

# End-to-End Pairwise Key Establishment using Multi-path in Wireless Sensor Network

Hui Ling Taieb Znati

Department of Computer Science, University of Pittsburgh, Pittsburgh PA 15260

{hling, znati}@cs.pitt.edu

**Abstract**— Resource aware random key pre-distribution schemes have been proposed to overcome the limitations of energy constrained wireless sensor networks. In most of these schemes, each sensor node is loaded with a key ring. Neighboring nodes are considered to be connected through a secure link if they share a common key. Nodes which are not directly connected establish a secure path which is then used to negotiate a symmetric key. However, since different symmetric keys are used for different links along the secure path, each intermediate node must first decrypt the message received from the upstream node. Notice that during this process, the negotiated key will be revealed to each node along the secure path. The objective of this paper is to address this shortcoming. To this end, we propose an end-to-end pairwise key establishment scheme which uses a properly selected set of node-disjoint paths to securely negotiate symmetric keys between sensor nodes. We show through analysis that our scheme is highly secure against node captures in wireless sensor networks. The proposed scheme can be combined with any existing key pre-distribution scheme to enhance the security of its path-key establishment procedure.

## I. INTRODUCTION

A Wireless Sensor Network (WSN) is a collection of sensor nodes deployed to support several types of applications, including environmental monitoring, crisis management, and military sensing and tracking. Sensor nodes in WSN do not rely on any pre-deployed infrastructure and are usually resource limited. Depending on the type of the deployed WSN and the nature of the applications it supports, different levels of security may be required. While some WSNs do not require a high level of security, the proper functioning of other WSNs depends on their ability to carry traffic securely between sensor nodes. If sensors are deployed in a hostile battlefield, for example, the safety of the soldiers depends strongly upon secure data exchange between communicating entities. The same holds true when WSNs are deployed for critical infrastructure protection.

Asymmetric (public key) cryptography has been widely used to secure communication in wired networks. However, power constraints, coupled with the limited communication and computation capabilities of the sensor nodes, make RSA-based encryption impractical in WSNs. Trusted server schemes, such as Kerberos based schemes are not practical in WSNs, as these schemes depend upon a trusted third party which is not always available in WSNs. To address these limitations, key pre-distribution schemes, where key information is distributed to sensor nodes before they are deployed, have been proposed as a viable alternative to secure information dissemination in WSNs.

An intuitive solution to key pre-distribution, in WSNs with  $n$  nodes, is to store in each sensor node  $n - 1$  pair-wise keys. For a given node, each stored key is exclusively shared with one of the  $n - 1$  remaining sensors. This scheme has potential to achieve perfect security assuming that all keys are

unique. However, it imposes large memory requirement on sensor nodes when  $n$  is large. To overcome this shortcoming, a random key pre-distribution has been proposed [4]. Based on this scheme, each node is loaded with a set of  $m$  keys randomly selected from a large pool of keys,  $P$ , before deployment. Two nodes exchange either key identifiers or challenges to discover common keys in their key rings. The common key is used to establish a secure communication link. Since only a small number of keys are loaded in each sensor, node pairs may not always share common keys. Nodes without a common key to other WSN nodes are required to negotiate symmetric keys through a secure path.

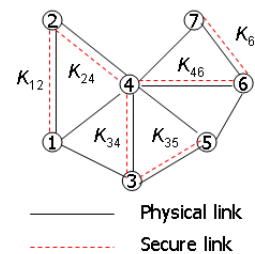


Fig. 1. An example sensor network after shared key discovery.

To illustrate the key pre-distribution process, consider the network depicted in Fig 1. In this network, as a result of the common key discovery phase,  $N1$  shares a key with  $N2$  but not with  $N3$  or  $N4$ . Consequently, to communicate with  $N3$ ,  $N1$  establishes a secure path to  $N3$ , e.g.,  $N1 \rightarrow N2 \rightarrow N4 \rightarrow N3$ , and sends a key  $K$  to  $N3$  through the secure path. As it travels from  $N1$  to  $N3$ ,  $K$  is encrypted with  $K_{12}$ ,  $K_{24}$  and  $K_{34}$ , respectively. Notice, however, that while the pair-wise key  $K$  is supposed to be exclusively shared between  $N1$  and  $N3$ , the need for successive encryptions and decryptions along the path causes the key to be exposed to the intermediate nodes  $N2$  and  $N4$ . This may lead to potential security compromise if a node along the path is captured. We refer to this problem as the “per-hop key exposure problem”. The likelihood of security breaches caused by key exposure is not negligible when random key pre-distribution is used to establish secure channels between a large number of WSN nodes. This is due to the fact that in key pre-distribution, the likelihood of a given node sharing a common key with a large number of other nodes is relatively small. Therefore, achieving secure communication between nodes which do not share a common key may lead to the establishment of a large number of secure paths for symmetric key negotiation and selection, thereby increasing the likelihood of security breaches.

In this paper, we propose an End-to-End Pairwise Key Establishment scheme to improve the security of path-key establishment. The scheme uses multiple node-disjoint paths to se-

cure the negotiation and selection of symmetric keys between non-neighbor nodes. Based on these paths, the pair-wise key,  $K$ , is divided into multiple fragments, each of which is transmitted along one of the established paths. All fragments are required in order to rebuild the key  $K$ . Consequently, the attacker will have to compromise all node-disjoint paths in order to capture  $K$ . The analysis presented in this paper shows that the security of the proposed scheme is improved. Furthermore, the proposed scheme could be integrated with any existing random key pre-distribution scheme to enhance its security.

The rest of paper is organized as follows: Section II introduces related work. Section III presents End-to-End Pairwise Key Establishment Scheme and security analysis. Section IV concludes this paper.

## II. RELATED WORK

The Random Key Pre-distribution scheme has been extensively investigated after being proposed by Eschenauer and Gligor [4]. Different methods of key generation and distribution have been proposed [1] [3] [2] [7]. A  $q$ -composite random key pre-distribution scheme has been proposed which increases the security of key set-up such that an attacker has to capture a large number of nodes to compromise a communication, with high probability [1]. The authors also propose a Multipath Key Reinforcement scheme to update an existing link key to a unique key, thereby ensuring that the key is not used by any other sensor nodes. Although the scheme proposed in this paper uses multiple paths, it differs from the one proposed in [1] in that we address a completely different problem and propose a detailed analysis of the level of security achieved using a node-disjoint path set. Furthermore, the proposed scheme goes beyond a 2-hop multipath.

In [2] a random key pre-distribution scheme is proposed. The scheme is based on deployment knowledge. With such knowledge, each node only needs to carry a fraction of the keys required by [4] while achieving the same level of security. It has the potential to reduce memory usage and improve the network's resilience against node compromise. However, the scheme relies on prior knowledge of node deployment. At times that information may not be available. Furthermore, if sensor nodes are moving, then the deployment knowledge cannot be used, even if it is available.

In [7] a seed-based key deployment strategy to discover shared key more efficiently is described. The scheme uses the seed of a given node to derive all key indexes assigned to it. By checking these indexes, a node can determine if it shares a common key with another node, thereby obviating the need to select a key by sending out a list of challenges or key identifiers. The work also proposes to enhance the security of a channel by seeking cooperation from other nodes, at the expense of increased communication overhead.

Both [3] and [1] propose a scheme to support key authentication by generating unique pairwise keys. In [1], a node loads a set of node IDs and a unique pairwise key  $K$  is generated for each pair of nodes. Hence, if  $K$  is used to secure communication, both nodes are certain of the identity of each other since no other node pair could hold  $K$ . In [3] the random key pre-distribution is combined with Blom's key pre-distribution scheme [1] to achieve  $\lambda$ -secure. This level of security is achieved only if an adversary cannot compromise more than  $\lambda$  nodes; uncompromised nodes remain perfectly secure.

When more than  $\lambda$  nodes are captured, the entire network may be compromised.

While significant advances in secure random key pre-distribution schemes have been achieved, the "per-hop key exposure" problem remains to be addressed. In this paper, we focus on path-key establishment in random key pre-distribution schemes. The objective is to eliminate "per-hop key exposure" in key pre-distribution schemes in order to improve the path-key security. The approach is to use multiple node-disjoint paths to send fragments of the negotiated key, in way such that no single node, other than the intended destination including an attacker, may be in possession of all fragments. The details of the proposed scheme are discussed next.

## III. END-TO-END PAIRWISE KEY ESTABLISHMENT

As stated above, the path-key establishment exposes keys to each intermediate node along the routing path. In order to enhance the security of symmetric key establishment, we propose an End-to-End Pairwise Key Establishment scheme which leverages multiple paths for key negotiation and establishment. The following assumptions are made in our scheme and security analysis.

- Sensor nodes are not tamper resistant.
- An attacker can randomly compromise at most  $x$  out of  $n$  nodes.
- A Node Disjoint Routing Protocol (*NDRP*), such as the one described in [5], is used to find node-disjoint paths<sup>1</sup>.
- Secure links have been established among neighboring nodes in the network. This can be easily achieved using a secure key pre-distribution scheme such as the one described in [4].

### A. End-to-End Key Establishment Scheme

Consider a network with a total number of  $n$  nodes, where a secure topology has been established using a shared-key discovery phase. Furthermore, assume that node  $N1$  needs to set up a pair-wise key with another node  $N2$ . This can be achieved using the following steps:

- $N1$  uses *NDRP* to find a set,  $PS$ , of node-disjoint secure paths to  $N2$ .
- Let  $s = |PS|$ , represent the size of the *NDRP* set  $PS$ . Node  $N1$  selects a key  $K$  and divides it into  $s$  fragments,  $K_1, K_2 \dots K_s$ , such as  $K = K_1 \cup K_2 \cup \dots \cup K_s$ , where  $K_i \cup K_{i+1}$  represents the concatenation of  $K_i$  and  $K_{i+1}$ . Each fragment contains a sequence number, and the last fragment contains a Cyclic Redundancy Checksum code to verify the correctness of the assembled packet.
- $N1$  sends  $K_i$  through the  $i_{th}$  secure path.
- Upon receiving all  $s$  fragments of the key, node  $N2$  reproduces the key  $K$ , and uses it for secure communication with  $N1$ .

Notice that the proposed scheme does not depend on the algorithm used to produce the key  $K$ . Consequently, any algorithm to produce a secure key can be used. An issue related to path selection, however, still remains to be addressed: how

<sup>1</sup>A large body of research work has focused on finding node-disjoint paths in a network. The focus of our scheme is not on building node-disjoint paths, but on what paths should be used to ensure the secure exchange of symmetric keys across a routing path. As such, the proposed scheme does not depend on a specific *NDRP* algorithm.

many node-disjoint paths must be discovered by the underlying *NDRP* to ensure a high level of security with low level of overhead? The answer to this question depends on the number of nodes an attacker may compromise.

Assuming that an attacker can compromise at most  $x$  nodes in the network, a trivial solution would be to build a set of at least  $x+1$  node-disjoint paths to achieve maximum security. However, if  $x$  is large, the network connectivity may not be “rich” enough to produce  $x+1$  node-disjoint paths. Furthermore, a first look at the problem may lead to believe that the larger the set of node-disjoint paths is the more secure the scheme would be. Contrary to intuition, however, we will show that this is not always true. To this end, we propose a security analysis model to determine how secure a path-key establishment scheme is, given different sets of node-disjoint paths.

### B. Security analysis

In this section, we present a model to determine the probability that a path-key  $K$  is revealed if a node-disjoint secure path set,  $PS$ , is used. The notation used in this analysis are listed in Table I. Assume that a node-disjoint path set,  $PS$ , is used. The

TABLE I  
NOTATION

$K$	: A pairwise key to be established
$n$	: Total number of nodes in the network <sup>2</sup>
$x$	: The maximum number of nodes an attacker can capture
$X$	: The set of nodes compromised. $ X  \leq x$
$PS$	: $\{P_1, P_2 \dots P_s\}$
$s$	: $ PS $ , number of secure paths in $PS$
$l_i$	: Intermediate hop counts of path $P_i$
$N_{p_i}$	: Intermediate nodes set of path $P_i, \forall 1 \leq i \leq s$
$N_{p_{s+1}}$	: The rest of nodes not in $PS$
$NX_i$	: The set of nodes compromised in $N_{p_i}$

key  $K$  is divided into  $s = |PS|$  fragments. Each fragment of  $K$  is transmitted over a selected path in  $PS$ . The key  $K$  could be reproduced if and only if all  $s$  fragments are received. An attacker, trying to capture  $K$ , must compromise at least one node in each path in the set  $PS$ . Obviously, if  $x < s$  then the probability of this event happening is zero. Otherwise, the probability of a path-key being exposed is the probability of selecting a set  $X$  out of  $n$  nodes such that  $X \cap N_{p_i} \neq \emptyset, \forall 1 \leq i \leq s$ .

In this framework, the security risk  $r$  is defined as the probability of a key sent through a set of  $s$  paths,  $\{P_1, P_2 \dots P_s\}$ , being revealed when an attacker captures  $x$  out of  $n$  nodes. We denote this probability as  $prob[\{P_1, P_2 \dots P_s\}, x, n]$ . There are  $\binom{n}{x}$  cases which result in  $x$  out of  $n$  nodes being randomly compromised. We need to compute how many of these cases will reveal the key  $K$  to an attacker. A single selection of  $x$  from  $n$  nodes could be represented as a  $(s+1)$ -tuple,  $(NX_1, NX_2 \dots NX_s, NX_{s+1})$ , in which  $NX_i$  is the set of nodes captured from  $P_i$  and  $NX_i \subseteq N_{p_i}, \forall 1 \leq i \leq s$ . Now the cases in which the key  $K$  would be exposed are equivalent to those tuples  $(NX_1, NX_2 \dots NX_s, NX_{s+1})$  such that

$NX_i \neq \emptyset, \forall 1 \leq i \leq s$ . So

$$prob[\{P_1, P_2 \dots P_s\}, x, n] = \frac{|\{(NX_1 \dots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\}|}{\binom{n}{x}} \quad (1)$$

A simple procedure to compute  $|\{(NX_1 \dots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\}|$ , would be to first fix the number of nodes in  $NX_i$  and then determine how many possible cases exist. It is hard, however, to list all possible distributions of  $x$  nodes over these  $s+1$  sets. We, therefore, use a different method to compute  $|\{(NX_1 \dots NX_{s+1}) \mid NX_i \neq \emptyset, \forall 1 \leq i \leq s\}|$ . In the following, we describe the proposed method.

We index nodes in each path  $P_i$  from 1 to  $l_i$ . So  $N_{p_i} = \{N_{i1} \dots N_{il_i}\}$ . A new  $s$  tuple  $S(j_1, j_2 \dots j_s)$  denotes the set of cases  $\{(NX_1 \dots NX_s, NX_{s+1}) \mid NX_i \neq \emptyset$ , and the largest index of  $NX_i$  is  $j_i, \forall 1 \leq i \leq s\}$ . Since  $j_i \leq l_i, \forall 1 \leq i \leq s$ , there are totally  $l_1 \times l_2 \dots \times l_s$  possible tuples. For example, if  $n=6, x=3, PS=\{P_1, P_2\}$  and  $l_1 = 2, l_2 = 3$ . We index nodes in  $P_1$  and  $P_2$  so that  $P_1 = \{N_{11}, N_{12}\}, P_2 = \{N_{21}, N_{22}, N_{23}\}$ . The node not in  $PS$  is  $N_6$ . So  $S(1,1)=(\{N_{11}\}, \{N_{21}\}, \{N_6\})$ ,  $S(2,2)=(\{N_{12}\}, \{N_{22}\}, \{N_6\}), (\{N_{12}, N_{11}\}, \{N_{22}\}, \emptyset), (\{N_{12}\}, \{N_{22}, N_{21}\}, \emptyset)$ .

**Lemma 1:**  $S(j_1, j_2 \dots j_s) \cap S(j'_1, j'_2 \dots j'_s) = \emptyset$  if  $\exists i, 1 \leq i \leq s$  and  $j_i \neq j'_i$

**Proof:** Let  $j_i$  be the largest index of node captured in  $P_i$ , if  $j_i \neq j'_i$ , then  $NX_i \neq NX'_i$ . Thus  $S(j_1, j_2 \dots j_s) \cap S(j'_1, j'_2 \dots j'_s) = \emptyset$ .

Based on Lemma 1, we derive that

$$prob[\{P_1, P_2 \dots P_s\}, x, n] = \frac{\sum_{j_1=1}^{l_1} \sum_{j_2=1}^{l_2} \dots \sum_{j_s=1}^{l_s} |S(j_1, j_2 \dots j_s)|}{\binom{n}{x}} \quad (2)$$

For the cases of  $S(j_1, j_2 \dots j_s)$ ,  $N_{1j_1} \dots N_{sj_s}$  are compromised and the rest of the nodes could be captured either from the nodes not in  $PS$  or from those nodes in path  $P_i$  with smaller index than  $j_i$ . There are  $n - \sum_{m=1}^s l_m$  nodes not in any path and  $(\sum_{i=1}^s j_i) - s$  nodes in  $PS$  with smaller index, So  $|S(j_1, j_2 \dots j_s)| = \binom{n - \sum_{m=1}^s l_m - s + \sum_{i=1}^s j_i}{x - s}$ . Thus

$$r = prob[\{P_1, P_2 \dots P_s\}, x, n] = \frac{\sum_{j_1=1}^{l_1} \sum_{j_2=1}^{l_2} \dots \sum_{j_s=1}^{l_s} \binom{n - \sum_{m=1}^s l_m - s + \sum_{i=1}^s j_i}{x - s}}{\binom{n}{x}} \quad (3)$$

From Equation 3, we know that the security risk of our scheme is related to both the number of node-disjoint paths in  $PS$  and the hop count of each path in  $PS$ . A path set  $PS$  with more paths could be less secure. For example, if  $n=100, x=10$  then  $prob[\{3, 4, 8\}, 100, 10] = 0.045$  however  $prob[\{1, 5\}, 100, 10] = 0.038$ , so path set  $\{1, 5\}$  is more secure than  $\{3, 4, 8\}$ . In order to study the relation between the security risk of our scheme and the number of node-disjoint paths, we isolate the effect of path hop count by assuming that we always find another node-disjoint path with the same length. Fig 2 depicts the relation between  $s, l$  and  $r$  in a 100 nodes network with 10 of them being captured randomly.

Fig 2 shows that although the security is improved by using more node-disjoint paths, the improvement by adding one more path decreases as the number of node-disjoint paths increases.

<sup>2</sup>During the analysis, we assume that the source and destination node will not be compromised, thus they are not counted in the total number of nodes  $n$ .

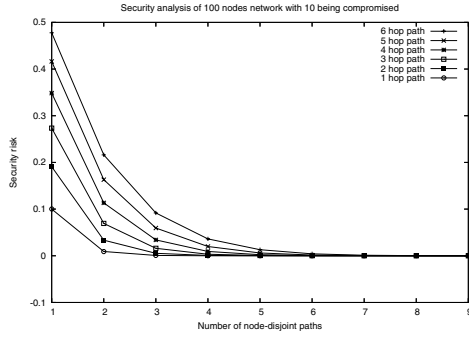


Fig. 2. Security analysis of equal path hop count

For example, when one 6-hop path is used, the security risk decreases from 0.47 to 0.21 by adding one more 6-hop node-disjoint paths. However if more than 5 node-disjoint paths are used, the security risk decrease by adding one more path will be less than 0.01.

In a real network, it is unlikely that the node-disjoint paths found would be the same length. We take a path set  $\{P_1, P_2, P_3, P_4, P_5, P_6\}$  with  $l_1 = 1, l_2 = 3, l_3 = 4, l_4 = 6, l_5 = 8, l_6 = 9$  and select these node-disjoint paths in the order of their hop count. Fig 3 shows how security risk varies with number of node-disjoint paths. We can see that the benefit from adding one more path is also decreasing. So after a number of node-disjoint paths are selected, it is not worth to find more node-disjoint paths. In other words, given security risk  $r$  we need only use a certain number of node-disjoint paths with hop count constraint. The following properties of path-set can help to define a condition of path-set selection.

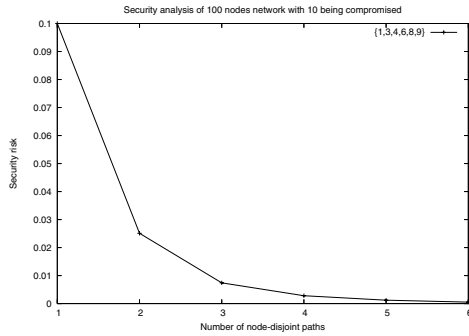


Fig. 3. Security analysis of a real path set

**Lemma 2:**  $prob\{\{l_1 \dots l_s\}, x, n\} \leq prob\{\{l \dots l\}_s, x, n\}^3$  if  $\sum_{j=1}^s l_j = s \times l$

**Lemma 3:**  $prob\{\{l_1 \dots l_s\}, x, n\} \leq prob\{\{l \dots l\}_s, x, n\}$  if  $\sum_{j=1}^s l_j \leq s \times l$

Lemma 2 can be proved by strong induction, and Lemma 3 follows from Lemma 2. See Appendix for details.

From Lemma 3, we know that given  $(s, l)$  a path-key sent through  $s$  node-disjoint paths set  $\{l_1, l_2 \dots l_s\}$  can be compromised with probability less or equal than  $prob\{\{l \dots l\}_s, x, n\}$  if  $\sum_{j=1}^s l_j \leq s \times l$ . Given  $r$  and  $s$ , we can use Equation 3 to compute the maximum  $l$  such that  $prob\{\{l \dots l\}_s, x, n\} \leq r$  and  $prob\{\{l \dots l\}_{s+1}, x, n\} > r$ . If node-disjoint path discovering algorithm *NDRP* can find such  $s$  node-disjoint paths, then we do not have to find more paths.

<sup>3</sup> $\{l \dots l\}_s$  means there are  $s$  paths with the same hop count  $l$  in this set.

### C. Reliable End-to-End Path-key Establishment

In our basic scheme, all fragments are required to reproduce the original key. If any path in  $PS$  is broken or if a node in this path drops the key information, then the key will have to be re-divided and resent. In order to make our scheme more reliable, we inject redundant information into our fragments such that not all fragments are required to rebuild the original key.

We use the scheme in [6] to generate redundant key information. Initially,  $c$  random  $b$ -vectors  $a_1, a_2 \dots a_c$  are selected with any  $b$  of them linearly independent to form a  $c \times b$  matrix  $A$ . The  $k$ -bit key  $K$  is divided into  $t = \frac{k}{b}$  segments  $K_1, K_2 \dots K_t$ , which form another  $b \times t$  matrix  $B$ . Then each piece  $w_i$  is gained by multiplying the original key  $K$  with corresponding random vector  $a_i$ .  $w_i = [a_i K_1, a_i K_2 \dots a_i K_t]$ . After any  $b$  segments  $W'_{b \times t} = \{v_1, v_2 \dots v_b\}$  are received, the random vectors associated with each piece are collected to form a  $b \times b$  matrix  $A'$ . Then the original key  $K$  can be reproduced as  $K = [A']^{-1} \times W'_{b \times t}$ . Please refer [6] for more details.

Since the key is divided into  $c$  pieces and these pieces are sent through  $s$  node-disjoint path set  $PS$  ( $c > s$ ), we need to determine how many pieces each node-disjoint path should carry. If every link is equally reliable, then the reliability of a path depends on the path length. Shorter paths are more reliable and also have less delay, so they should be able to carry more pieces. We allocate one piece for each path in  $PS$  first to ensure that every path will be used, and then the extra  $(c - s)$  segments are allocated according to path reliability. Assume every link fails with the same probability  $p$ , then a path with length  $l_j$  will get  $n_j = 1 + (c - s) \times \frac{(1-p)^{l_j}}{\sum_{i=1}^s (1-p)^{l_i}}$  segments.

We sort  $n_j$  such that  $n_1 \geq n_2 \geq \dots \geq n_s$ . If our scheme tolerates failure of the first  $f$  paths, then it could tolerate any  $f$  path failure because the first  $f$  paths carry the maximum number of pieces among any  $f$  paths. In order to tolerate the first  $f$  paths, the total segments in the rest paths should be equal to or larger than  $b$ . The maximum number of path failure can be tolerated is such  $f$  that  $\sum_{j=f+1}^s n_j \geq b$  and  $\sum_{j=f+2}^s n_j < b$ . While tolerating  $f$  path failure, our scheme becomes less secure. However, it is at least as secure as using any  $d$  paths in  $PS$ ,  $\sum_{j=1}^{d-1} n_j < b$  and  $\sum_{j=1}^d n_j \geq b$ .  $\sum_{j=1}^{d-1} n_j < b$  ensures that any  $d-1$  paths in  $PS$  will carry fewer than  $b$  key fragments, so an attacker has to compromise at least  $d$  paths in  $PS$  in order to get the key. For example, If  $b = 5, c = 9, s = 3, p = 0.05$  and  $l_1 = 2, l_2 = 4, l_3 = 6$ , then  $n_1 = 3, n_2 = 3, n_3 = 3$ . In this case one path failure is allowed but the scheme is as secure as any two paths. If we increase  $b$  to 7, then an attacker has to compromise all three paths in order to capture 7 pieces to rebuild the key. Thus the scheme becomes more secure, but then no path failure will be allowed. Generally the smaller  $b$  is, the more reliable and the less secure our scheme would be. The parameters  $b, c$  can be tuned to balance the reliability and security of our scheme.

### D. Overhead Analysis

There are two kinds of overhead in our scheme: computational overhead and communication overhead. Given a security risk  $r$ , number of nodes  $n$  and maximum number of nodes  $x$  an attacker could compromise, we compute a sequence of tuples  $(s, l)$  and use these tuples to determine when *NDRP* stops. It could be done off-line before sensors are deployed and a table of  $(s, l)$  could be loaded into each sensor.

In our scheme, the *NDRP* will need to find multiple node-disjoint paths for key establishment. It will incur extra routing overhead. However since the path-key establishment only needs to be run once for data communication unless the path-key is revoked and a new key needs to be negotiated. Generally, the path-key establishment occurs once during a long period of time for a pair of nodes. Also because we only send a small piece of key information through each path, every path is used for a very short period of time. So path maintenance is unnecessary if mobility is not extremely high.

#### IV. CONCLUSION

This paper addresses the “per-hop key exposure problem” commonly encountered in key pre-distribution schemes in WSNs. To overcome this problem, we propose an End-to-End Pairwise Key Establishment scheme, which improves the security of symmetric key distribution in WSNs. The scheme uses multiple node-disjoint paths for the negotiation and exchange of symmetric keys. The analysis shows that the scheme is highly secure against node compromise. Furthermore the scheme can be added to most existing key pre-distribution schemes without significant changes.

#### REFERENCES

- [1] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” *IEEE Symposium on Security and Privacy*, pp. 197–213, May 2003.
- [2] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” *Proceedings of the IEEE INFOCOM*, March 2004.
- [3] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” *Proceedings of the 10th ACM Conference on Computer and Communication Security(CCS)*, pp. 42–51, October 2003.
- [4] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” *Proceedings of the 9th ACM conference on Computer and Communication Security*, pp. 41–47, November 2002.
- [5] X. Li and L. Cuthbert, “Node-disjointness-based multipath routing for mobile ad hoc networks,” *Proceedings of the 1st ACM international workshop on PE-WASUN*, pp. 23–29, October 2004.
- [6] P. Papadimitratos and Z. Haas, “Secure message transmission in mobile ad hoc networks,” *Elsevier Ad Hoc Networks Journal*, vol. 1, Jan/Feb/March 2003.
- [7] R. D. Pietro, L. V. Mancini, and A. Mei, “Random key assignment for secure wireless sensor networks,” *ACM Workshop on Security of Ad Hoc and Sensor Networks*, October 2003.

#### APPENDIX

##### A. Prove of Lemma 2

We reform Equation 3 into another form in order to prove Lemma 2.  $prob\{\{l_1 \dots l_s\}, x, n\}$

$$= \frac{\sum_{j=1}^{l_1} \sum_{j_2=1}^{l_2} \dots \sum_{j_s=1}^{l_s} \binom{n - \sum_{m=1}^s l_m - s + \sum_{i=1}^s j_i}{x-s}}{\binom{n}{x}}$$

$$= \frac{\sum_{j=1}^s \sum_{i=1}^{l_i} e_j \times \binom{j+n-s-\sum_{m=1}^s l_m}{x-s}}{\binom{n}{x}}.$$

$e_j$  is the number of s-tuples whose summation of all elements is  $j$ .  $e_j = e(\{l_1, l_2 \dots l_s\}, j) = |E(\{l_1, l_2 \dots l_s\}, j)| = |\{(i_1 \dots i_s) \mid \sum_{m=1}^s i_m = j \text{ and } i_m \leq l_m \forall 1 \leq m \leq s\}|$ . Before we prove Lemma 2, let us look at three properties of  $e_j$ .

*Property 1:*  $e(\{l_1 \dots l_{s-1}, l_s\}, j) = e(\{l_1 \dots l_{s-1}, l_s - 1\}, j) + e(\{l_1 \dots l_{s-1}\}, j - l_s)$

For all s-tuples in  $\{(i_1 \dots i_s) \mid \sum_{m=1}^s i_m = j \text{ and } i_m \leq l_m \forall 1 \leq m \leq s\}$ , the  $s$ th index  $i_s$  is either  $l_s$  or less than  $l_s$ . If  $i_s = l_s$  then the summation of all other elements should be  $j - l_s$ . There is totally  $e(\{l_1 \dots l_{s-1}\}, j - l_s)$  such s-tuples.

When  $i_s < l_s$ , the number of tuples equal to  $e(\{l_1 \dots l_{s-1}, l_s - 1\}, j)$ . Thus  $e(\{l_1 \dots l_{s-1}, l_s\}, j) = e(\{l_1 \dots l_{s-1}, l_s - 1\}, j) + e(\{l_1 \dots l_{s-1}\}, j - l_s)$

*Property 2:*  $e(\{l_1, l_2 \dots l_s\}, j) = e(\{l_2 \dots l_s\}, j - 1) + e(\{l_2 \dots l_s\}, j - 2) + \dots + e(\{l_2 \dots l_s\}, j - l_1)$

Property 2 can be proved by applying Property 1 on  $l_1$ .  $e(\{l_1, l_2 \dots l_s\}, j) = e(\{l_1 - 1, l_2 \dots l_s\}, j) + e(\{l_2 \dots l_s\}, j - l_1) = e(\{l_1 - 2, l_2 \dots l_s\}, j) + e(\{l_2 \dots l_s\}, j - (l_1 - 1)) + e(\{l_2 \dots l_s\}, j - l_1) = e(\{0, l_2 \dots l_s\}, j) + e(\{l_2 \dots l_s\}, j - 1) + e(\{l_2 \dots l_s\}, j - 2) + \dots + e(\{l_2 \dots l_s\}, j - l_1) = e(\{l_2 \dots l_s\}, j - 1) + e(\{l_2 \dots l_s\}, j - 2) + \dots + e(\{l_2 \dots l_s\}, j - l_1)$

*Property 3:*  $e(\{l_1 \dots l_m\}, j) \leq e(\{l_1 \dots l_m + y\}, j + y)$

For any s-tuple  $(i_1 \dots i_{m-1}, i_m)$  in  $E(\{l_1 \dots l_m\}, j)$ , we can construct another s-tuple  $(i'_1 \dots i'_{m-1}, i'_m)$  in  $E(\{l_1 \dots l_m + y\}, j + y)$  as following:  $i'_m = i_m + y$  and  $i'_t = i_t \forall 1 \leq t \leq m - 1$ .  $\sum_{t=1}^m i'_t = \sum_{t=1}^m i_t + y = j + y$  and  $i'_t = i_t \leq l_t \forall 1 \leq t \leq m - 1$ ;  $i'_m = i_m + y \leq l_m + y$ . Thus, this new tuple belongs to  $E(\{l_1 \dots l_m + y\}, j + y)$ .

If  $\sum_{j=1}^s l_j = s \times l$ , then  $prob\{\{l_1 \dots l_s\}, x, n\} = \frac{\sum_{j=s}^{l \times s} e(\{l_1 \dots l_s\}, j) \times \binom{j+n-s-s \times l}{x-s}}{\binom{n}{x}}$  and  $prob\{\{l \dots l\}, x, n\} = \frac{\sum_{j=s}^{l \times s} e(\{l \dots l\}, j) \times \binom{j+n-s-s \times l}{x-s}}{\binom{n}{x}}$ . So we can prove Lemma 2 by proving  $e(\{l_1 \dots l_s\}, j) \leq e(\{l \dots l\}, j) \forall s \leq j \leq s \times l$  using strong induction.

Step 1:  $s = 1$ . Since  $l_1 = l$ ,  $e(\{l_1\}, j) = e(\{l\}, j) \leq e(\{l\}, j)$

Step 2: Assume  $e(\{l_1 \dots l_m\}, j) \leq e(\{l \dots l\}_m, j) \forall m \leq j \leq m \times l$  for all  $m \leq s - 1$ .

$$e(\{l_1, l_2 \dots l_s\}, j) = e(\{l_2 \dots l_s\}, j - 1) + e(\{l_2 \dots l_s\}, j - 2) + \dots + e(\{l_2 \dots l_s\}, j - l_1)$$

$$= e(\{l_2 \dots l'_s\}, j - 1) + e(\{l_2 \dots l_{s-1}\}, j - 1 - l_s) + \dots + e(\{l_2 \dots l_{s-1}\}, j - 1 - l_s + (l - l_1 - 1)) + e(\{l_2 \dots l'_s\}, j - 2) + e(\{l_2 \dots l_{s-1}\}, j - 2 - l_s) + \dots + e(\{l_2 \dots l_{s-1}\}, j - 2 - l_s + (l - l_1 - 1)) + \dots$$

$$+ e(\{l_2 \dots l'_s\}, j - l_1) + e(\{l_2 \dots l_{s-1}\}, j - l_1 - l_s) + \dots + e(\{l_2 \dots l_{s-1}\}, j - l_1 - l_s + (l - l_1 - 1))$$

$$= e(\{l_2 \dots l'_s\}, j - 1) + e(\{l_2 \dots l'_s\}, j - 2) + \dots + e(\{l_2 \dots l'_s\}, j - l_1) + e(\{l_1, l_2 \dots l_{s-1}\}, j - l_s) + e(\{l_1, l_2 \dots l_{s-1}\}, j - l_s + 1) + \dots + e(\{l_1, l_2 \dots l_{s-1}\}, j - l_s + (l - l_1 - 1))$$

$$\leq e(\{l \dots l\}_{s-1}, j - 1) + e(\{l \dots l\}_{s-1}, j - 2) + \dots + e(\{l \dots l\}_{s-1}, j - l_1) + e(\{l_1, l_2 \dots l_{s-1} - l + l_s\}, j - l) + e(\{l_1, l_2 \dots l_{s-1} - l + l_s\}, j - l + 1) + \dots + e(\{l_1, l_2 \dots l_{s-1} - l + l_s\}, j - l_1 - 1)$$

$$\leq e(\{l \dots l\}_{s-1}, j - 1) + e(\{l \dots l\}_{s-1}, j - 2) + \dots + e(\{l \dots l\}_{s-1}, j - l_1) + e(\{l \dots l\}_{s-1}, j - l) + e(\{l \dots l\}_{s-1}, j - l + 1) + \dots + e(\{l \dots l\}_{s-1}, j - l_1 - 1) = e(\{l, l \dots l\}_s, j).$$

##### B. Prove of Lemma 3

Obviously,  $prob\{\{l_1 \dots l_s\}, x, n\} \leq prob\{\{l_1 \dots l_s + y\}, x, n\}$  if  $y \geq 0$ . So if  $\sum_{j=1}^s l_j \leq s \times l$ , then  $prob\{\{l_1 \dots l_s\}, x, n\} \leq prob\{\{l_1 \dots l_s + s \times l - \sum_{j=1}^s l_j\}, x, n\} \leq prob\{\{l \dots l\}_s, x, n\}$ .