Key Splitting for Random Key Distribution Schemes

Mohammad Ehdaie *, Nikos Alexiou ‡, Mahmoud Ahmadian *, Mohammad Reza Aref † and Panos Papadimitratos ‡
* CCL, K.N.Toosi University of Technology, Iran
mohammad@ehdaie.com, mahmoud@eetd.kntu.ac.ir
† ISSL, Sharif University of Technology, Iran
aref@sharif.edu
‡ School of Electrical Engineering, KTH, Sweden
{alexiou,papadim}@kth.se

***Abstract*—A large number of Wireless Sensor Network (WSN) security schemes have been proposed in the literature, relying primarily on symmetric key cryptography. To enable those, Random Key pre-Distribution (RKD) systems have been widely accepted. However, WSN nodes are vulnerable to physical compromise. Capturing one or more nodes operating with RKD would give the adversary keys to compromise communication of other benign nodes. Thus the challenge is to enhance resilience of WSN to node capture, while maintaining the flexibility and low-cost features of RKD. We address this problem, without any special-purpose hardware, proposing a new and simple idea: key splitting. Our scheme does not increase per-node storage, and computation and communication overheads, and it can increase connectivity. More important, it achieves a significant increase in resilience to compromise compared to the state of the art, notably when the adversary does not have overwhelming computational power.**

I. Introduction

Wireless Sensor Networks (WSNs) can serve numerous applications, including monitoring for industrial processes, environmental pollution, and agriculture, as well as environmental monitoring, forest fire detection, health care, traffic control and home automation. In principle, WSN nodes have limited computing and storage capabilities and operate on a stringent energy budget. At the same time, without appropriate security mechanisms, WSNs are vulnerable to a broad range of attacks.

This has been understood early and a gamut of security schemes have been proposed. Asymmetric key cryptography, while feasible [1], has high cost (computation, communication overhead, and thus power) [2] and its use is limited to infrequent operations (e.g., [3]) or it is precluded altogether. Given the resource constraints of WSN nodes, symmetric key cryptography is widely accepted to secure WSNs. Each key shared by a pair of nodes is used to secure their communication. Keys can be shared by neighboring (i.e., in direct communication) but also physically remote nodes and can be used to establish in both cases security associations and secure links and be used in different ways to secure the WSN protocols.

However, WSN nodes are low-cost small-footprint platforms. This makes it impossible to offer tamper-resistant features [4]. Moreover, they operate unattended [5]. As a result, nodes may be physically compromised: an adversary can extract their symmetric keys and then use them to attack the WSN protocols. Equipped with captured keys, adversaries masquerade legitimate nodes and inject arbitrary messages that are still authenticated.

The harm that can be done when a node is captured depends on the way keys are managed and used by WSN nodes. Each node is equipped with multiple symmetric keys. One option is to make a centralized assignment that prescribes exactly the keys each node can use for specific peers (other nodes) [6], [7]. In that case, if each key is used only for one link, i.e., one pair of nodes, a captured node can at most misbehave to its a priori associated peers.

On the other hand, a looser yet more flexible and scalable approach can be used: nodes are first given a randomly chosen set of keys (called the *key ring*) from a large pool, then they discover which symmetric keys (if any) they share with which peers, and finally use the resultant key to securely communicate with the related peer [8], [9]. The flip-side of broader applicability for those RKD schemes, e.g., compared to the aforementioned pair-wise schemes [6], [7], is higher vulnerability to node capture. A compromised key ring allows attacks against the associated peers. More important, it includes keys that may be used by other nodes that had no association with the captured victim. This means that the adversary could compromise secure communication of other node pairs. Intuitively, the more numerous the captured nodes the more likely the compromise of other secure links in the network.

Given the advantages of RKD, is it possible to enhance its resilience to node capture? Can we do this without altering its salient features, reducing its efficiency, or without special purpose tamper-resistant hardware? This work addresses this problem: we propose a simple, yet as shown, a very effective solution we term *key splitting*. Our solution is applicable and can extend any RKD scheme.

In a nutshell, rather than using key rings we use rings of key parts, which can be put together to form entire keys. Rather than looking for matching keys, any two peers search for a sufficient number of parts; if they have them in common, they establish a secure link using a transformation of these parts as the shared key. The improvement in resilience is significant: as our analysis shows, the probability of link compromise we

achieve can be up to half of that for existing widely used schemes. At the same time, we maintain the same storage requirements: simply put, rather than keeping a ring of m keys, we keep a ring of $m \times z$ key parts each $1/z$-th of the key. Moreover, the connectivity, i.e., the likelihood two nodes be securely associated, can be enhanced.

In the rest of the paper, we first briefly define our system and adversary models and the problem at hand (Sec. II). We discuss related work in Sec. III and then we present our key splitting scheme in Sec. IV. We perform a detailed analysis of its resilience in Sec. V. We conclude with a discussion of future work.

II. Problem Statement & System Model

System Model. Each WSN node, V is assigned a set of m keys, $K_V = k_1...k_m$, termed the *key ring*. Keys are randomly chosen from a *key pool*, of size P, and they are stored at each node before its deployment. At any point in time, two sensors can run a Key Discovery Protocol (KDP) to discover their common keys, in order to then establish a secure communication link. Consider two sensors V_1 and V_2 that wish to communicate securely and run the KDP. In the basic RKD approach [8], only one common key k_{V_1,V_2} is required to establish secure connection. In a widely referenced variant of the basic scheme, the so-called q-composite RKD approach, V_1 and V_2 have to have (and discover) at least q common keys $k^1_{V_1,V_2}...k^q_{V_1,V_2}$ in order to establish a secure connection [9]. For the rest of the paper, we use both the basic and the q-composite RKD schemes.

Adversary Model. We assume an adversary, *Adv*, that can capture a number of wireless sensors, and thus construct a set S of compromised key rings, $S = \{K_{V_1}, \ldots, K_{V_i}\}$. A secure link between a pair of sensors can be compromised if the gamut of captured keys by *Adv* includes all those used to establish the link. If *Adv* wants to compromise the link between two nodes V_1 and V_2 using the basic scheme, k_{V_1,V_2} should be included in S. If V_1 and V_2 use the q-composite scheme, then all the $x \geq q$ keys shared between the nodes need to be stolen to compromise the link.

Adv does not have a priori knowledge of the keys that two nodes share. It can capture nodes at a rate of R_c nodes per time unit. Moreover, we assume that the adversary's computational power allows C_l cryptographic operations per time unit. We define C_l as the total number of operations brute-force check all possible 2^l inputs per time unit. We do not make any specific assumption on either parameter. Rather, we show in Sec. V how the relative power ("skills") of the adversary can affect the WSN resilience.

Problem Statement. An adversary *Adv* compromises a number of nodes and obtains their key rings. Links between benign nodes are vulnerable if their corresponding shared keys are included in the captured set of keys. As a result of the node compromise a fraction, α, of the WSN links between benign, non-compromised nodes is compromised, that is, *Adv* possesses the corresponding shared key(s). Or *Adv* knows for another fraction, β, of the WSN links, a part of the key(s) shared and used by the non-compromised nodes. In the latter case, *Adv* can attempt to compute the missing key(s) and compromise the link.

Overall, the challenge is how to enhance RKD resilience against node capture, in particular how to reduce link compromise. Given the above-defined adversary attacking over a period of time $[0, t]$, we assume that there is no reparative action by the WSN. Thus, $s = R_c \times t$ nodes are captured and their keys are compromised during that period. Then, assuming a set processing power, we want to minimize the total fraction of compromised links at time t. In particular, minimize the compromised link fraction, given s, $CL_s(t)$.

Notation:

- n: network size, in nodes.
- P: Key Pool Size.
- V: a benign node.
- K_{V_i}: V_i's key ring.
- m: size of the key ring.
- q: the minimum number of shared keys for two nodes to have a secure link.
- L: the length of a key in bits.
- k: an original key of full length.
- k_i: a split of the original key, of fewer length.
- p_c: probability that a pair of nodes share q keys (to have a secure link).
- $p(i)$: probability of two nodes sharing exactly i keys.

III. Related Work

An overview of the large palette of security problems in WSNs is given in [10], [11], [12]. Some of the most important security attacks are tampering of sensors and node capture [4], Denial of Service [13], secure routing and secure neighbour discovery [14], [15] and Sybil attacks [6], [16]. Security challenges of sensor networks are unique in nature, due to their limitations in storage, computation and communication capabilities. Traditional security approaches, such as public key cryptography, are therefore unsuitable for frequent usage [2]. Symmetric key approaches, relying on RKD schemes have been proposed instead, to overcome these limitations [8], [9], [7], [17].

Each sensor is equipped with a set of symmetric keys called the key ring, randomly chosen from a key pool. RKD schemes define the way that two sensors can establish a common symmetric key, using the overlaps in their key rings. To discover common keys, a KDP has to be run. The main RKD schemes proposed are the basic [8], the q-composite [9] and the random pairwise schemes [17], [7], [9]. The first two are briefly discussed in Sec. II. The random pairwise scheme associates a sensor identity with one key from the sensor's key ring and a comparison of the schemes with regard to node capture resilience is given below.

The starting point for many attacks is an adversary who compromises a sensor and obtains its key ring. A node replication attack is described in [18] a collusion attack against random pairwise schemes in [19]. Tamper-resistant devices can be used to protect against node capture. However, the increased

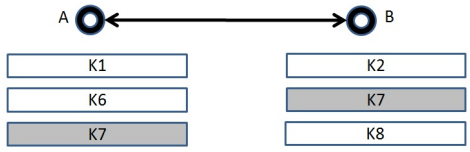

Fig. 1. RKD scheme for A and B that discover a shared key of size L.

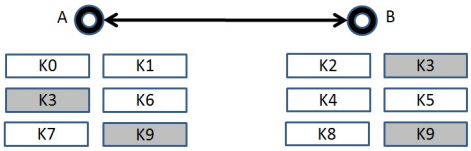

Fig. 2. Key splitting scheme for nodes A and B that discover two shared key parts of size $L/2$.

cost of tamper-resistant devices is a limiting parameter for extended usage of such devices [10].

Communication links between sensors may be compromised by adversaries who manage to capture the keys used to set-up the secure connection [9], [7], [17]. Having captured a number of nodes, the adversary can derive the secret keys used for communication between two sensors. RKD schemes play an important role in the resiliency against the attack. For a network of 10.000 randomly deployed sensors and $m = 200$ keys stored per sensor, an adversary has to capture 50 nodes to compromise 10% of the links [9]. For the same set-up, the 2-composite scheme has better resilience against the attack, and more than 70 nodes have to be captured to compromise the same percentage of links. The random pairwise scheme achieves increased resilience, since at most m links can be compromised from one captured node. Advanced stealthy strategies from colluding adversaries are studied in [20], where adversarial nodes combine their knowledge of captured keys to maximize their link compromise.

Despite the obvious advantages of the random pairwise scheme, it introduces restrictions in the maximum supported network size. The total size it can support can be computed as $n = m/p_c$, and is directly constrained by the sensors' limited memory [9], [7]. This limitation may render the random pairwise schemes unsuitable for a number of applications. Large and and flexible network deployments, as well as frequent addition of new nodes are better handled by the q-composite or the basic scheme. However, the security vulnerabilities of these schemes, bring out new challenges on how to make these schemes more resilient against attacks.

IV. The Key Splitting Scheme

The key splitting scheme is a method to increase the resilience against node capture attacks. We split each of the keys in P, in z equal $parts$, creating a new key pool of size $z \times P$.

Example: Basic Scheme. Nodes A and B with key rings of size m use the basic RKD scheme. Each key k has a length of L bits, and therefore each sensor needs $m \times L$ of storage for its key ring. Fig. 1 shows an instance of a KDP run between A and B. After completing the KDP protocol, the two nodes discover they share k_7. This key can then be used to establish a secure channel. Now consider the adversary *Adv*; if it was *"lucky"* enough to capture a node with k_7 stored in its memory, then it is only a matter of time and a few trials to compromise the link of A and B.

Example: Key Splitting Scheme. Consider again A and B; now each key in P is split in z equal parts. Each of the new key parts, will now have a length of L/z bits. With the same memory usage at each sensor, $z \times m$ keys can now be stored, creating a key ring of $z \times m$ keys. For the example, let $z = 2$, i.e., we split keys in two equal parts. A and B run the KDP to find common key parts. To preserve the same security level with a symmetric key of size L, we require that A and B should now discover *two* key parts of size $L/2$, instead of just one. Fig. 2 illustrates that. Having discovered k_3 and k_1, A and B can establish a symmetric key of size L as in the previous example.

Now they have a great advantage over *Adv* who tries to compromise their link. Instead of having to guess one key, as in the previous example, the adversary has to guess two key parts now. Having compromised the same number of nodes in both cases, *Adv* is less likely to hold both secret pieces in the second case than it is for one. Building on this observation, we analytically prove in Sec. V the effectiveness of the scheme, and we verify that network connectivity is not deteriorated.

The steps of the key splitting scheme are:

- **Split the keys:** Each key in P of size L is split in z equal parts of length L/z. The new key pool has a size of $z \times P$.
- **Assign the keys to sensors:** $m \times z$ keys are randomly assigned to each sensor.
- **Establish secure connections:** Sensors run the KDP in pairs and discover z keys in common, with $z = 2$ the simplest case.

V. Scheme Analysis

We analyze the resilience to node capture; for easier presentation, first analyze the case of a single node/key ring captured, and then the case of multiple key rings compromised. Finally, we analyze performance issues, notably the connectivity and memory storage. In brief, we find that key splitting increases resilience against node capture, as long as the adversary does not have overwhelming computational power.

We calculate the probability that an adversary possesses all of the key parts and compare this probability with the probability that adversary posses all of the keys in q-composite scheme as well as probability of link compromise in the basic RKD scheme. Then, we consider the case of a subset of the key parts for a link are compromised.

A. Security Analysis

The main metric of interest is the *resilience* against node capture. In particular, we are interested in compromised secure links, i.e., secure communication, between pairs of benign

nodes. This is expressed by the measure $Fail(s)|\mathcal{N} \rightarrow [0,1]$, where s is the number of compromised nodes or equivalently key rings. This is the fraction of compromised links of a WSN, given the adversary captured s key rings.

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

In what follows, unless mentioned otherwise, we analyze the resilience when keys are split in two pieces each. Moreover, we assume that the adversary knows the number of shared keys.[1]

Single Node Capture. Consider $Fail()$ with $s = 1$. Let a link V_1, V_2 in the network and V the captured node; we need to compute the probability of the event $A =$\{the link is compromised\}. Let B be the event $\{V = V_1$ or $V = V_2\}$, that is the captured node is one of the end-nodes of the considered link. $Fail(1)$ is equal to $Pr\{A\}$.

Single Node Capture: Basic scheme. We show that $Fail(1)$ is $\frac{2}{n} + \frac{m}{P}$. Bayes' rule gives us:

$$Fail(1) = Pr\{A\} = Pr\{A|B\}Pr\{B\} + Pr\{A|B'\}Pr\{B'\} \quad (2)$$

$Pr\{B\} = \frac{2}{n}$, as there are n nodes in the network and each link has two end nodes. Thus, $Pr\{B\prime\} = 1 - \frac{2}{n}$ Moreover, $Pr\{A|B\} = 1$, i.e., all secure links of the captured node are certainly compromised. Finally, for $Pr\{A|B'\}$: recall that when a node is captured, m keys are compromised. The considered link uses a key from the key pool. There are P options for this link, while m keys are compromised. So, the probability that the link is broken becomes $\frac{m}{P}$:

$$Pr\{A|B'\} = \frac{m}{P} \quad (3)$$

By substitution and assuming $n \rightarrow \infty$:

$$Fail(1) = 1 \times \frac{2}{n} + \frac{m}{P} \times (1 - \frac{2}{n}) \simeq \frac{2}{n} + \frac{m}{P} \quad (4)$$

Single Node Capture: Key Splitting scheme For the key splitting scheme, consider the case that each pair of nodes use only two half-part keys even if they have more. The fail function, as per Eq. (2), we also have $Pr\{B\} = \frac{2}{n}$ and $Pr\{B'\} = 1 - \frac{2}{n}$, and $Pr\{A|B\} = 1$. Now, for the $Pr\{A|B'\}$: note that when a node is captured, $2m$ key slices are compromised. The considered link uses two halves of keys from the key pool and it will be compromised if both of them are known to the adversary, i.e. $(2m/2P)^2 = (m/P)^2$. Note that if one half of the key is revealed and one remains secure (unknown to the adversary), we do not count such link as a compromised link. We discuss specifically the ramifications of the adversary having partial knowledge of the key splits further below. Thus, we get:

$$Pr\{A|B'\} = (\frac{m}{P})^2 \quad (5)$$

By substitution we have:

$$Fail(1) = 1 \times \frac{2}{n} + (\frac{m}{P})^2 \times (1 - \frac{2}{n}) \simeq \frac{2}{n} + (\frac{m}{P})^2 \quad (6)$$

[1]This is essentially a worst case for the scheme, or a best case for the adversary: it removes uncertainty for the adversary.

From Eq.(4) and Eq.(6), we see that the $Fail$ metric is much lower for the key splitting scheme compared to the basic RKD scheme. Note that the result can be even better in favor of key splitting if nodes use more than two key slices, as mentioned in the protocol. We show that in the next part.

Multiple Node Capture.

Multiple Node Capture: Basic scheme. Recall the probability from Eq.(3); thus, the probability to not reveal a key is $(1-\frac{m}{P})$. When s nodes are captured, the probability that a key is not revealed is: $(1-\frac{m}{P})^s$. As a result, the probability that a key, and consequently a link that uses the key, is comprised:

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

Multiple Node Capture: q-composite scheme [9]. For the q-composite scheme, assume that nodes share i keys (with $i \geq q$). $p(i)$ is the probability that two nodes have i keys in common, i.e.,

$$p(i) = \frac{\binom{m}{i} \times \binom{P-m}{m-i}}{\binom{P}{m}} \quad (8)$$

and p_c denotes the probability that two nodes can establish a secure link, i.e. $p_c = p(q) + p(q+1) + \cdots + p(m)$.

Then, with s compromised nodes, the probability of that link being compromised is: $(1-(1-\frac{m}{P})^s)^i$ Hence, the probability that any link between two benign (non-compromised) nodes is compromised is:

$$\sum_{i=q}^{m} (1 - (1 - \frac{m}{P})^s)^i \times \frac{p(i)}{p_c} \quad (9)$$

Multiple Node Capture: Key Splitting scheme. A link would be compromised if all of i key parts (splits) shared by the two incident nodes are known to the adversary. Let $p'(i)$ be the probability that the two nodes have i key parts in common:

$$p'(i) = \frac{\binom{2m}{i} \times \binom{2P-2m}{2m-i}}{\binom{2P}{2m}} \quad (10)$$

where p_c again denotes the probability that two nodes can establish a secure link, i.e. $p_c = p(2) + p(3) + \cdots + p(2m)$.

Then, the probability for a link to be compromised is:

$$\sum_{i=2}^{2m} (1 - (1 - \frac{m}{P})^s)^i \times \frac{p'(i)}{p_c} \quad (11)$$

In Fig. 3, we plot the probability of compromising a link when s nodes are captured for different schemes:

- Basic RKD scheme (Eq.(7)),
- q-composite scheme with $q = 1$ (Eq. (9) with $q = 1$)
- q-composite scheme with $q = 2$ (Eq. (9) with $q = 2$)
- Key splitting scheme (Eq. (11)).

We observe that our key splitting scheme achieves a lower probability of link compromise. For example, if 50 nodes in the network are captured, 29.1% and 24.2% of other links would be compromised with basic scheme and 2-composite scheme, respectively; while, in the key splitting, only 7.6% of links are compromised.

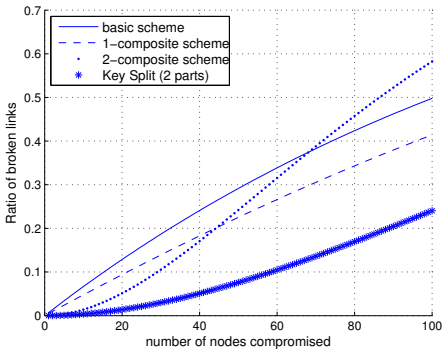

Fig. 3. Resilience against node capture; comparison.

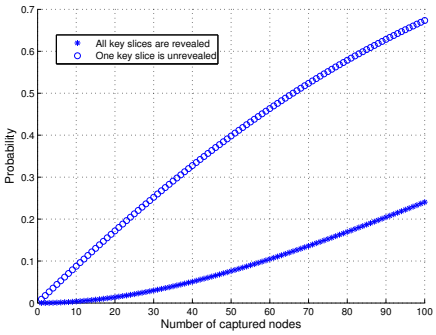

Fig. 4. Fraction of fully compromised links and fraction of links with information leakage.

In the Key Splitting scheme, it is probable that an adversary gets all half keys except one of them. In this case, we have an information leakage. But given that the key parts are inputs to a one-way hash function, f to produce the key, the adversary cannot obtain the link key. The only option would be to brute-force check all possible missing key parts. We calculate this probability, that out of i shared key parts one of them is missing and plot it in Fig. 4.

$$\sum_{i=2}^{2m} \binom{i}{i-1} (1-\frac{m}{P})^s (1-(1-\frac{m}{P})^s)^{i-1} \times \frac{p'(i)}{p_c} \qquad (12)$$

Computational compromise of links. We can see the trade-off of the key splitting when we consider the relative strengths of the adversary, notably R_c and C_l. In the case of two key parts, the brute force check would require strings of 64 bits long. Essentially, depending on how fast the adversary can do that, it can compromise additional links. With node capture rate R_c equal for all schemes, we add this latter factor to key splitting in Fig. 5.

We assume a network with 100,000 links and an adversary that can capture 1 node per time unit. We also assume really the worst case for our scheme, where the adversary is able to run 10 brute-force checks in a time unit, i.e. run 10×2^{64} computations per time. We plot the number of compromised links during the time. We observe that the curve for key splitting arises slightly only. Of course, this really depends

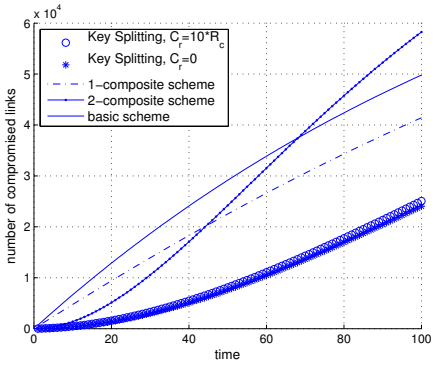

Fig. 5. Resilience against node capture and exploitation of information leakage; comparison.

on what the adversary is capable of. Still, key splitting raises the bar.

B. Performance Analysis

We consider next the connectivity achieved by our scheme in comparison with the basic and q-composite RKD schemes. Consider a key pool of size P and sensor nodes with memory m. The probability that two nodes have at least a common key in the basic scheme is:

$$p_{c,basic} = \sum_{i=1}^{m} p(i) \qquad (13)$$

where $p(i)$ denotes the probability that two nodes share exactly i keys (Eq. 8). For the q-composite scheme, the conditions are tighter. Two nodes can communicate securely if they have at least q common keys. Thus, the connectivity decreases to:

$$p_{c,q-comp} = \sum_{i=q}^{m} p(i) \qquad (14)$$

For key splitting, note that the key pool has $2P$ half keys and each WSN node stores $2m$ such key parts. Two nodes can establish a connection if they share at least two half keys:

$$p_{c,split} = \sum_{i=2}^{2m} p'(i) \qquad (15)$$

where $p'(i)$ is given in Eq. (10) above.

We plot p_c in Fig.6, for different values of m in a constant pool size (in bits) and compare the schemes. For the q-composite scheme, we consider the case $q = 2$ (For $q = 1$, its connectivity is the same as that of the basic scheme).

We observe that the key splitting scheme achieves much better connectivity than the q-composite scheme. Since there is a trade-off between connectivity and memory usage of this scheme, one may be interested in reducing the memory usage by remaining the connectivity constant. Besides, reducing the memory usage leads to an improvement in resiliency against node capture; because, the resiliency against node capture is related to the memory usage in a reverse manner. In comparison with the basic scheme, sometimes we get a better

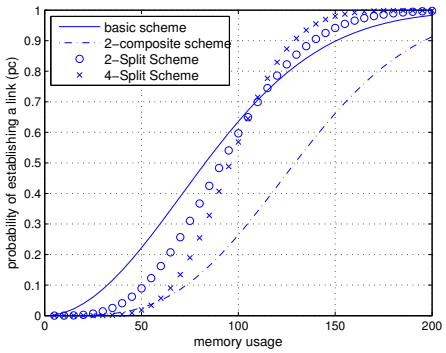

Fig. 6. Connectivity vs. Memory Usage

connectivity, while sometimes the connectivity is lower. We also consider the case that keys are split into four parts.

This figure shows that for example if we consider the memory usage $m = 140$ (140 complete keys, 280 half-part keys, 560 quarter-part keys), then the value of p_c will be 0.863, 0.9056 and 0.9549 for the three schemes, respectively. This means a 5% and 11% improvement in p_c in two versions of our scheme rather than the basic scheme. To address the importance of such improvement, consider as an example that we have a network with $n = 10,000$ nodes and we want to increase the probability that the network be connected from 0.99999 to 0.999999. Such increase needs an 11% improvement over p_c.

On the other hand, if we are interested in a connectivity that corresponds to $p_c = 0.9$, the amount of memory usage will be 151, 139 and 129, respectively; i.e. an 8% and 15% reduction in memory usage. Such reduction could be so important in memory constraint devices as sensor nodes, where a 4KB RAM memory is very typical. Also, remember that decreasing the memory usage yields in increasing the resiliency against node capture, since an adversary gets a lower number of keys by capturing a constant number of nodes.

VI. Conclusion and Future Works

We considered the problem of increasing the resilience of RKD schemes to node capture. We proposed a new scheme, key splitting, which can significantly increase resilience compared to existing schemes. This can be achieved as long as the adversary does not have overwhelming computational power, to brute-force links for which it has partial knowledge of the secret material. Essentially, key splitting raises the bar against a powerful and sophisticated adversary.

We focused on key splitting using two parts of key, with aggregate length equal to the original key length. We achieve a significantly lower fraction of compromised links esp. when the number of compromised nodes increases. In future work, we will analyze the general case, splitting keys in more than two parts. Then, we will analyze the cases of different fractions of the key material (number of parts) being known to the adversary, and investigate further the trade off and effect of node compromise vs. computational power of the adversary.

Acknowledgment

We would like to thank Iranian Research Institute for ICT for partially supporting this work (grant T/500/19241).

References

[1] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in *Proc. of the 7th IPSN*, 2008.

[2] K. Piotrowski, P. Langendoerfer, and S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," in *Proc. of the 4th ACM SASN*, 2006.

[3] P. Papadimitratos, J. Luo, and J.-P. Hubaux, "A Randomized Countermeasure Against Parasitic Adversaries in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1036–1045, September 2010.

[4] C. Hartung, J. Balasalle, and R. Han, "Node Compromise in Sensor Networks: The Need for Secure Systems," Tech. Rep. CU-CS-990-05, University of Colorado at Boulder, 2005.

[5] J. Bohli, P. Papadimitratos, D. Verardi, and D. Westhoff, "Resilient Data Aggregation for Unattended WSNs," in *IEEE LCN SenseApp*, pp. 994 –1002, October 2011.

[6] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd IEEE International Symposium on Information Processing in Sensor Networks*, pp. 259–268, 2004.

[7] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 42–51, 2003.

[8] 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.

[9] 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.

[10] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, pp. 53–57, June 2004.

[11] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 1–24, 2008.

[12] E. Shi and A. Perrig, "Designing secure sensor networks," *Wireless Communications, IEEE*, vol. 11, pp. 38 – 43, dec. 2004.

[13] A. D. Wood, J. A. Stankovic, A. D, and J. A, "Denial of service in sensor networks," in *Upper Saddle River*, Prentice Hall, Inc, 2002.

[14] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Towards provable secure neighbor discovery in wireless networks," in *ACM Workshop on Formal Methods in Security Engineering*, (Alexandria, VA, USA), pp. 31–42, October 2008.

[15] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," in *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, (Tokyo, Japan), pp. 189–200, March 2008.

[16] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, (London, UK), pp. 251–260, 2002.

[17] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and Communications Security (CCS)*, pp. 52–61, 2003.

[18] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Security and Privacy, 2005 IEEE Symposium on*, pp. 49 – 63, may 2005.

[19] T. Moore, "A collusion attack on pairwise key predistribution schemes for distributed sensor networks," in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pp. 5 pp. –255, March 2006.

[20] 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.