)d)$ for the mutli-hop network. The second example also shows that secrecy can be achieved by exploiting the network topology only and not the benefit of the erasure channel. In the general scheme we are going to exploit both ways of achieving secrecy, combining them together.

B. General scheme

The two special cases already indicated that the best possible performance is achieved if all nodes of the network generate a key and acts as source. This is a consequence of the symmetry in the network. We can state in general that:

Lemma 1. *There exists a scheme that achieves the optimal performance and is symmetric.*

Proof: Let us find the scheme with optimal performance. If it is not symmetric, we can easily create a symmetric scheme by repeating it with all possible different configurations. This symmetric scheme has the same performance as the optimal. ■

For this reason, in our proposed scheme, every node performs the same actions.

A high-level description of the our exchange scheme is the following:

1) *Local key exchange.* Every node establishes a local key with its k hop neighbors. For this purpose, they make use of the erasure channel. This key generation for $k = 1$ is illustrated in Figure 1. We already note that possibly these keys are not all perfectly secure from the adversary.
2) *Dissemination of local keys.* Nodes share the local keys through reliable transmissions by performing $d-k$ steps of the described broadcast protocol.
3) *Generating group keys.* An appropriate number of linear combinations of the local keys are created to establish the secret group key.

The most straightforward approach is to create local keys such that they are all perfectly secure from the adversary. In this case all nodes assume the worst case possibility regarding the location of the adversary. However, in reality there are $2(d-k)+1$ nodes in the network that are located more than k hops away from the adversary, thus none of their y-packets are eavesdropped and they could generate larger local keys securely. We do not know which nodes could do so as the location of the adversary is not known, but in the end we know the expected number of eavesdropped packets and take it into account in the final step, when the group key is generated. For this reason, we introduce a parameter $\beta \in (0;1]$ that defines the size of the generated local keys as a ratio between the number of key packets and the number of y-packets they are generated of. This way we may allow that not all local keys are perfectly secure but achieve a larger key size and a better performance in the end.

Next, we describe each step of the scheme in detail.

Local key exchange:

1) Every node generates N random packets, the x-packets. These packets are transmitted through the erasure channel.
2) Nodes are expected to receive $N(1-\delta)$ x-packets from each of their two neighbors, say $x_1^1, x_2^1, \ldots, x_n^1$ and $x_1^2, x_2^2, \ldots, x_n^2$. They xor the newly received x-packets and transmit the resulting packets $x_1^1 \oplus x_1^2 \ldots x_n^1 \oplus x_n^2$ again through the erasure channel. This step is repeated k times. Note that correctly received xor-ed packets can always be decoded regardless of the previous erasures.
3) By the end of the previous step, from its i-hop neighbor, a node is expected to receive $N(1-\delta)^i$ x-packets. Nodes generate y-packets as linear combinations of their own x-packets. The number of y-packets they can securely generate for sure (assuming a neighboring adversary) is $N(1-\delta^2)\delta_E$ just like in [1].
4) Nodes create linear combinations of y-packets (z-packets) such that its two immediate neighbors can reconstruct all its y-packets. The number of required z-packets is $N\delta_E((1-\delta^2)-(1-\delta))$. The z-packets are sent using reliable transmissions.
5) One-hop neighbors can now reconstruct all y-packets of a node. However, they do not simply forward z-packets, but they create new z-packets such that the two hop neighbors can also reconstruct. This requires $N\delta_E((1-\delta^2)-(1-\delta)^2)$ z-packets. Every node creates two sets of such z-packets, say $z_1^1, z_2^1, \ldots, z_{n'}^1$ and

$z_1^2, z_2^2, \ldots, z_{n'}^2$, from the y-packets of the two one-hop neighbors. The elements of the two sets are xor-ed and resulting packets $z_1^1 \oplus z_1^2 \ldots z_{n'}^1 \oplus z_{n'}^2$ are transmitted reliably. Note that knowing the coefficients used to create z-packets every node can decode the z-packets of interest, as it can compute the other z-packets from the already reconstructed y-packets.

6) The process of the previous step is performed until k-hop neighbors also can reconstruct y-packets. In the ith round, $N\delta_E((1-\delta^2)-(1-\delta)^i)$ packets are transmitted per node. At the end of this phase, all nodes know all y-packets of its $0 \ldots k$ hop neighbors.
7) Linear combinations of y-packets of each node are generated as local keys. Nodes create $N\delta_E\beta(1-\delta^2)$ such linear combinations. We call these packets k-packets and they together serve as the local keys. All nodes have $1+2k$ such local keys.

Dissemination of local keys:

1) Nodes already have the keys generated from the y-packets of up to their k-hop neighbors. They finish disseminating the local keys according to the broadcast protocol: they first transmit the xor of the local keys of their two k-hop neighbors, then in the next round the xor of the newly decoded two keys and so on until all local keys are known by all nodes. This requires $d-k$ reliable transmissions per node.

Generating group keys:

1) This step is similar to the step when y-packets were generated of x-packets. With the same method, group keys are generated as linear combinations of all k-packets of the network. We have to take into account the information that the adversary could learn of the space of k-packets. Not all k-packets were generated perfectly securely, moreover, during the previous step the adversary could eavesdrop $2(d-k)$ local keys out of the $(2d+1)$ local keys in the system. We compute the achievable key size with respect to different values of β in the next section.

Note that this scheme incorporates the two special cases as well for $k=0$ and for $k=d$. The coding schemes that we use when creating secure y-packets and z-packets are known and can be found in [1], [4].

C. Example key exchange

We illustrate the steps of the above scheme with a simple example. Let us assume that $d=3$, $k=2$, $\beta=1$ and $\delta=0.7$, $\delta_E=0.8$. There are 7 nodes $n_1 \ldots n_7$ in the network. We can follow the example with the help of Figure 1.

a) Local key exchange: First, nodes generate x-packets, say $N=10$ of them $(x_1^1, x_2^1, \ldots, x_{10}^1) \ldots (x_1^7, x_2^7, \ldots, x_{10}^7)$, and send them to their neighbors through the erasure channel. Let us look at node n_3. It expected to receive 3 x-packets both from n_2 and n_4, say (x_3^2, x_5^2, x_{10}^2) and (x_1^4, x_3^4, x_7^4). As $k=2$, one more step is carried out over the erasure channel. Node n_3 produces xor-ed packets from the received x-packets: $(x_3^2 \oplus x_1^4, x_5^2 \oplus x_3^4, x_{10}^2 \oplus x_7^4)$ and transmits them. From these, n_2 and n_4 receive one, e.g. $x_3^2 \oplus x_1^4$ and $x_5^2 \oplus x_3^4$ respectively. Knowing x_3^2 and x_3^4 they both can decode the unknown packets. Every node does the same, so by the end of this step, nodes have 10 x-packets of their own, 2×3 x-packets from their one-hop neighbors and 2×1 form their 2-hop neighbors.

Let us look at node n_3 and the packets generated by this node. It can generate $N\delta_E(1-\delta^2) \approx 4$ y-packets out of which their one-hop neighbors n_2 and n_4 may know 3 and their two-hop neighbors n_1 and n_5 1 – the ones generated from x-packets they correctly received. So, it needs to send 1 z-packet to make its one-hop neighbors know all y-packets. Meanwhile, n_3 also receives the z-packet of n_2 and n_4 and reconstructs their y-packets. One more step is needed, n_3 needs to generate z-packets from the y-packets of n_2 and n_4 such that n_2 may receive the y-packets of n_4 and n_4 may receive the y-packets of n_2. To that end 2×3 z-packets are needed. These z-packets are sent xor-ed, both neighbors can decode them and reconstruct the unknown y-packets. Now, n_3 generates 4 linear combinations of its y-packets as k-packets. The same k-packets are produced by n_1, n_2, n_4, n_6 as well.

Similarly, every node shares a set of common k-packets with its one- and two-hop neighbors.

b) Dissemination of local keys: Let us call k_i the set of k-packets generated by node i from its own y-packets. So, n_3 has k_1, k_2, k_3, k_4, k_5. It produces the xor-ed packets $k_1 \oplus k_5$ (element-wise) and transmits them. From this, node n_2 can decode k_5 and n_4 can decode k_1. Node n_3 receives $k_7 \oplus k_4$ and $k_2 \oplus k_6$. This way, it learns all k-packets and so does every other node as well.

c) Generating group keys: Every node has $7 \times 4 = 28$ k-packets, while Eve overheard overall 10 linear combinations of them (2 during the last step and 8 from the z-packets). So, it is possible to create a secure group key consisting of 18 linear combinations of k-packets.

V. Analysis

We investigate the performance of the scheme described in the previous section. We consider the achievable key size and the required number of transmissions separately.

Let us first look at the achievable key size. We have to consider the amount of information that Eve may have of the k-packets in the network. The y-packets are generated such that they are perfectly secure from the adversary, hence eavesdropped x-packets do not provide any useful information for the adversary. Now, consider the number a_i of z-packets that Eve learns during the local key exchange from its i-hop neighbor. The z-packets are transmitted reliably, hence Eve learns all of them that go through the observed link. For the i-hop neighbor, this equals

$$a_i = N\delta_E((1-\delta^2)-(1-\delta)^i)$$

overheard packets. This means that the ith hop neighbor of the adversary could create $N\delta_E(1-\delta^2)-a_i$ k-packets securely. In general, the knowledge of the adversary is equivalent to knowing $(()^+$ denotes the positive part of a number)

$$A_i = (N\delta_E\beta(1-\delta^2) - (N\delta_E(1-\delta^2) - a_i))^+ = \\ = N\delta_E(\beta(1-\delta^2) - (1-\delta)^i)^+ \quad (1)$$

of the k-packets generated by its i-hop neighbor. Thus, before the dissemination of the local keys the information that the adversary has of all k-packets in the network is equivalent to know $A = 2\sum_{i=1}^{k} A_i$ k-packets. Besides, during the dissemination of the local keys it learns

$$2(d-k)N\delta_E\beta(1-\delta^2)$$

further k-packets. Form this, the number of packets that can be securely generated to serve as a group key is

$$\mathcal{K} = (2d+1)N\delta_E\beta(1-\delta^2) - A - 2(d-k)N\delta_E\beta(1-\delta^2).$$

The achieved key size is already given, now we compute the number of required wireless transmissions. When disseminating x-packets, k rounds are carried out, and the ith round requires $N(1-\delta)^{i-1}$ transmissions. This is overall

$$N\frac{1-(1-\delta)^k}{\delta} \quad (2)$$

transmissions per node, or simply Nk, if $\delta = 0$.

Next, z-packets are disseminated, again in k rounds with $N\delta_E((1-\delta^2) - (1-\delta)^i)$ transmissions per node in the ith round. In this step

$$N\delta_E\left(k(1-\delta^2) - (1-\delta)\frac{1-(1-\delta)^k}{\delta}\right) \quad (3)$$

packets are transmitted per node, or 0, if $\delta = 0$.

In the last step local keys are sent performing $d-k$ steps of the broadcast protocol, that is

$$(d-k)N\delta_E\beta(1-\delta^2) \quad (4)$$

packets sent per node. The sum of (2), (3) and (4) results

$$\mathcal{C} = N\left(\delta_E(1-\delta^2)(d\beta + k(1-\beta)) + \\ + \frac{1-(1-\delta)^k}{\delta}(1-\delta_E(1-\delta))\right)$$

packets sent per node overall. The total number of transmissions in the network is $t = (2d+1)\mathcal{C}$, the cost of creating unit size secret key with this the scheme is thus

$$\frac{(2d+1)\mathcal{C}}{\mathcal{K}}. \quad (5)$$

A. Towards understanding the algorithm parameters

There are two parameter of the scheme that affect the performance of the key exchange. These are the number of hops $0 \le k \le d$ over which the dissemination of x-packets is performed and the local keys are established, and the size of the generated local keys β. Interestingly, under different settings of parameters δ and δ_E different values of k and β give the lowest cost.

Our first observation is that when either δ goes to 1 or δ_E goes to 0, then the achieved key size goes to zero while the associated cost remains non-zero whenever $k \ne 0$ due to (2). This means that for these values the cost of a secret key goes to infinity if $k \ne 0$, thus there is a certain region where $k = 0$ gives the best performance. Of course, this region includes the special case we analyzed, when $\delta_E = 0$. To be more specific, $k = 0$ gives the best performance – regardless of the value of β – when (we omit the detailed deduction here)

$$\delta_E \le \frac{1}{(2d+1)(1-\delta)}. \quad (6)$$

This result means that it is not worth exploiting the benefits of the erasure channel if the quality of the adversarial channel is over a certain threshold. It is interesting to note that the size of this region decreases with increasing the number of hops d in the network.

For other parameter values outside the region given by (6), the analysis becomes more complicated, mainly because of (1). Let us define the points where A_i become positive:

$$\beta_i = \frac{(1-\delta)^i}{1-\delta^2}.$$

Further, let $\beta_0 = 1$. We can see that within intervals $[\beta_j; \beta_{j-1}]$ both the achieved key size $\mathcal{K}$ and the required number of transmissions t are linear functions of β, if k is fixed. From this it follows, that the minimum value for (5) is given by a boundary of these intervals, i.e. the best performance is achieved with $\beta = \beta_j$ for some $0 \le j \le k$.

The assignment $\beta = \beta_j$ means that the adversary has no information of the local key generated from the y-packets of its $1 \ldots j$-hop neighbors, while has partial knowledge of local keys generated from y-packets of its $j+1 \ldots k$-hop neighbors. It is interesting to note that the further the adversary is located from the source, the more information Eve gains of the generated key. This is because of the z-packets, the further a node is from the source, the more z-packets it receives and so does the adversary.

To illustrate the various values that achieve the best performance with the scheme we numerically evaluated which values of k and β provide the lowest cost to establish a group key. We plotted these figures for a relatively large d, in particular for $d = 50$, to see a larger variety of these values. An example of different values of k and β are shown for this case in Figures 2 and 3 respectively. We can see that for β, the higher values dominate, only the 5 highest possible occurs. This means that in most cases, it is not worth creating perfectly secure local keys. On Figure 4 we can see the characteristic of the cost function for the special case when $\delta = \delta_E$. For this figure we set $d = 5$, which corresponds to 11 nodes in the network.

VI. Related work

The problem of achieving secrecy in the presence of an adversary exploiting noisy broadcast transmissions over the wireless channel was first considered in [5]. This schemes achieve secrecy if the adversarial channel is worse than the honest one. [6] showed that a public feedback to the sender

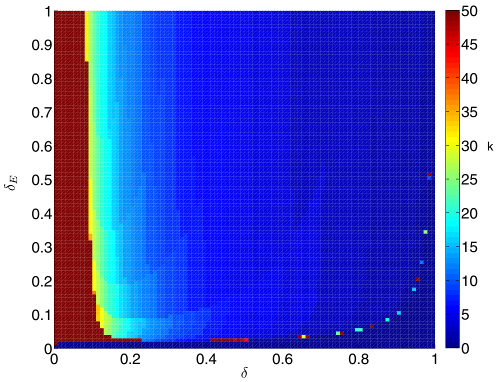

Figure 2. Values of k that give the lowest cost in a network with $d = 50$. The local key exchange is performed in the k-hop neighborhood of every node.

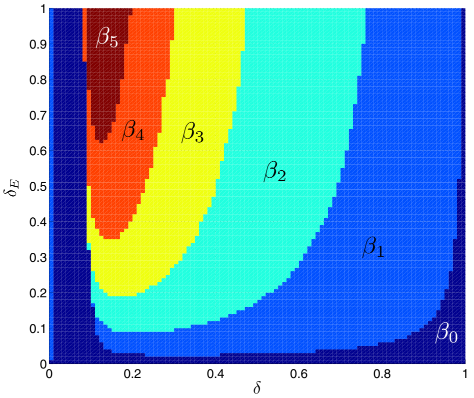

Figure 3. Values of β that give the lowest cost in a network with $d = 50$. For $i > j$, $\beta_i < \beta_j$. With $\beta = \beta_j$ k-packets are secure from Eve up to its jth neighbor.

increases significantly the achievable secrecy rate. With feedback available secrecy can be achieved even if the adversarial channel is better than the channel between honest nodes.

The achievable secrecy rate between multiple nodes that can use a public channel is given by [7]. The problem of secrecy when both a public channel and an erasure channel are available with multiple nodes in a one-hop network is investigated by [8], [9], [10], [1]. Both an upper bound and a computationally efficient scheme achieving the bound were presented in [1]. To the best of our knowledge, for multi-hop networks no bounds on achievable group secrecy are available.

VII. Conclusions and Discussion

We presented a group key exchange scheme for multi-hop wireless networks that achieves unconditionally secure group keys against an eavesdropping adversary. To that end we make use of the wireless medium both as an erasure and as a reliable channel. We analyzed the scheme in detail in a circular topology under different parameter settings. As a cost metric we used the total number of wireless transmissions in the network required to achieve a unit size group key.

We presented and analyzed our multi-hop key exchange protocol for a special circular topology, however we argue that it can be extended for more general topologies as well. We build our protocol from two components, one exploits the topology of the network and relies on a forwarding protocol, the other exploits the erasure channel and relies on another forwarding scheme that disseminates packets to the k-hop neighborhood over the erasure channel. Given such communication protocols, each step of our key exchange scheme can be generalized easily. In a sense, the topology-dependent part is encapsulated into the two communication protocols. Of course, the achieved key size and the actual performance highly depends on the current topology. Our future work includes designing such general key-exchange schemes, as well as investigating the optimality of our algorithms.

References

[1] S. Diggavi, C. Fragouli, M. Jafari Siavoshani, U. K. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," in *Asilomar Conference on Signals, Systems, and Computers*, 2010.

[2] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding," in *In Proceedings of NETCOD*, 2005.

[3] J.-Y. Le Boudec, C. Fragouli, and J. Widmer, "A network coding approach to energy efficient broadcasting: from theory to practice," in *Proceedings of INFOCOM*, 2006.

[4] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*. Foundations and Trends in Networking, 2007, vol. 2.

[5] A. D. Wyner, "The wire-tap channel," *The Bell system Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[6] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[7] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3047–3061, 2004.

[8] ——, "Secrecy capacities for multiterminal channels," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 2437–2452, 2008.

[9] A. A. Gohari and V. Anantharam, "Information-theoretic key agreement for multiple terminals – part I." *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3973–3996, 2010.

[10] ——, "Information-theoretic key agreement for multiple terminals – part II." *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3997–4010, 2010.

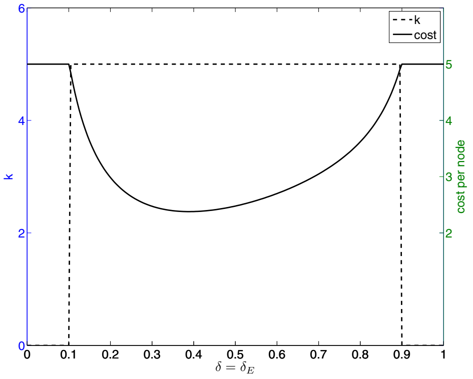

Figure 4. Cost and k for $d = 5$. $k = 0$ in the region defined by (6).