

A Location-Adaptive Key Establishment Scheme for Large-Scale Distributed Wireless Sensor Networks

Ashok Kumar Das

Department of Computer Science and Engineering
International Institute of Information Technology, Bhubaneswar 751 013, India
Email: iitkgp_akdas2006@yahoo.co.in

Abstract—The establishment of pairwise keys between communicating neighbor nodes in sensor networks is a challenging problem due to the unsuitability of public-key cryptographic techniques for the resource-constrained platforms of sensor networks and also due to vulnerability of physical captures of sensor nodes by an adversary/enemy. In this paper, we propose a new location-adaptive key establishment scheme which is considered as an improved alternative to the path key establishment phase of bootstrapping protocol in a sensor network. Our proposed scheme offers significantly better network connectivity compared to that for the path key establishment. Moreover, our scheme has better trade-off between communication overhead, computational overhead, network connectivity and resilience against node capture attack than the path key establishment.

Index Terms—Distributed sensor networks, Location-adaptive key pre-distribution, Path key establishment, Key establishment, Security

I. INTRODUCTION

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, multi-functional sensor nodes that are small in size and communicate untethered in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks. Thus, the sensor networks give a significant improvement over the traditional sensors.

In a sensor network, many tiny computing nodes called sensors are scattered in an area for the purpose of sensing some data and transmitting data to nearby *base stations* for further processing. The transmission between the sensors is done by short range radio communications. The base station is assumed to be computationally well-equipped whereas the sensor nodes are resource-starved. The sensor nodes are usually scattered in a *sensor field* (i.e., deployment area or target field). Each of these scattered sensor nodes has the capabilities to collect data and route data back to the base station. Data are routed back to the base station by a multi-hop infrastructure-less architecture through sensor nodes. The base station may

communicate with the *task manager node* via Internet or Satellite. A survey on sensor networks could be found in [1], [2].

Key establishment protocols are used to set up the shared secrets, but the problem is too much complicated by the sensor nodes' limited computational capabilities, battery energy, and available memory. As a result, asymmetric cryptography such as RSA [15] or Diffie-Hellman key exchange protocol [6] or Elliptic Curve cryptography (ECC) [17] or ElGamal cryptosystem [9] is unsuitable for most sensor architectures due to high energy consumption and increase code storage requirements. Hence, a symmetric cipher such as DES/IDEA/RC5/AES [14], [16], [17] is the viable option for encryption or decryption of data for secret communication in sensor network.

Pairwise key establishment between neighboring sensor nodes in a sensor network is done by using a protocol which is popularly known as the *bootstrapping protocol*. A bootstrapping protocol usually involves several steps. In *key pre-distribution* procedure, each sensor node is initialized by a set of pre-distributed keys in its memory called the key ring. This is done before deployment of the sensor nodes in a target field. After deployment, a *direct key establishment* (also called the shared key discovery) procedure is performed by sensor nodes in order to establish direct pairwise keys between them. Two nodes u and v are called the *physical neighbors* if they are within communication ranges of one another. They are called the *key neighbors* if they share at least one common key in their key rings. They are finally called the *direct neighbors* if they are both the physical neighbors and the key neighbors. After direct key establishment, if the nodes fail to establish direct keys between them, they perform the *path key establishment phase*. In this phase, a secure path is established between two neighbor nodes and a secret key is transmitted securely along that path.

Several techniques [4], [7], [8], [10], [12], [13] are already proposed in order to solve the bootstrapping problem. Eschenauer and Gligor [10] proposed the basic random key predistribution called the EG scheme, in which each sensor is assigned a set of keys randomly selected from a big key pool of the keys of the sensor nodes. Chan et al. [4] proposed the q -composite key predistribution and the random pairwise keys schemes. For both the EG and

This paper is based on "A Key Reshuffling Scheme for Wireless Sensor Networks," by A. K. Das, which appeared in the Proceedings of the International Conference on Information Systems Security (ICISS 2005), Lecture Notes in Computer Science (LNCS), Volume 3803, pages 205-216, December 2005. © 2005 Springer-Verlag.

the q -composite schemes, if a small number of sensors are compromised, they may reveal to a large fraction of pairwise keys shared between non-compromised sensors. However, the random pairwise keys predistribution is perfectly secure against node captures, but there is a problem for maximum supported network size. Liu and Ning's polynomial-pool based key predistribution scheme [13] and the matrix-based key predistribution proposed by Du et al. [8] improve security considerably. An improved alternative of path key establishment is proposed in [5]. This scheme allows to establish direct keys between non-neighbor nodes in a sensor network. The main advantage of this scheme is that it provides secure routing between two non-neighbor nodes via a multi-hop path using end-to-end encryption/decryption procedure rather than link-to-link encryption/decryption procedure. As a result, this scheme requires less computational overhead for secure routing. In this paper, we present a new location-adaptive key establishment scheme which is an improved alternative to the path key establishment in order to establish pairwise keys between neighbor sensor nodes.

The rest of the paper is organized as follows. Next Section describes briefly existing related works on key pre-distribution techniques in wireless sensor networks. Section III introduces our proposed scheme. Our scheme is an improved alternative to the path key establishment of the bootstrapping protocol. Section IV presents a theoretical analysis of this scheme. In Section V, we report our simulation results. In Section VI, we compare the performances of our scheme with those for the path key establishment. Finally, we conclude the paper in Section VII.

II. RELATED WORK

In this section, we describe briefly the existing key pre-distribution schemes: the basic random key pre-distribution scheme [10], the polynomial-based key pre-distribution scheme [3] and the polynomial-pool based key pre-distribution scheme [13].

A. The basic random key pre-distribution scheme

Eschenauer and Gligor in 2002 first proposed a random key pre-distribution scheme [10]. Their scheme, henceforth referred to as the EG scheme, consists the following three phases. In the *key pre-distribution phase*, the (key) setup server chooses a large key pool \mathcal{K} of M randomly generated symmetric keys. Each key is assigned a unique identifier in the pool \mathcal{K} . For each sensor node u to be deployed, the setup server picks a random subset K_u of size m from the pool \mathcal{K} , called the key ring of the node u , and then loads this subset into its memory.

After the sensor nodes are deployed in some target field, a *direct key establishment phase* (also called the shared key discovery phase) is performed by each sensor node in the network. First of all, each sensor node locates its all physical neighbors within its communication range. Two physical neighbors can establish a secret key

between them if there exists at least one common key between their key rings. To establish a secret key between them, they exchange the key ids from their key rings in plaintext. If there is a common key id between their key rings, the corresponding key is taken as the secret key between them and they use this key for their future secure communication. Nodes which discover that they have a shared secret key in their key rings then verify that their neighbor actually holds the key through a challenge-response protocol. Since the random subsets for the nodes are drawn from the pool \mathcal{K} randomly without replacement, the same key may be used for secret communication by several pairs of neighbor nodes in the network.

The *path key establishment phase* is an optional stage, and if executed, adds to the connectivity of the network. Suppose two neighbor nodes, say, u and v fail to establish a secret key between them in the direct key establishment phase, but there exists a secure path. Once such a secure path is discovered, u generates a new random key k and securely transmits it along this path to the desired destination node. In this way, u and v can communicate secretly and directly using k . However, the main problem is that the communication overhead increases significantly with the number h of hops. For this reason, in practice, h is restricted to a small value, say 2 or 3.

This scheme provides better network connectivity if the key pool size is smaller. In this scheme, when the key pool size is chosen smaller, it leads to compromise a large fraction of secure communication links in the network even if an adversary captures a small fraction of sensor nodes in the network.

B. The polynomial-based key pre-distribution scheme

The polynomial-based key pre-distribution scheme proposed by Blundo et al. in [3] is described as follows. In the key pre-distribution phase, an offline key setup server assigns unique identifiers to all the sensor nodes to be deployed in a target field. The setup server then generates randomly a t -degree symmetric bivariate polynomial $f(x, y)$, defined by $f(x, y) = \sum_{i,j=0}^t a_{ij} x^i y^j$, where the coefficients a_{ij} ($0 \leq i, j \leq t$) are randomly chosen from a finite field $F_q = GF(q)$, q is a prime that is large enough to accommodate a symmetric cryptographic key, with the property that $f(x, y) = f(y, x)$. For each sensor node u to be deployed, the setup server computes a polynomial share $f(u, y)$. We note that $f(u, y)$ is a t -degree univariate polynomial. The setup server finally loads the coefficients of $f(u, y)$ in the memory of the sensor node u .

In the direct key establishment phase, each sensor node u first locates its physical neighbors in its communication range and broadcasts its own id to its neighbors. Let u and v be two neighbors. After receiving the id of the node v , u computes the secret key shared with v as $k_{u,v} = f(u, v)$. Similarly, v computes the secret key shared with u as $k_{u,v} = f(v, u)$. Since $f(u, v) = f(v, u)$, both the nodes u and v store the key $k_{u,v}$ for their future secret communication.

The advantage of this scheme is that any two neighbor

nodes can establish a secret key using the same symmetric bivariate polynomial $f(x, y)$, and there is no communication overhead during the pairwise key establishment process. The main drawback is that if more than t nodes in the network are compromised by an adversary, he/she easily reconstructs the original polynomial using the *Lagrange Interpolation* [11]. As a result, all the pairwise keys shared between the non-compromised nodes will also be compromised. Thus, this scheme is *unconditionally secure and t -collusion resistant*. Although increasing the value of t can improve the security property of this scheme, but it is not feasible for wireless sensor networks due to the limited memory in sensors.

C. The polynomial-pool based key pre-distribution scheme

In order to improve resilience against node capture of the polynomial-based key pre-distribution scheme [3], Liu et al. proposed the polynomial-pool based key distribution scheme [13]. The polynomial-pool based key pre-distribution scheme can be described as follows. Let $F_q = GF(q)$ be a finite field with a q (either a prime or 2^m for some positive integer m) just big enough to accommodate a symmetric cryptographic key. Let $f(x, y) \in F_q[x, y]$ be a t -degree symmetric bivariate polynomial i.e., $f(x, y) = f(y, x)$. The coefficients of the polynomial $f(x, y)$ are chosen from the finite field F_q . A *polynomial share* of $f(x, y)$ is a univariate polynomial $f(u, y)$ for some $u \in F_q$. We have, $f(u, v) = f(v, u)$.

Thus, if two shares $f(u, y)$ and $f(v, y)$ of the same polynomial $f(x, y)$ are given to two nodes, say, u and v , they can come up with the common value $f(u, v) \in F_q$ as a shared key between them. If $(t + 1)$ or more shares of $f(x, y)$ are known, one can easily reconstruct $f(x, y)$ uniquely using the *Lagrange's Interpolation* [11]. Thus, the disclosure of up to t shares does not reveal the polynomial $f(x, y)$ to an adversary and non-compromised shared keys based on $f(x, y)$ remains completely secure.

The key setup server selects a random pool \mathcal{K} of s symmetric bivariate polynomials in $F_q[x, y]$ each of degree t in x and y . Some ids $u_1, u_2, \dots, u_n \in F_q$ are also generated for the sensor nodes in the network, where n is the network size. For each sensor node u to be deployed in the network, s' polynomials, say, $f_1(x, y), f_2(x, y), \dots, f_{s'}(x, y)$ are randomly selected from \mathcal{K} and the polynomial shares $f_1(u, y), f_2(u, y), \dots, f_{s'}(u, y)$ are loaded in the key ring K_u of u . Immediately after deployment, each sensor u transmits the ids of the polynomial shares residing in its key ring. Two physical neighbors u and v having shares of some common polynomial(s) can establish a pairwise key between them.

The polynomial-pool based key distribution scheme provides significantly better resilience against node capture than the other existing schemes [3], [4], [10].

III. OUR PROPOSED SCHEME

In this section, we first discuss the path key establishment phase of the bootstrapping protocol. We then

introduce the main motivation behind the development of our proposed scheme. We describe our scheme and also discuss the details of the messages exchanged during the key establishment procedure of our scheme.

A. Path Key Establishment

This is an optional stage, and if executed, adds to the connectivity of the network. After direct key establishment, the nodes u and v which are physical neighbors not sharing a pairwise key, can establish a direct key between them as follows.

- *Step-1:* u first finds for a path $\langle u = u_0, u_1, u_2, \dots, u_{h-1}, u_h = v \rangle$ such that each (u_i, u_{i+1}) ($i = 0, 1, 2, \dots, h - 1$) is a secure link.
- *Step-2:* u generates a random number $k_{u,v}$ as the shared pairwise key between u and v and encrypts it using the shared key k_{u,u_1} between u and u_1 , and sends it to node u_1 .
- *Step-3:* u_1 retrieves $k_{u,v}$ by decrypting the encrypted key using k_{u,u_1} and encrypts it using the shared key k_{u_1,u_2} between u_1 and u_2 and sends it to u_2 .
- *Step-4:* This process is continued by every intermediate sensor node along this path until the key $k_{u,v}$ reaches to the desired destination node v .

Nodes u and v use $k_{u,v}$ as the direct pairwise key shared between them for their future secret communication.

The main issue in this phase is the *path discovery* problem, which specifies how to find a secure path between two sensor nodes. One approach is to discover a path between a source node and a destination node, the source node picks a set of intermediate nodes with which it has established direct keys. The source node then sends requests to its all these intermediate nodes. Now, if one of these intermediate nodes can establish a direct key with the destination node, a secure path will be discovered. Otherwise, this process may continue with the intermediate nodes forwarding the request. We thus note that the discovery of a secure path between two nodes is similar to a route discovery process used to establish a route between two nodes. Since this process involves more communication overhead to establish a pairwise key between nodes as the number h of hops of the path increases, in practice $h = 2$ or 3 is recommended.

B. Motivation

Due to the random selection of keying materials for the key rings of the sensor nodes, there remain some unused keys in each key ring, which are of no use for establishing secure links with the physical neighbors. A key material in the basic probabilistic key pre-distribution scheme (the EG scheme) [10], the q -composite scheme [4], or the random pairwise keys scheme [4] is simply a pre-distributed key. In the polynomial pool-based scheme [13], a key material is a t -degree symmetric bivariate polynomial from which a sensor node can compute keys shared with its physical neighbors, whereas in the pairwise key pre-distribution scheme [8], a key material is a row of the

secret matrix A_i in a key space S_i . An unused key, say k in the key ring of a sensor node u may help another node v to establish a secure link between v and its physical neighbor w with which it does not currently share a secret key. If one can discover a secure $u - v$ path, then transmitting k securely from u to v along this path achieves this goal. Thus, using this key k , two neighbor sensor nodes v and w can establish a new pairwise key k' for their future communication. This situation is depicted in Figure 1.

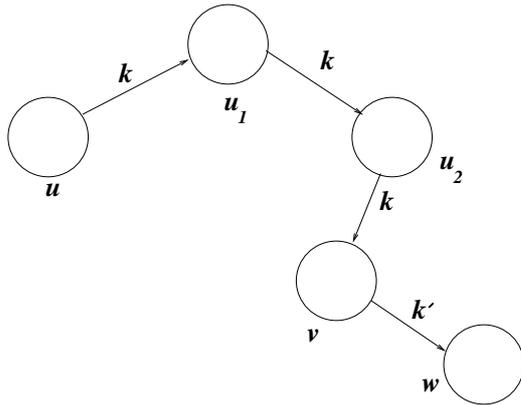


Figure 1. Indirect key establishment scenario between two neighbor nodes v and w with 3 hops. An unused key k in u 's key ring is securely transmitted to node v . This key helps another physical neighbor w of node v with which v does not share a key in order to establish a new secret key k' between v and w .

This is accomplished by the following protocol.

C. Protocol

The protocol consists of the following steps:

- 1) for each sensor node u in the sensor network:
- 2) for each unused key k in its key ring:
- 3) Transmit k securely to u 's direct neighbors.
- 4) Each direct neighbor re-transmits k securely to its direct neighbors and then deletes the key k from its memory.
- 5) Continue this process for a pre-determined number of hops.
- 6) Let v be a node that receives k securely during this process. Then, v broadcasts its id and the id of the key k to its physical neighbors with which it does not currently share any keys. (If sending the id of the key k is a potential threat for the network, v can ask its physical neighbors to solve some puzzle. The puzzle is encrypted with the key k and this encrypted puzzle is sent. Thus, those physical neighbors can solve this puzzle will have the key k .)
- 7) Let w be a physical neighbor such that the id of the key k is found in its key ring. w sends a request to v to transmit a newly pairwise key securely to be established between v and w .
- 8) v generates a random number, say k' which is considered as a symmetric secret key between v

and w . v encrypts this key k' with k , and sends the encrypted key and the id of u to w .

- 9) w retrieves k' by decrypting the encrypted key using k .
- 10) Nodes v and w store the key k' for their future communications.
- 11) Node v deletes the unused key k from its memory.

We note that during the key establishment procedure, all the intermediate nodes along a secure path delete the transmitted unused key. However, discovering a secure $u - v$ path, in particular, a long one, is costly in terms of communication overhead. For this reason, we restrict the lengths of these paths to a pre-determined threshold.

D. Protocol Messages

The above protocol is summarized below. In this notation, k_{uv} refers to the unique pairwise key shared by u and v . $u \rightarrow v : M$ refers to a message M sent from u to v . $E_k\{M\}$ refers to a message M encrypted using key k . $MAC_k(M)$ refers to the message authentication code (MAC) for the message M , under the key k . We refer N_u as a nonce generated by node u .

In order to establish a direct key between two neighbor sensor nodes v and w via a secure $u - v$ path, the following messages to be exchanged:

- 1) To transmit an unused key k along a secure $u - v$ path: $\langle u = u_0, u_1, u_2, \dots, u_{h+1} = v \rangle$ having h intermediate nodes u_1, u_2, \dots, u_h :

$$u_{i-1} \rightarrow u_i : (E_{k_{u_{i-1}, u_i}}\{u_{i-1}, u_i, k\}, MAC_{k_{u_{i-1}, u_i}}(E_{k_{u_{i-1}, u_i}}\{u_{i-1}, u_i, k\})) \text{ for } i = 1, 2, \dots, h + 1.$$

- 2) To transmit a randomly generated key k' to w by v :
 $v \rightarrow w : E_k\{w, v, u, k'\}, MAC_k(E_k\{w, v, u, k'\})$.
- 3) To send a nonce to v by w :
 $w \rightarrow v : E_{k'}\{w, v, N_w\}, MAC_{k'}(E_{k'}\{w, v, N_w\})$.

The above steps are to be carried out after the direct key establishment phase. In order to reduce communication overhead, we restrict the number of hops to 2 or 3.

IV. ANALYSIS OF OUR SCHEME

In this section, we derive the key connectivity of the network, i.e., the probability of establishing keys between two nodes directly or indirectly. We also analyze the security for our proposed scheme.

A. Network Connectivity

1) Network Connectivity of Our Scheme under the EG Scheme: From the analysis of [10], it follows that the probability of two neighbor sensor nodes establishing a pairwise key is given by

$$p = 1 - \prod_{i=0}^{m-1} \frac{M - m - i}{M - i}, \tag{1}$$

where M denotes the size of the key pool, and m the size of the key ring (in terms of the keys) of each sensor

node (i.e., each sensor node is capable of storing m pre-distributed keys in its key ring).

Let d denote the average number of neighbors that each sensor node can contact. Let p' be the probability that an unused key k is found in a node's key ring, and P_h denote the probability that there exist a secure $u - v$ path, say $\langle u = u_0, u_1, u_2, \dots, u_{h+1} = v \rangle$ having h intermediate nodes u_1, u_2, \dots, u_h .

Derivation of p' :

We have $p' = 1 -$ (probability that an unused key k is not found in a node's key ring). The total number of ways to select m keys from the key pool of size M is $\binom{M}{m}$. For a fixed key ring K_u of a node u , the total number of ways that key k will not be found in K_w of another node w is $\binom{M-1}{m}$. Thus, we have,

$$p' = 1 - \frac{\binom{M-1}{m}}{\binom{M}{m}} = \frac{m}{M}. \tag{2}$$

Derivation of P_h :

Let us first consider $h = 1$. In this case, we have to compute the probability P_1 that there exists a secure 1-hop path between nodes u and v , say $\langle u, u_1, v \rangle$. Consider any one of the d neighbors of the source node u . The probability that it shares a pairwise key with both the source node u and the destination node v is p^2 . As long as one of the d nodes can act as an intermediate node, the source and the destination nodes can establish a common key. Thus, we have:

$P_1 =$ probability that a secure 1-hop $u - v$ path exists between u and v
 $=$ probability that u and v establish a pairwise key (directly or indirectly)
 $= 1 - (1 - p)(1 - p^2)^d$.

We can generalize this formula for h hops. Hence, we obtain:

$$P_h = \begin{cases} 1 - (1 - p)(1 - p^2)^d, & \text{if } h = 1. \\ 1 - (1 - P_{h-1})(1 - p P_{h-1})^d, & \text{if } h \geq 2. \end{cases} \tag{3}$$

Let $P_{indirect}$ represent the probability that two neighbor sensor nodes v and w establish a pairwise key after applying our scheme for h hops. It is easy to observe from the protocol described in Section III.C that two neighbor nodes v and w can establish a pairwise key only if there exists a secure h -hop $u - v$ path: $\langle u = u_0, u_1, u_2, \dots, u_{h+1} = v \rangle$ having h intermediate nodes u_1, u_2, \dots, u_h as well as the id of an unused key k in key ring K_u of node u exists in key ring K_w of node w . As a result, we obtain the formula for $P_{indirect}$ as:

$$P_{indirect} = 1 - (1 - P_h)(1 - p') \text{ for } h \geq 1. \tag{4}$$

For example, let us consider $h = 1$. Then, we have, $P_{indirect} = 1 - (1 - p')(1 - P_1) = 1 - (1 - p')(1 - p)(1 - p^2)^d$.

Figure 2 illustrates the probabilities, $P_{indirect}$ for $M = 100000$, $m = 100$ (so that $p = 0.0953$), and for

several values of d . This figure shows that the network connectivity increases as the average number of neighbors increases. It also tells us that even if the network is likely to remain disconnected with high probability initially, one can obtain high network connectivity after applying few hops of our scheme.

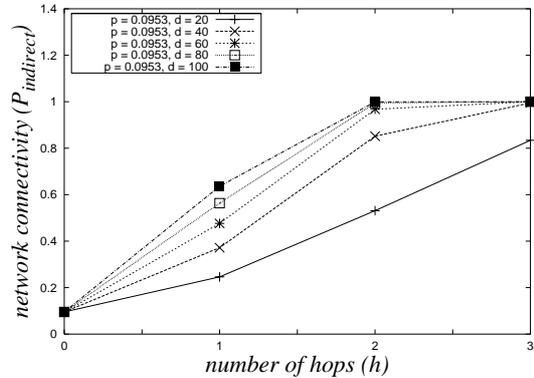


Figure 2. Theoretical network connectivity of our scheme applied under the EG scheme. Assume $M = 100000$, $m = 100$, and $d = 20, 40, 60, 80, 100$.

2) Network Connectivity of Our Scheme under the Polynomial-Pool Scheme:

Let s be the size of the polynomial-pool and s' the number of the polynomial shares given to each node's key ring. From the analysis of [13], it follows that the probability of two neighbor sensor nodes establishing a pairwise key is

$$p = 1 - \prod_{i=0}^{s'-1} \frac{s - s' - i}{s - i}. \tag{5}$$

The probability $P_{indirect}$ of two neighbor sensor nodes sharing a key using our protocol can be derived analogously to the derivation in Section IV.A.1. Hence, we have:

$$P_{indirect} = 1 - (1 - P_h)(1 - p') \text{ for } h \geq 1, \tag{6}$$

where the symbols h , P_h , and p' have the same meanings as in Section IV.A.1, and

$$p' = 1 - \frac{\binom{s-1}{s'}}{\binom{s}{s'}} = \frac{s'}{s}. \tag{7}$$

Figure 3 shows the probabilities of establishing direct keys between neighbor sensor nodes when our protocol is applied under the polynomial-pool scheme for $s = 500$, $s' = 5$, and for different values of d . Under these parameters we have $p = 0.0492$ initially, i.e., the network is likely to remain disconnected with high probability. It is clear from this figure that the network becomes connected with high probability if the number of hops as well as the average number of neighbors of each sensor node are increased.

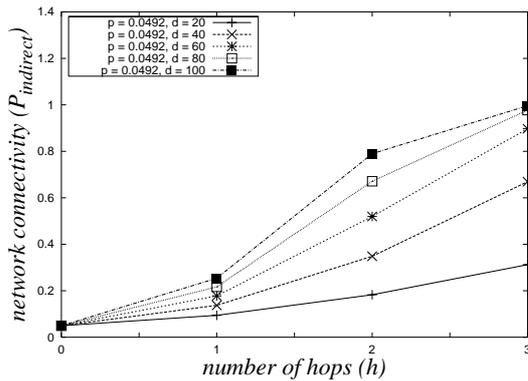


Figure 3. Theoretical network connectivity of our scheme applied under the polynomial-pool scheme. Assume $s = 500$, $s' = 5$, and $d = 20, 40, 60, 80, 100$.

B. Security Considerations

In path key establishment, two physical neighbors, say u and v establish a direct key via a secure h -hop path: $\langle u = u_0, u_1, u_2, \dots, u_{h+1} = v \rangle$. In this case, a link $u - v$ is compromised if either of its endpoint nodes u and v is compromised, or if either of the intermediate nodes u_1, u_2, \dots, u_h is compromised.

In our scheme, two physical neighbors, say v and w establish a direct key via a secure h -hop $u - v$ path: $\langle u = u_0, u_1, u_2, \dots, u_{h+1} = v \rangle$. Thus, a link $v - w$ is compromised if either of its endpoint nodes v and w is compromised, or if either of the intermediate nodes u_1, u_2, \dots, u_h is compromised, or if the initiating node u is compromised.

We observe that the security of the intermediate nodes is same for both cases. As a result, for the path key establishment, the probability that any link between two nodes is compromised is the probability that either of its endpoint nodes is compromised. If some fraction f of the total number of nodes in the network is compromised, then the required probability of the link being compromised is $1 - (\text{probability that neither endpoint nodes are compromised}) = 1 - (1 - f)^2 \approx 2f$ if f is small.

On the other hand, for our scheme, a link is compromised only if either of its endpoint nodes v and w is compromised, or the initiating node u is compromised. Thus, if some fraction f of the total number of nodes in the network is compromised, then the fraction of total links compromised will be about $1 - (\text{probability that neither endpoint nodes } v \text{ and } w \text{ are compromised nor the initiating node } u \text{ is compromised}) = 1 - (1 - f)^3 \approx 3f$ if f is small. We note that the security against node capture of our scheme compares favorably with that for the path key establishment for the ideal case.

V. SIMULATION RESULTS

In this section, we discuss the simulation results of the network connectivity of our scheme under both the EG scheme and the polynomial-pool based scheme. We also compare the simulation results of network connectivity of our scheme with the path key establishment under both the EG scheme and the polynomial-pool based scheme.

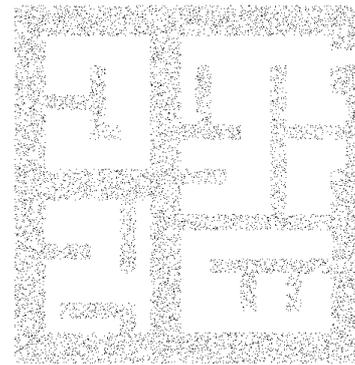


Figure 4. Arbitrary deployment model-I

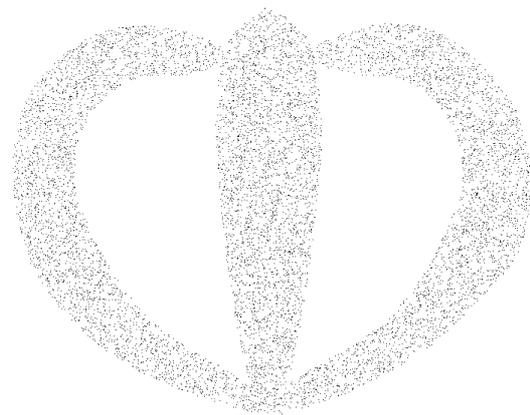


Figure 5. Arbitrary deployment model-II

The networks are simulated on several arbitrary deployment fields. Our simulation suggests that the results of network connectivity of our scheme are not much sensitive with respect to different arbitrary deployment models. We have considered two deployment models which are shown in Figures 4 and 5 for simulation of our scheme. Each dot in the figures represents the deployment location of a sensor node.

For the EG scheme, we have considered the following parameter values:

- The number of nodes in the network is $n = 10000$.
- The size of the key pool (in terms of keys) is $M = 100000$.
- The size of the key ring of each sensor node is $m = 100$.
- The average number of sensor nodes of each node is $d = 100$.
- The communication range of each sensor node is $\rho = 30$ meters.
- The area A of the deployment field is chosen so that the maximum network size becomes $n = \frac{A \times (d+1)}{\pi \rho^2}$.

The theoretical as well as simulated network connectivity probabilities for our scheme under the EG scheme are shown in Figure 6. This figures clearly illustrates that the theoretical results closely tally with the simulation results under the EG scheme.

For the polynomial-pool scheme, we have taken the following parameter values:

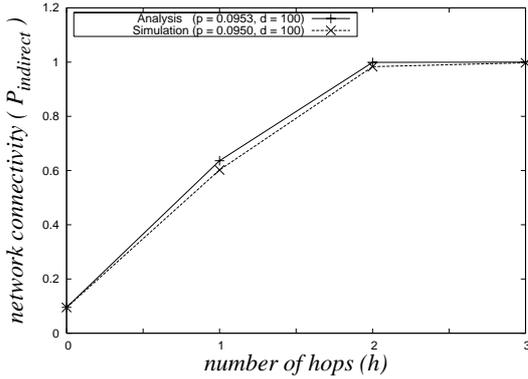


Figure 6. Network connectivity of our scheme under the EG scheme, with $n = 10000$, $d = 100$, $M = 100000$, $m = 100$.

- The number of nodes in the network is $n = 10000$.
- The size of the polynomial pool (in terms of t -degree symmetric bivariate polynomials) is $s = 500$.
- The number of polynomial shares given to each sensor node is $s' = 5$.
- The average number of sensor nodes of each node is $d = 100$.
- The communication range of each sensor node is $\rho = 30$ meters.
- The area A of the deployment field is chosen so that the maximum network size becomes $n = \frac{A \times (d+1)}{\pi \rho^2}$.

The theoretical as well as simulated network connectivity probabilities of our scheme under the polynomial-pool scheme are plotted in Figure 7. It also shows that both the results are closed.

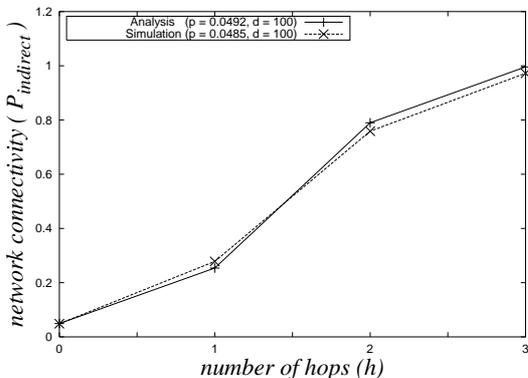


Figure 7. Network connectivity of our scheme under the polynomial-pool scheme, with $n = 10000$, $d = 100$, $s = 500$, $s' = 5$.

VI. PERFORMANCE COMPARISON OF OUR SCHEME WITH THE PATH KEY ESTABLISHMENT

In this section, we compare the performances of our scheme under the EG scheme as well as the polynomial-pool scheme with those of the path key establishment.

A. Computational overhead

We have simulated the computational overhead required by each node for our scheme and the path key establishment phase for h -hop ($h = 1, 2, 3$). The computational overhead is measured by the average number of

encryptions and decryptions carried out by each sensor node during each hop.

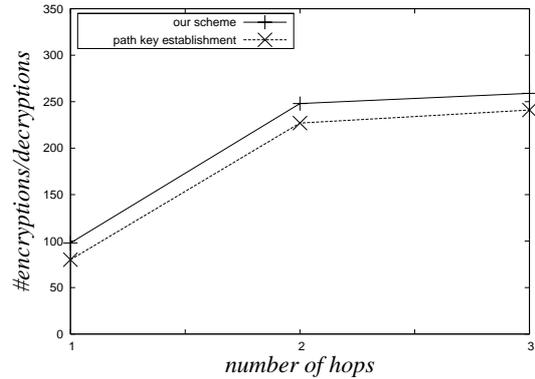


Figure 8. Comparison of computational overhead between our scheme and the path key establishment under the EG scheme. Assume $n = 10000$, $d = 100$, $m = 100$, and $M = 100000$.

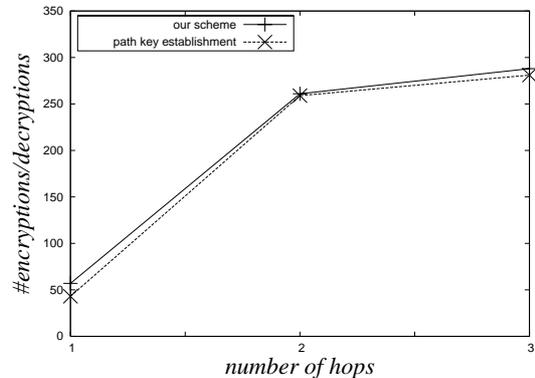


Figure 9. Comparison of computational overhead between our scheme and the path key establishment under the polynomial-pool based scheme. Assume $n = 10000$, $d = 100$, $s' = 5$, and $s = 500$.

Figure 8 shows the comparison of computational overhead per each node between our scheme and the path key establishment under the EG scheme. In this figure, we have considered the initial network connectivity during the direct key establishment phase as $p = 0.0953$, with $m = 100$ and $M = 100000$.

The comparison of computational overhead per each node between our scheme and the path key establishment under the polynomial-pool based scheme is illustrated in Figure 9, considering the initial network connectivity during the direct key establishment phase as $p = 0.0492$, with $s' = 5$ and $s = 500$.

From these figures, we note that our scheme requires some more computational overhead compared to that for the path key establishment phase. However, it is justified by considering better trade-off among network connectivity, resilience against node capture, communication and computational overheads compared to those for the path key establishment phase.

B. Communication overhead

We note that the path key establishment requires a communication overhead proportional to the square of

the number of hops. On the other hand, our scheme also incurs an overhead proportional to the square of the number of hops for establishing a secure path between two nodes u and v plus the communication overhead due to establishment of a new pairwise key between neighbor nodes v and w .

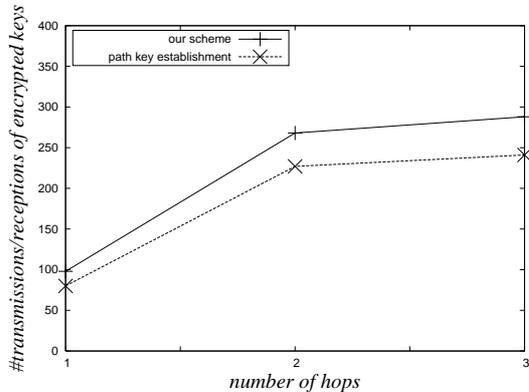


Figure 10. Comparison of communication overhead between our scheme and the path key establishment under the EG scheme. Assume $n = 10000, d = 100, m = 100$, and $M = 100000$.

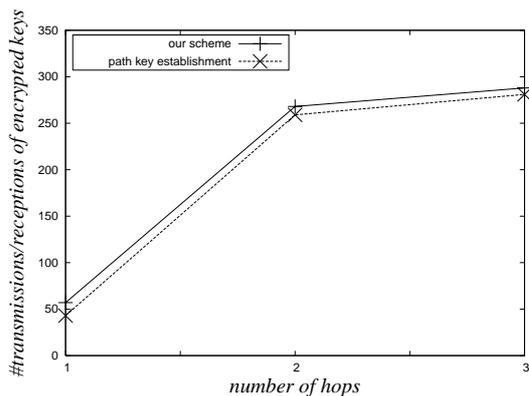


Figure 11. Comparison of communication overhead between our scheme and the path key establishment under the polynomial-pool based scheme. Assume $n = 10000, d = 100, s' = 5$, and $s = 500$.

For simulation of communication overhead, we have measured the communication overhead by the average number of transmissions and receptions of encrypted keys by each sensor node during each hop of our scheme and path key establishment. Figures 10 and 11 show the comparison of communication overhead per each node between our scheme and the path key establishment under the EG scheme and the polynomial-pool based scheme respectively.

C. Network connectivity

We have compared network connectivity of our scheme with the path key establishment under both the EG scheme as well as the polynomial-pool scheme. The simulation results of network connectivity are illustrated in Figures 12 and 13. From these figures, we observe that our scheme improves the network connectivity than the path key establishment under both the EG scheme and the polynomial-pool scheme.

From the theoretical and simulation results we conclude that our scheme yields significantly better connectivity than the path key establishment under both the EG scheme as well as the polynomial-pool scheme.

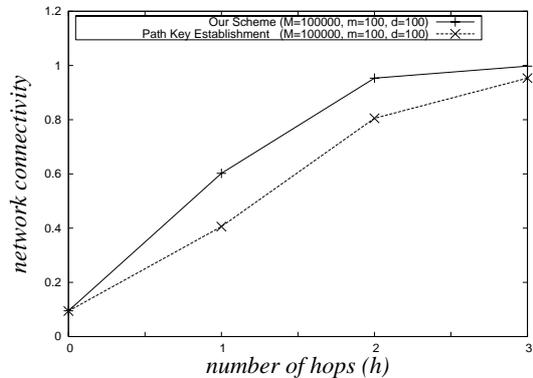


Figure 12. Comparison of network connectivity between our scheme and path key establishment under the EG scheme. Assume $n = 10000, d = 100, M = 100000$, and $m = 100$.

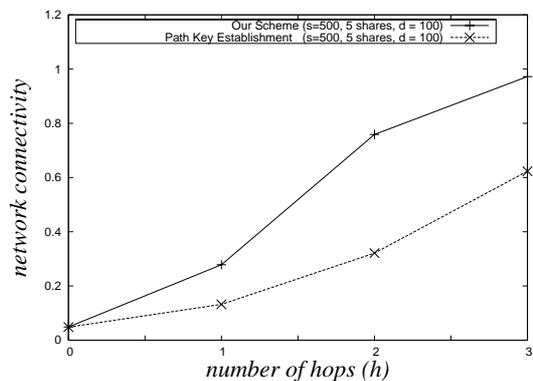


Figure 13. Comparison of network connectivity between our scheme and path key establishment under the polynomial-pool scheme. Assume $n = 10000, d = 100, s = 500$, and $s' = 5$.

D. Resilience against node capture

From the analysis, we observe that the security of our scheme compares favorably with that of the path key establishment for ideal situation if the fraction of the total number of nodes compromised is small. However, it is known that the resilience of the network increases dramatically with the pool size. But, bigger pool sizes lead to lower connectivity. Our scheme addresses this issue as follows. We first start with the parameter values leading to high resilience but poor initial connectivity. Then our scheme subsequently adds to the connectivity by using a few number of hops.

The security of the path key establishment is based on the assumption that the bootstrapping is done securely, that is, no nodes are compromised during the direct key establishment phase. This assumption is true due to the following considerations. In the direct key establishment phase of the bootstrapping, each sensor node only establishes direct pairwise keys with its neighbor nodes in its communication range. As described in [18], due to the

short time period of the direct key establishment phase of the bootstrapping, the sensor nodes can be protected fairly well during this phase; otherwise an adversary could easily compromise all the sensor nodes in a network and then take over the network. In fact, our scheme is also based on this assumption. So, our scheme provides better resilience against node compromise than that for the path key establishment as our scheme ensures better connectivity than the path key establishment.

VII. CONCLUSION

In this paper, we have proposed an improved alternative to the path key establishment phase of the bootstrapping protocol in a sensor network. Our scheme has better trade-off between the communication overhead, the computational overhead, the network connectivity and also the resilience against node compromise than the path key establishment under both the EG scheme as well as the polynomial-pool scheme. Better connectivity lets one start with bigger networks and/or bigger key pool sizes, both leading to better security against node capture. Hence, our proposed scheme would be a more attractive choice than the path key establishment.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : A survey. *Computer Networks*, 38(4):393–422, 2002.
- [3] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *Advances in Cryptology - CRYPTO'92, LNCS 740*, pages 471–486, Berlin, August 1993.
- [4] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages 197–213, Berkeley, California, 2003.
- [5] A. K. Das. A Key Reshuffling Scheme for Wireless Sensor Networks. In *International Conference on Information Systems Security (ICISS 2005), LNCS 3803*, pages 205–216, 2005.
- [6] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.
- [7] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *23rd Conference of the IEEE Communications Society (Infocom'04)*, Hong Kong, China, March 21–25 2004.
- [8] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *ACM Conference on Computer and Communications Security (CCS'03)*, pages 42–51, Washington DC, USA, October 27–31 2003.
- [9] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, July 1985.
- [10] L. Eschenauer and V. D. Gligor. A key management scheme for distributed sensor networks. In *9th ACM Conference on Computer and Communication Security*, pages 41–47, November 2002.
- [11] F. B. Hildebrand. *Introduction to Numerical Analysis*. New York: Dover, second edition, 1974.
- [12] D. Liu and P. Ning. Improving key pre-distribution with deployment knowledge in static sensor networks. *ACM Transactions on Sensor Networks*, 1(2):204–239, 2005.
- [13] D. Liu, P. Ning, and R. Li. Establishing Pairwise Keys in Distributed Sensor Networks. *ACM Transactions on Information and System Security*, 8(1):41–77, 2005.
- [14] R. L. Rivest. The RC5 Encryption Algorithm. In *Proceedings of the second International Workshop on Fast Software Encryption*, volume 1008, pages 86–96, 1994.
- [15] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.
- [16] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, 3rd edition, 2003.
- [17] D. R. Stinson. *Cryptography Theory and Practice*. Chapman & Hall/CRC, third edition, 2006.
- [18] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. *ACM Transactions on Sensor Networks*, 2(4):500–528, November 2006.

Ashok Kumar Das is currently working as Assistant Professor in the International Institute of Information Technology, Bhubaneswar 751 013, India. He has submitted Ph.D. Thesis in the Department of Computer Science and Engineering of the Indian Institute of Technology, Kharagpur 721 302, India, in 2008. He received the M.Tech. degree in Computer Science and Data Processing from the Indian Institute of Technology, Kharagpur, India, in 2000. He also received the M.Sc. degree in Mathematics from the Indian Institute of Technology, Kharagpur, India, in 1998. Prior to join in Ph.D. in 2004, he worked with C-DoT (Centre for Development of Telematics), a premier telecom technology centre of Govt. of India at New Delhi, India from March 2000 to January 2004. During that period he worked there as a Research Engineer on various projects in the fields of SS7 (Signaling System No. 7) protocol stack, GSM (Global System for Mobile Communications) and GPRS (General Packet Radio Services).

He received the INSTITUTE SILVER MEDAL for his first rank in M.Sc. from the Indian Institute of Technology, Kharagpur, India in 1998. He has seventh All India Rank in the Graduate Aptitude Test in Engineering (GATE) Examination in 1998. He received the DIVISIONAL AWARD for his individual excellence in development of SS7 protocol stack from C-DoT, New Delhi, India in 2003. He received a 'Certificate of Special Mention' for the best paper award in the First International Conference on Emerging Applications of Information Technology (EAIT 2006) in 2006 and also a best paper award in the International Workshop on Mobile Systems (WoMS) in 2008. His biography was also selected for inclusion in the 26th Edition of the Marquis Who's Who in the World, USA in 2009. He has 14 publications in international journals/conferences in the area of key distribution mechanisms in wireless sensor networks. His current research interests include cryptography and wireless sensor network security.