Mitigating Node Capture Attack in Random Key Distribution Schemes through Key Deletion

Mohammad Ehdaie[1], Nikos Alexiou[2], Mahmoud Ahmadian[3], Mohammad Reza Aref[4] and Panos Papadimitratos[2]
[1] *Parsa Sharif Research Center and K.N.Toosi University of Technology, Iran, ehdaie@parsasharif.ir*
[2] *School of Electrical Engineering, KTH, Sweden, {alexiou,papadim}@kth.se*
[3] *CCL, K.N.Toosi University of Technology, Iran, mahmoud@eetd.kntu.ac.ir*
[4] *ISSL, Sharif University of Technology, Iran, aref@sharif.edu*
Corresponding author: ehdaie@parsasharif.ir

***Abstract*- Random Key Distribution (RKD) schemes have been widely accepted to enable low-cost secure communications in Wireless Sensor Networks (WSNs). However, efficiency of secure link establishment comes with the risk of compromised communications between benign nodes by adversaries who physically capture sensor nodes. The challenge is to enhance resilience of WSN against node capture, while maintaining the flexibility and low cost features of RKD schemes. We address this problem by proposing an effective technique, namely KDel, which don't require any special-purpose hardware or expensive computations. We demonstrate that our approach significantly increases the resilience of RKD schemes against node capture at the cost of a little communications, while maintaining network connectivity at the same level. Moreover, our scheme is generally applicable as it can improve the resilience of any RKD scheme.**

***Index Terms*-** Key Deletion, Node Capture, Random Key Distribution, Wireless Sensor Networks.

I. INTRODUCTION

A large number of Random Key Distribution (RKD) schemes have been proposed in the literature to secure Wireless Sensor Network (WSN) applications. Some of RKD schemes are surveyed in [1] and [2]. However, sensor nodes are exposed to physical compromise by adversaries, who target the keys stored at each node. With the stolen keys in their possession, the adversaries are then able to compromise communication links between benign nodes (e.g., see [3]). Thus, the following challenge arises: how to increase resilience of RKD schemes to WSN node capture, while maintaining the connectivity level, the flexibility and low cost features of RKD?

To address this challenge, we introduce *KDel*, a key erasure method after the sensor network deployment. We term this: each node discards the keys not used to establish secure links with its

Manuscript received 15-Feb., 2017 and revised 5-Oct.-2017,
Accepted on 12- Dec.-2017
P- ISSN: 2322-4088
E- ISSN: 2322-3936

neighbors. We study its effects, the problems that it may create as well as the solution. We show that KDel does not increase per-node storage, nor introduces significant additional computation or communication overhead. Moreover, we demonstrate that our method can significantly improve resilience against node capture attacks and thus, the overall security of RKD schemes. Using KDel, the chance of adversary to compromise links between benign nodes given that he/she captures some nodes is decreased. The obtained improvement depends on the parameters of the system.

The remainder of this paper is organized as follows: Sec. 2 defines the adversary model and the problem at hand. In Sec. 3 we present our KDel method and its improved resilience on RKD schemes. Finally, in Sec. 4 we conclude the results.

II. SYSTEM MODEL & PROBLEM STATEMENT

Assume a WSN with N nodes and L links. Each node is assigned a set of m keys called the key ring, which are randomly selected from a large pool of keys with size $|P|$. Sensor nodes establish secure communication links based on the RKD scheme that is used. In order to discover the possibly common keys in their key ring, sensors are running a Key Discovery Protocol (KDP) [4]. In the basic RKD scheme [4], every pair of sensors that discovers at least one common key can establish a secure link. In the q-composite scheme [5], the security increases as sensors have to discover at least q common keys to establish a secure channel.

We consider an adversary that randomly captures s nodes and thus is able to obtain the m stored keys per captured sensor. The aim of the adversary is to compromise communication links between benign nodes (notably other than the compromised ones, which are trivially compromised). A communication link is compromised if the key (or the keys) used to secure the link is/are included in the key ring of any of the captured nodes. Finally, we quantify the adversarial gain in terms of broken communication links given s captured nodes, using a fail function per key distribution scheme [4].

$$Fail(s) = \frac{\#\text{comp. links given s captured nodes}}{\#\text{all network links}} \quad (1)$$

Our objective is to maintain the flexibility of RKD schemes at the same level, while increasing the resilience. The intuition is for the RKD schemes to achieve desired connectivity (in terms of security associations), they require each node to carry a relatively large key ring. However, only a small fraction of the pre-installed sensor keys are finally used to establish secure links, and the remainder of the keys remains *unused*. The problem is that, when a node is captured each of the compromised keys, whether used by the sensor or not, has the utility for the attacker; that is it has the same usefulness to compromise a secure link in the network. Considering this, we propose KDel, a novel mechanism to reduce the chance of the adversary to break secure communication links given captured sensors.

Notation: We summarize the notations we use throughout the paper here:

N: Number of nodes in the network.

n: Average number of neighbors (for every node)

P: Key pool.

$|P|$: Key pool size.

m: Key ring size in normal case (prior to key deletion).

m': Key ring size after key deletion.

β: Fraction of undeleted keys.

p_c: Probability that two nodes can establish a secure connection.

p_c': Probability that two nodes can establish a secure connection, after replacement of some nodes in the key deletion scheme.

γ: Fraction of replaced nodes in the network.

L: Number of primary links in the network.

L': Number of secondary links (links established via a Path Key Establishment protocol).

s: Number of captured nodes in the network.

$p(i)$: Probability of two nodes sharing exactly i keys

q: The minimum number of shared keys for two nodes to have a secure link (q-composite scheme).

III. KEY DELETION TECHNIQUE

Given that the left unused keys in each node key ring can be a vulnerability, we propose to *delete the unused keys* and significantly decrease the chances of an adversary to compromise additional links in the network. We show that, if we delete all the keys except a fraction of β, the adversary's chance is approximately decreased by β. This method is different with pre-loading the sensors with less keys, since pre-loading with less keys decrease the chance of sensors to establish secure links with their neighbors while our method does not affect the connectivity level of nodes.

A. Key Deletion: The Basic Scheme

We use the formulations in [4], [5] and [6] to analyze the new scheme. We denote the primary probability of finding a shared key by p_c. For the basic scheme, if we set the values for $|P|$ and m, the probability that two nodes can establish a secure link can be written as: $p_c = p(1) + p(2) + \cdots + p(m)$; where $p(i)$ is the probability that two nodes have exactly i keys in common. $p(i)$ is equal to:

$$p(i) = \frac{\binom{m}{i} \times \binom{|P|-m}{m-i}}{\binom{|P|}{m}} \quad (2)$$

We use the Fail function to measure resilience against node capture. The details for the calculating the Fail function is given in [5]. For the basic scheme, the Fail function is written as:

$$Fail_{basic}(s) = 1 - (1 - \frac{m}{|P|})^s \quad (3)$$

By applying the KDel technique, we erase some of the unused keys of the nodes after the KDP terminates and thus, only m' keys finally remain in each sensor. The problem is that when we replace some of the nodes in the network (due to their limited life time), the connectivity between new nodes and old nodes is affected. If a new node is introduced to the network, the probability to establish a secure link with its neighbors decreases to: $p'_c = p'(1) + p'(2) + \cdots + p'(m')$; where $p'(i)$ is the probability that a new node (with m keys) and a previously established node (with only m' keys) have exactly i keys in common and is equal to:

$$p'(i) = \frac{\binom{m'}{i} \times \binom{|P|-m'}{m-i}}{\binom{|P|}{m}} \quad (4)$$

Despite the decrease in connectivity for new nodes, KDel offer a great advantage. If the adversary captures a node, only m' keys are now obtained, compared to m keys that would be captured without KDel. Thus, according to Eq.3, the adversary's chance to compromise new secure links between benign nodes when s sensors are captured is reduced to:

$$Fail_{basic,KDel}(s) = 1 - (1 - \frac{m'}{|P|})^s \quad (5)$$

If we set $m' = \beta \times m$ for a given β factor, i.e. erasing all the keys except a fraction of β, we have:

$$Fail_{basic,KDel}(s) = 1 - (1 - \beta \times \frac{m}{|P|})^s \simeq \beta \times (1 - (1 - \frac{m}{|P|})^s) \quad (6)$$

The last part is obtained according to the binomial approximation while $m \ll |P|$ and number of captured nodes is reasonably low. This formula shows that applying KDel on the top of the basic RKD scheme yields to an improvement in resilience by a factor of $1 - \beta$ (the Fail function is decreased by β).

Consider a typical WSN setting with a key pool of size $|P| = 20{,}000$ and key rings of size $m = 100$ keys. Given these specifications and Eq. 2, a pair of nodes can establish a secure channel with probability $p_c = 0.4$. The new size of the key ring after KDel is $m' = 50$. Fig.1 shows the fraction of compromised links for a given number of captured nodes, for the basic RKD scheme without and with KDel (Note: we used Eq.5 to plot the figure, not the approximation part of Eq.6). We observe that KDel greatly improves the resilience of the basic scheme against node capture attacks. As the percentage of deleted keys increases, the resilience of the basic scheme against the attack also increases. For the case of deleting half the key ring, the probability of successful link compromise is reduced in half because $\beta = \frac{m'}{m} = 0.5$.

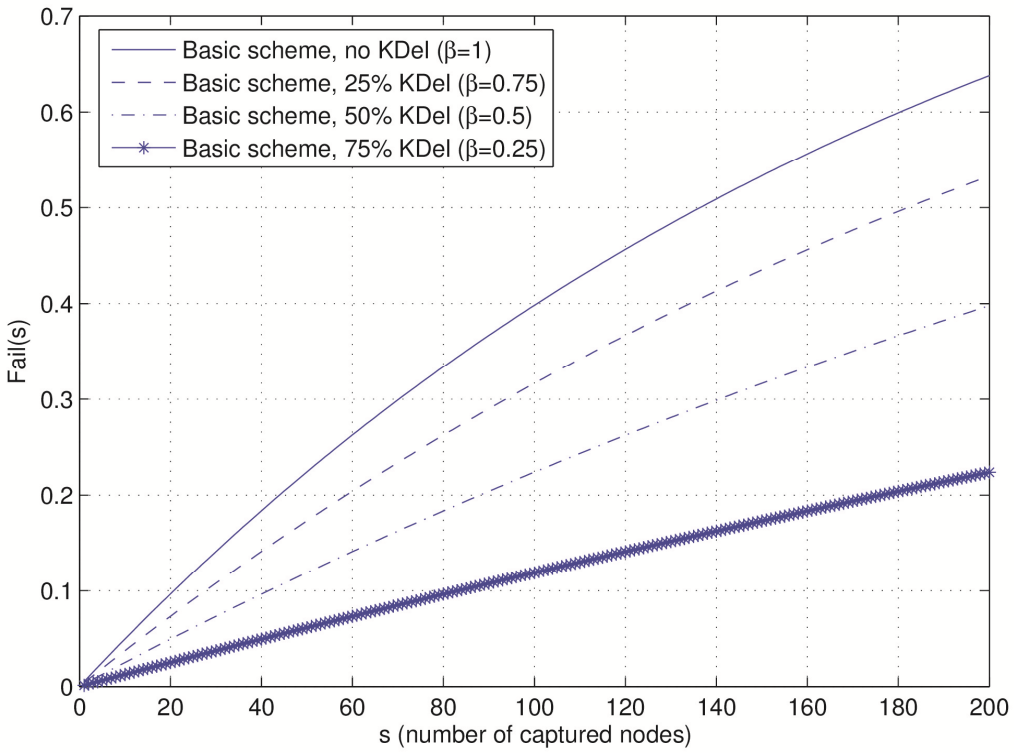

Fig. 1. Fail function for the basic scheme, with and without Key Deletion

On the other hand, increasing the β yields to a reduction in connectivity. Since the connectivity needed is a function of the topology and the application, we can choose a proper value for β that keeps the connectivity at the desired level while improving the resilience. However, in Subsection 3.3, we propose a solution to compensate the connectivity reduction.

B. Key Deletion: q-Composite Scheme

We extend the previous analysis for the q-composite RKD scheme. The probability that two nodes can establish a secure link becomes: $p_c = p(q) + p(q+1) + \cdots + p(m)$; where $p(i)$ is the same as Eq. 2.

Since the adversary has to get all the q keys used for a specific link to compromise it, the probability to compromise a link with exactly i keys (with $i \geq q$) becomes:

$$(1 - (1 - \frac{m}{|P|})^s)^i \quad (7)$$

Hence, the Fail function is:

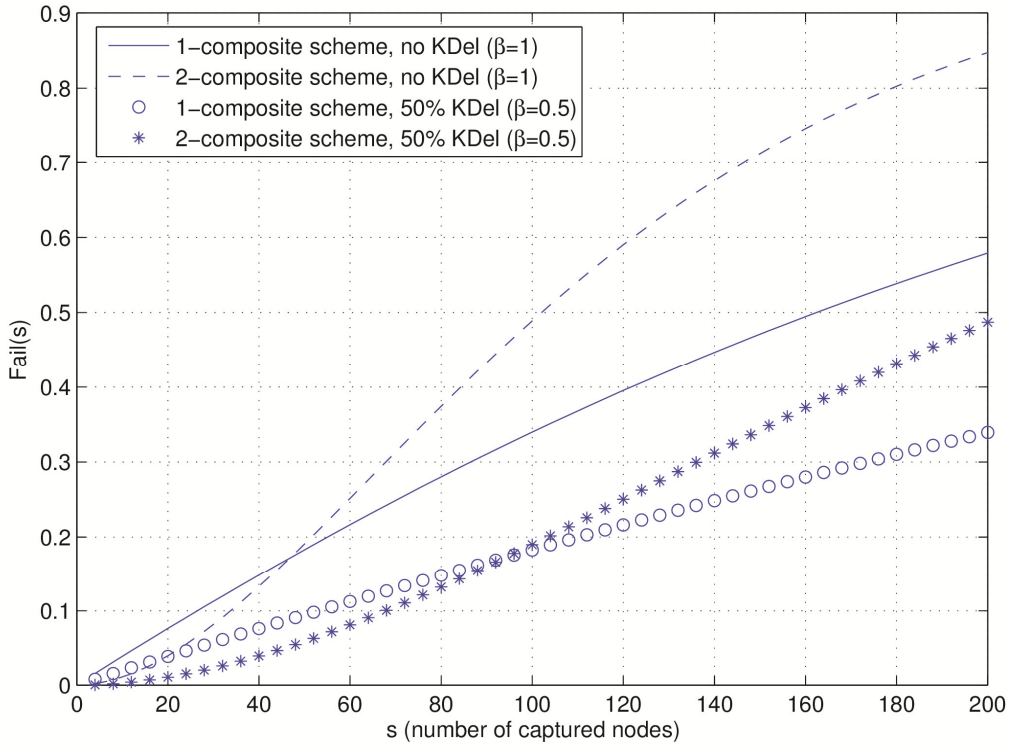

Fig. 2. Fail function for the 1-composite and 2-composite schemes with and without Key Deletion

$$Fail_{q-comp}(s) = \sum_{i=q}^{m} \left(1 - \left(1 - \frac{m}{|P|}\right)^{s}\right)^{i} \times \frac{p(i)}{p_c} \tag{8}$$

Using KDel, the Fail function can be rewritten as:

$$\begin{aligned} Fail_{q-comp,KDel}(s) &= \sum_{i=q}^{m} \left(1 - \left(1 - \beta \times \frac{m}{|P|}\right)^{s}\right)^{i} \times \frac{p(i)}{p_c} \simeq \\ &\sum_{i=q}^{m} \beta^{i} \times \left(1 - \left(1 - \frac{m}{|P|}\right)^{s}\right)^{i} \times \frac{p(i)}{p_c} \\ &\leq \beta^{q} \times \sum_{i=q}^{m} \left(1 - \left(1 - \frac{m}{|P|}\right)^{s}\right)^{i} \times \frac{p(i)}{p_c} \end{aligned} \tag{9}$$

Eq. 9 shows that in the best case, the improvement in the resilience becomes $1 - \beta^{q}$. In the other words, the KDel has a better effect on the q-composite than the basic scheme.

For numerical illustration, consider the same setup with the basic scheme ($m = 100$, $m' = 50$, $p_c = 0.4$) and assume that we apply KDel on the 1-composite scheme and the 2-composite schemes ($q = 1$ and $q = 2$ respectively). Fig. 2 presents the comparison between the schemes, before and after KDel. Again, this figure is plotted according to the non-approximation part of Eq.9. We observe that in the best case, the effect of KDel on 1-composite scheme is about $\beta = 0.5$, while for the 2-composite scheme, by applying KDel, the fail funtion is reduced by a factor of $\beta^{q} = (0.5)^{2} = 0.25$.

C. Introducing new nodes to the WSN

On the flip side of the coin, KDel creates a problem when new nodes need to be introduced to the network. Since the lifetime of WSNs nodes is limited, replacing a fraction of the old nodes with new ones (new sensors with new key rings) is expected. However, for the case of sensors that deleted their unused set of keys, the probability that a new node can discover a new communicating peer decreases. To overcome this problem, we suggest the PKE (Path Key Establishment) protocol, as defined in [4]. PKE is a method to establish a secure path between two nodes with no shared keys, through a common neighbor. Assume that a new node introduced to the network is running the KDP with its neighbors. The new node may discover that it shares enough keys to establish a secure channel with $n \times p_c'$ nodes. However, according to the specifications of RKD schemes, each node should be able to establish a secure channel with $n \times p_c$ nodes. To achieve this, the new sensor can follow the PKE protocol to discover $n \times (p_c - p_c')$ extra nodes that it shares common keys with. By following PKE, the sensor picks $n \times (p_c - p_c')$ of its neighbors, and establishes a secure channel with a node that its neighbor is communicating with. In other words, communicating peers are used as a stepping stone to establish communication links with sensors that no keys are pre-shared. The new node has to generate a key and forward it through the common neighbor to the intended recipient. In Appendix, we show that for typical WSN parameters (they are according to references such as [4] and [5], or they are experimental in some cases), pairs of new and existing nodes can establish connections via common neighbors with a high probability.

D. Effect of Attack on Path Key Establishment Protocol

We now consider an attack against the WSN employing KDel, and after the replacement of some old nodes with new ones. If an adversary captures the *common neighbor*, the new established link will subsequently be compromised. While it is possible to counteract this attack by some methods such as multi-path key reinforcement [5], we study this scenario and show that it has no significant impact on our method. Even if we consider such an attack, the percentage of compromised links with KDel is much better than the percentage of compromised links without KDel. The intuitive for this is that the secondary links can be compromised just by capturing the common neighbor node, while for the primary links, there are several nodes throughout the network that their capturing leads to compromising the link. Therefore, compromising the links by such attack is less probable than normal case. We show the analysis in this part.

Consider a network that runs the basic scheme and assume that we replace a fraction of nodes, say γ, with new nodes. We now have $\gamma \times N$ new nodes and $(1-\gamma) \times N$ old nodes. The connectivity between new nodes and old nodes is p_c'. Using the PKE idea, we create *secondary links* between new nodes and old nodes to increase the connectivity from p_c' to p_c (to increase the degree of every new

node from $n \times p'_c$ to $n \times p_c$). Therefore, we should create $(p_c - p'_c) \times n$ secondary links for each new node, i.e. $\gamma \times N \times (p_c - p'_c) \times n$ secondary links in total. Use L' to denote the total number of secondary links:

$$L' = \gamma \times N \times (p_c - p'_c) \times n \tag{10}$$

Each secondary link uses an old node as a medium (common neighbor) to share a key with the desired node. Thus, each old node, on average, should store $\frac{\gamma \times N \times (p_c - p'_c) \times n}{(1-\gamma) \times N}$ *secondary keys* to reach the same number of achieved links between the nodes when KDel is not applied. When s nodes are compromised (averagely $\gamma \times s$ new nodes and $(1-\gamma) \times s$ old nodes), the adversary can break some primary links as well as some secondary links. The number of compromised primary links is:

$$f_1 = (1 - (1 - m'/|P|)^s) \times L \tag{11}$$

where L denotes the number of primary links. Besides, the number of compromised secondary links is:

$$f_2 = (1-\gamma) \times s \times \frac{(\gamma \times N \times (p_c - p'_c) \times n}{(1-\gamma) \times N} = \frac{s}{N} \times L' \tag{12}$$

i.e. *number of compromised old nodes* multiplied by *number of secondary keys per old node*. Hence, we have:

$$fail(s) = \frac{(f1+f2)}{(L+L')} \tag{13}$$

The result is shown in Fig. 3. Even with considering such attack, since the total number of links increases, the fraction of compromised links decreases. It is clear, because, the chance of compromising secondary links is lower than the chance of compromising primary links. Every secondary link has a key that is stored on only one other sensor (the common neighbor); while, every primary link has a key that is stored on $N \times \frac{m}{|P|} \gg 1$ nodes.

E. The overhead

The resilience improvement costs only a few numbers of extra communications for new nodes to share keys with existing nodes by PKE. For every new node, $n \times (p_c - p'_c)$ keys should be shared by PKE, each of them needs two communications (one from the new node to the common neighbor and one from the common neighbor to the desired old node). So, the number of communications for every new node is:

$$2 \times n \times (p_c - p'_c) \tag{14}$$

F. Comparison and Discussion

KDel can be applied to any currently available RKD scheme. In this sense, KDel is not comparable

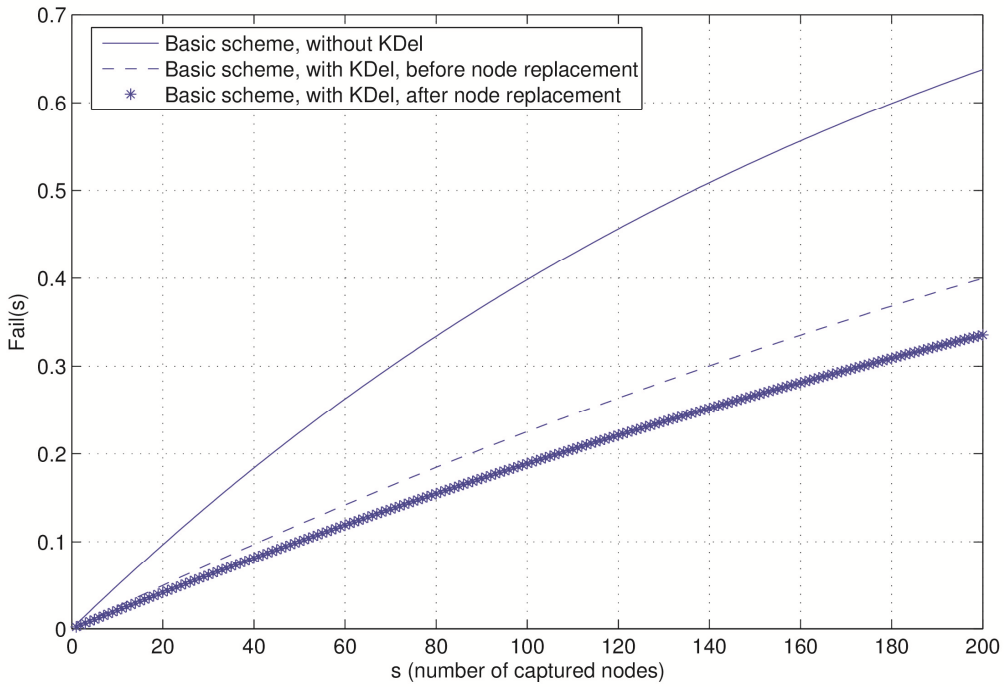

Fig. 3. Fail function for the basic scheme, before and after replacement of some nodes

to any other scheme. In other words, KDel is a technique, not a scheme, which can be added to any RKD scheme as a top layer and improves its resilience against node capture attack.

However, if we want to compare it with other schemes that aim to improve resilience of basic RKD scheme, we can note that KDel does not inroduce any significant overhead. There are several examples

in the literature that increasing the resilience yields to communication or computational overhead. For instance, q-composite improves the basic scheme at the cost of connectivity reduction. In addition to q-composite, the authors of [5] propose two other schemes, one of them needs more storage and the other needs more communications between nodes. Moreover, the scheme in [7], which uses a pool of polynomials to distribute keys between nodes, requires more computations to achieve a shared key. Also, [8] that employs a one-way function to create a hash chain of keys as its key pool, leads to computational overhead, too.

IV. CONCLUSION

We considered the problem of increasing the resilience of RKD schemes to node capture. We proposed a new technique, entitled Key Deletion, which can significantly increase the resilience of

RKD schemes. This can be applied to any of existing schemes without increasing the memory storage or mitigating the connectivity level. Finally, we have shown that all these benefits come at a few extra communications for the sensors.

APPENDIX

Number of common neighbors with secure links:

The number of common neighbors of two nodes is about $0.58 \times n$ [5], where n denotes the number of neighbors for a single node. In KDel, if we consider only the old nodes as common neighbors, the number of common neighbors between a new node and an existing node is $0.58 \times (1-\gamma) \times n$. So, the number of common neighbors with secure links is $0.58 \times (1-\gamma) \times n \times p_c \times p_c'$, since p_c is the probability that the existing node has a secure link with that common neighbor and p_c' is the probability that the new node has a secure link with that common neighbor.

If we assume a typical setting, $n = 40$, $p_c = 0.5$, $p_c' = 0.25$ and replace $\gamma = 20\%$ of nodes, for every new node there are averagely more than 2 common neighbors with secure links.

REFERENCES

[1] A.S.K. Pathan, "Security of Self-Organizing Networks: MANET, WSN, WMN, VANET," CRC Press, 2016.

[2] SS Iyengar, RR Brooks, "Distributed sensor networks: sensor networking and applications, CRC Press, 2016.

[3] P. Papadimitratos and J. Deng, "Stealthy pre-attacks against random key pre-distribution security," in Proceedings of the IEEE International Conference on Communications - Communication and Information Systems Security Symposium (ICC'12 CISS), (Ottawa, Canada), pp. 251-260, 2012.

[4] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the 9th ACM conference on Computer and Communications Security (CCS), pp. 41–47, 2002.

[5] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proceedings of the 2003 IEEE Symposium on Security and Privacy, (Washington, DC, USA), pp. 197–213, 2003.

[6] M. Ehdaie, N. Alexiou, M. Ahmadian, M. Aref, and P. Papadimitratos, "Key splitting for random key distribution schemes," in Proceedings of the 7th workshop on Secure Network Protocols (NPSec), 2012.

[7] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks." ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 1, pp. 41-77, 2005.

[8] J. Kur, V. Matyas, and P. Svenda, "Two improvements of random key predistribution for wireless sensor networks," in Proceedings of the International Conference on Security and Privacy in Communication Networks, 2012.