An Identity-Based Random Key Pre-Distribution Scheme for Direct Key Establishment to Prevent Attacks in Wireless Sensor Networks

Ashok Kumar Das
Department of Computer Science and Engineering, Indian Institute of Technology
Kharagpur 721 302, India (Email: akdas@cse.iitkgp.ernet.in)

(Received Feb. 8, 2006; revised and accepted May 7, 2006 & June 13, 2006)

Abstract

Key establishment in sensor networks is a challenging problem because of resource constraints of the sensors. The classical public-key routines are impractical in most sensor network architectures. In this paper, we propose a new random key pre-distribution scheme which is based on the identity-based approach for key establishment between two neighbor nodes in wireless sensor networks. Our proposed scheme provides better security against node capture, in particular, against node fabrication attack than the existing random key pre-distribution schemes. Moreover, our scheme has better trade-off between communication overhead, network connectivity and security against node capture compared to the existing random key pre-distribution schemes. In addition, our scheme supports dynamic node addition after initial deployment and also works for any deployment configuration.

Keywords: Direct key establishment, node fabrication attack, random key pre-distribution, sensor networks

1 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) as shown in Figure 1. 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 multihop infrastructure-less architecture through the base station as shown in Figure 1. The base station may communicate with the *task manager node* via Internet or satellite.

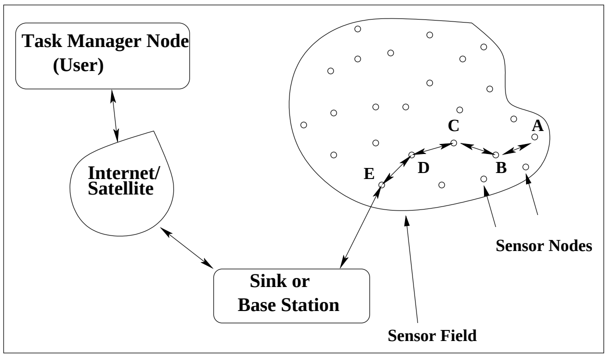

Figure 1: Sensor nodes scattered in a target field

A sensor node is made up of four basic components, as shown in Figure 2: a sensing unit, a processing unit, a transceiver unit, and a power unit. They may also have additional components like a location finding system, power generator, and mobilizer. Sensing units are usually composed of two subunits: sensors and analog-to-digital converters (ADCs). First, the analog data produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC unit, and then fed into the processing unit. A processing unit manages the procedures that make the sensor nodes collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit, which is in general battery powered. In most of the sensor networks, routing techniques and sensing tasks require knowledge of location with high accuracy. Thus, it is common that a sensor node has a location finding system. A mobilizer may sometimes be needed to move sensor nodes when it is required to carry out the assigned tasks.

The topology of sensor networks changes due to the following three phases:

- *Pre-deployment and deployment phase:* Sensor nodes can be deployed from the truck or the plane in the sensor field.
- *Post-deployment phase:* Topology can change after deployment because of irregularities in the sensor field like obstacles or due to jamming, noise, available energy of the nodes, malfunctioning, etc.
- *Redeployment of additional nodes phase:* Additional sensor nodes can be redeployed at any time to replace the faulty or compromised sensor nodes.

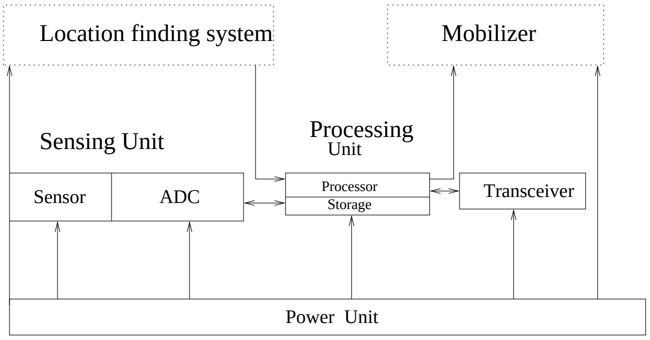

Figure 2: The components of a sensor node

Sensor networks are classified into two categories: master-slave networks and mesh-networks [3]. In a *master-slave network*, the sensor nodes communicate directly with the base station in one hop. For this, the base station allocates predefined time slots for the sensors, and the sensor nodes are allowed only to communicate with the base station during those time slots. On the other hand, the *mesh-networks* are ad hoc networks where the sensor nodes communicate each other to form a multi-hop radio network and then communicate with the base stations. In this paper, we mainly consider the sensor network's structure as the mesh-network.

The following issues make secure communication between sensor networks different from usual (traditional) networks:

- *Limited resources in sensor nodes:* Each sensor node contains a primitive processor featuring very low computing speed and only small amount of programmable memory. An example is the popular Atmel ATmega 128L processor.
- *Limited life-time of sensor nodes:* Each sensor node is battery-powered and is expected to operate for only few days. Therefore, once the deployed sensor nodes expire, it is necessary to add some fresh nodes for continuing the data collection operation. This is referred to as the *dynamic management of security objects (like keys).*
- *Limited communication abilities of sensor nodes:* Sensor nodes have the ability to communicate each other and the base stations by the short range wireless radio transmission at low bandwidth and over small communication ranges (typical example is 30 meters (100 feet)).
- *Lack of knowledge about deployment configuration:* Most of cases the post-deployment network configuration is not possible a priori. As a result, it is unreasonable to use security algorithms that have strong dependence on locations of sensor nodes in a sensor network.
- *Issue of node capture:* A part of the network may be captured by the adversary/enemy. The resilience measurement against node capture is computed by comparing the number of nodes captured, with the fraction of total network communications that are exposed to the adversary *not including* the communications in which the compromised nodes are directly involved.

Thus, it is not feasible to use a public-key cryptosystem such as RSA [20] or Diffie-Hellman key exchange protocol [7] or Elliptic Curve cryptography (ECC) [25] or ElGamal cryptosystem [10] in most resource constrained sensor networks. Hence, a symmetric cipher such as DES/IDEA/RC5/AES [6, 21, 25] is the viable option for encryption/decryption of secret data. But setting up symmetric keys among communication nodes is a challenging task in a sensor network.

A protocol that establishes cryptographically secure communication links among the sensor nodes is called the *bootstrapping protocol*. Several methods [5, 8, 11, 13] are already proposed in order to solve the bootstrapping problem. All these techniques are based on random deployment models, that is, they do not use the pre-deployment knowledge of the deployed sensor nodes. Eschenauer and Gligor [11] 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 keys in the key pre-distribution phase. During the direct key establishment phase, two neighbor nodes can establish a secret key if they share at least one common key. The path key establishment phase is applied if two neighbor nodes fail to establish a secret key in the direct key establishment phase.

Chan et al. [5] proposed the q-composite key predistribution and the random pairwise keys schemes. The q-composite keys scheme first proceeds in a similar manner to the basic random key pre-distribution scheme (the EG scheme). That is, the key set up server picks a set K of random keys out of the total key space, and for each sensor node in the network, selects m random keys from K and stores them into the node's key ring.

Nodes then perform direct key establishment phase with their neighbors. Let each node can identify every neighbor node with which it shares at least q keys. Let the number of actual keys shared be q', where $q' \geq q$. A new communication link key k is then generated as follows: $k = H(k_1 \| k_2 \| \cdots \| k_{q'})$, where H is a one-way hash function (for example, H = SHA-1 [23]) and k_1, k_2,…, $k_{q'}$ are the common keys shared between two neighbor nodes. Thus, key-setup is not performed between nodes that share fewer than q keys.

For both the EG and the q-composite schemes, if a small number of sensors are compromised, it may reveal to compromise a large fraction of pairwise keys shared between uncompromised sensors. The q-composite scheme offers greater resilience against node capture than the EG scheme when the number of nodes captured is small, whereas both the EG and q-composite schemes support arbitrarily large networks. As a result, the q-composite scheme is better than the EG scheme. However, the random pairwise keys scheme is perfectly secure against node captures, but there is a problem in supporting the large network. Liu and Ning's polynomial-pool based key predistribution scheme [13] and the matrix pool-based key predistribution proposed by Du et al. [9] improve security considerably.

The first application of sensor networks was the Sound Surveillance System (SOSUS) [19] which had been used during the cold war in the early 1950s for the purposes to detect as well as track Soviet submarines with the help of acoustic sensors or hydrophones. Other applications are as follows:

- military applications.
- environmental monitoring.
- classroom/home [15, 24].
- health monitoring of patients [22].
- habitat monitoring [4, 15], etc.

A survey on sensor networks could be found in [1, 2].

The remainder of the paper is organized as follows. We introduce a new random key pre-distribution scheme in static sensor networks, which is based on the identity-based technique in Section 2. In Section 3, we provide detailed theoretical analysis of our proposed scheme. In Section 4, we compare the performances of our scheme with those for the existing random key pre-distribution schemes [5, 11, 17]. Finally, Section 5 concludes the paper.

2 The Proposed Scheme

In this section, we discuss the main motivation behind development of our scheme. We also describe various phases of this scheme.

2.1 Motivation

In the basic random key pre-distribution scheme (the EG scheme) [11], keys are assigned to the unique key ids in the key pool $\mathcal{K}$, and thus transmitting only the ids of the keys from the key rings of sensor nodes to establish pairwise secret keys between neighbor sensors does not serve the security in the network. In this case, an attacker may simply record the ids of the keys from all communication channels and later on by capturing few nodes in the network, he may get connect to the uncompromised nodes. As a result, this procedure is not at all secure one. We shall describe in details this issue in sub-section 3.5.3.

Another procedure of key discovery which is more secure, but slower, could utilize client puzzles such as a Merkle puzzle [16]. In this case, each node generates m client puzzles, say, $P_1, \ldots, P_m$, one for each of the m keys in its key ring. Let u and v be two neighbor sensor nodes. u then sends a list $\{E_{k_i}(P_i), MAC_{k_i}(P_i)\}$, $i = 1, 2, \ldots, m$ to node v. Here $E_k(P)$ represents an encrypted puzzle of the puzzle P with the key k and $MAC_k(P)$ the message authentication code for the puzzle P, under the key k. If v is able to solve at least one of the puzzles, then only u and v will share a secret key. To do so, first of all, node v decrypts an encrypted puzzle, say, $E_{k_i}(P_i)$ with one of the keys residing in its key ring. After that, v computes the message authentication code (MAC) for P_i under that key. If the computed MAC and the received MAC are equal, then u and v use this key for future communication. Thus, we see that though this method is secure one, but it requires more communication overhead to establish pairwise keys among neighbor nodes.

As a result, this approach requires more communication overhead in order to establish pairwise keys among neighbor nodes in a sensor network. This problem also pertains to the q-composite scheme proposed by Chan et al. [5] and the polynomial-pool based scheme proposed by Liu and Ning [13].

To overcome two aforementioned problems, we introduce an alternative approach to direct key establishment (the shared key discovery) of random key pre-distribution scheme which is an identity based scheme. In our proposed scheme, we exchange the ids of nodes and also the ids of keys in such a fashion that it remains completely secure from the adversary.

2.2 Notations

We use the following notations for describing our proposed scheme:

- k_{uv} refers to a unique pairwise key shared by two neighbor nodes u and v.
- RN_u: a random nonce generated by the node u. Nonce is a one-time random bit-string, usually used to achieve freshness.
- $u \rightarrow v$: M refers to a message M sent from the node u to node v.

- $MAC_k(M)$: a message authentication code (MAC) for the message M, under the key k.
- id_u: the unique identifier of a sensor node u, that is, a name given to u.
- A||B: data A concatenates with data B.
- $KeyRing_u$: the key ring of a sensor node u.
- key_{id}: the identifier of a key.
- $PRF_k(M)$: output of a pseudo-random function (PRF) applied over the message M using the key k.
- d: the average number of neighbor nodes of a sensor node.
- m: size of the key ring of a sensor node, that is, the number of symmetric keys given to that node.
- M: size of the key pool $\mathcal{K}$.

The different phases for our proposed scheme are as follows.

2.3 Key Pre-Distribution

This phase is done by the (key) setup server in offline. It consists of the following steps.

Step 1. For each sensor node u to be deployed in the sensor network, the (key) setup server assigns a unique identifier id_u.

Step 2. The (key) setup server generates a big key pool $\mathcal{K}$ of size M which consists of randomly generated numbers called the symmetric keys.

Step 3. For each sensor node u, a random subset K_u of size m from the key pool $\mathcal{K}$ is selected. K_u is then loaded in its memory.

Thus, we note that before deploying a sensor node u in the target field, it's key ring $KeyRing_u$ contains only its own identifier id_u and the m symmetric keys selected randomly from the key pool $\mathcal{K}$.

2.4 Direct Key Establishment

For this phase (also called the shared key discovery phase), we use a secret one-way function $f(\cdot)$. It takes the inputs as (i) two neighbor nodes' identifiers and (ii) a key, and finally produces a unique identifier of that key. The function $f(\cdot)$ is programmed into each sensor node's memory. It may be noted that any pseudo-random function which can produce output uniformly distributed in a given range for given inputs can be used. For example, $f(\cdot)$ can be a pseudo-random function (PRF) proposed by Goldreich et al. in 1986 [12].

After deployment of the nodes, each sensor node locates its all neighbors within its communication range. Each sensor node broadcasts its own id to its all neighbors. Assume that u and v be two neighbor nodes. After receiving the ids of each other, nodes u and v compute *"dynamically"* the key ids of the keys residing in their key rings as follows.

```
At node u:
for (∀ key ∈ KeyRing_u) do:
generate $key_{id} := PRF_{key}(id_u||id_v)$;
add $key_{id}$ corresponding to the key in its key ring;

At node v:
for (∀ key ∈ KeyRing_v) do:
generate $key_{id} := PRF_{key}(id_u||id_v)$;
add $key_{id}$ corresponding to the key in its key ring;
```

In order to establish a common key between nodes u and v, they need now only to exchange the key ids just generated by the nodes. If there is a common key id, then the corresponding key is taken as the shared secret key between them. Thus, u and v use this key for their future communication. The key ids are deleted from their key rings as soon as they establish secret keys between them in order to thwart against node capture.

We assume that after key discovery, each node can identify each neighbor node with which it shares at least q keys, where $q \geq 1$. Let the actual number of keys shared be q', where $q' \geq q$. The secret key between two neighbor nodes, say, u and v is computed as

$$k_{uv} = H(id_u||id_v||k_1||k_2||\ldots||k_{q'}), \quad (1)$$

where k_1, k_2, …, $k_{q'}$ are the q' common keys between nodes u and v, H is a secure one-way hash function (for example, H = SHA-1 [23]) and || is a concatenation operation.

We have the following important properties during this process:

1) If a key is same between two nodes' key rings, then the key identifiers generated by the nodes are also same.

2) Since the generation of the key identifiers are being done using PRF function with seeds as a key of a node's key ring and two neighbor nodes' ids, so the key ids for the same key are different for each pair of sensor nodes throughout the network.

3) Our scheme always defines a relationship between the ids of the neighbor nodes and the key ids generated by them.

4) Since the pairwise secret key between two neighbor nodes is always computed using their ids as given in Equation (1), our scheme also defines a relationship between the ids of the neighbor nodes and the secret key generated by them.

Thus, even if an adversary knows the ids of keys for two neighbor nodes, but he could not able to gather much more information in order to fabricate fake nodes to get connect to the network. This is possible because of the above properties.

Communication Steps:
This phase is summarized below. In order to establish a secret pairwise key between two neighbor nodes, say, u and v, the following messages are to be exchanged between them.

1) u transmits its own id to v: $u \rightarrow v : id_u$.

2) v transmits its own id to u: $v \rightarrow u : id_v$.

3) u generates a random nonce RN_u. Then, it sends this nonce RN_u, a list of the already generated key ids and its own id as well as the id of v to node v: $u \rightarrow v : \{\text{list of generated } key_{id} \text{ s}\}||id_u||id_v||RN_u$.

4) v also generates a random nonce RN_v. Then, it sends RN_v, a list of the already generated key ids and its own id as well as the id of u to node u: $v \rightarrow u : \{\text{list of generated } key_{id} \text{ s}\}||id_u||id_v||RN_v$.

5) Assume that after exchanging the key ids, u computes the secret pairwise key k_{uv} shared with its neighbor node v as computed in Equation (1). u sends a message to v which consists of its own id as well as the id of v, the random nonce RN_v of node v and a message authentication code (MAC) of these fields, under the computed key k_{uv} as follows: $u \rightarrow v : (id_u||id_v||RN_v)||MAC_{k_{uv}}(id_u||id_v||RN_v)$.

   Sending of the random nonce RN_v of v ensures the transaction uniquely between the nodes u and v.

6) v computes the secret pairwise key k_{uv} shared with its neighbor node u as computed in Equation (1). v sends a message to u which consists of its own id as well as the id of u, the random nonce RN_u of node u and a message authentication code (MAC) of these fields, under the computed key k_{uv} as follows: $v \rightarrow u : (id_u||id_v||RN_u)||MAC_{k_{uv}}(id_u||id_v||RN_u)$.

After receiving the last message by the nodes u and v, they perform one MAC verification on that message. If the MAC verification is successful, they store the key k_{uv} for their future communications. Thus, we see that node u needs to follow only Steps 1, 3 and 5 in order to establish a secret pairwise key shared with the node v.

2.5 Addition of Sensor Nodes

Sometimes nodes may be faulty due to battery-energy consumption problem, malfunctioning, etc. or compromised due to capturing of some nodes by the adversary. Therefore, it is necessary to redeploy some new sensor nodes to replace those faulty or compromised sensor nodes in the sensor network.

To add a new sensor node u, the (key) setup server first picks a random subset K_u from the key pool $\mathcal{K}$ and loads this key ring K_u in u's memory. The (key) setup server also assigns a unique identifier id_u for the node u. After deployment, the sensor node u performs the direct key establishment phase in order to establish pairwise keys with its neighbor sensor nodes.

As a result, we see that it is easy to add new sensor nodes after initial deployment of the sensor nodes in the network.

2.6 Path Key Establishment

This is an optional phase, and if executed, adds to the connectivity of the network. This phase is applied after the direct key establishment phase.

Assume that two physical neighbors, say, u and v fail to establish a pairwise key between them in the direct key establishment phase, but there exists a secure h-hop $u-v$ path, say, $\langle u = u_0, u_1, \ldots, u_h, u_{h+1} = v\rangle$ such that each (u_i, u_{i+1}) is a secure link, for $i = 0, 1, \ldots, h$. Sensor node u proceeds as follows:

1) u generates a random number k' as the shared secret key between u and v. u encrypts k' using the key shared between u and u_1, and transmits the encrypted key to u_1.

2) u_1 retrieves k' by decrypting the received encrypted key using the key shared between u and u_1. u_1 encrypts k' using the key shared between u_1 and u_2, and sends to u_2.

3) This process is continued until the key k' reaches to the desired destination node v.

Finally, u and v store this key k' and use it for their future communication. We observe that if the number of hops of the path is increased, then the communication overhead also increases. To reduce the communication overhead, the number h of hops of the path is restricted to a small value, say, 2 or 3.

3 Analysis

In this section, we describe network connectivity of our proposed scheme. We analyze the storage overhead, computational overhead and communication overhead required for our proposed scheme. We also analyze various security aspects of our scheme against node capture attacks.

3.1 Network Connectivity

3.1.1 Probability of Establishing Direct Keys between Neighbors

We observe from the direct key establishment phase of our proposed scheme that the key setup is not performed between two neighbor nodes that share fewer than q keys, where $q \geq 1$. Let $p_{connect}$ denote the probability of two neighbor nodes sharing sufficient keys ($\geq q$

keys) to form a secure connection. If M and m be the key pool size and the key ring size respectively, then $p_{connect} = 1-$(probability that two nodes share insufficient keys ($< q$ keys) to form a connection). Thus, we have,

$$\begin{aligned} p_{connect} &= 1 - \frac{\binom{M-m}{m}}{\binom{M}{m}} \\ &= 1 - \prod_{i=0}^{m-1} \frac{M-m-i}{M-i}, \quad \text{if } q = 1. \end{aligned} \qquad (2)$$

and

$$p_{connect} = 1 - \sum_{i=0}^{q-1} p_i, \quad \text{if } q \geq 2, \qquad (3)$$

where p_i is the probability that any two nodes have exactly i keys in common from their key rings and

$$p_i = \frac{\binom{M}{m}\binom{m}{i}\binom{M-m}{m-i}}{\binom{M}{m}^2}. \qquad (4)$$

Let us further simplify the quantity p_i in Equation (4). We use the Stirling's Formula [18] for computing the factorial of a large positive integer n. The Stirling's formula is given by $n! \approx \sqrt{2\pi}\, n^{n+\frac{1}{2}}\, e^{-n}$, which can be further simplified as $n! \approx \sqrt{2\pi}\, e^{[-n+(n+\frac{1}{2})\ \ln(n)]}$. Thus, from the Equation (4) we have:

$$\begin{aligned} p_i &= \frac{(m!)^2\,((M-m)!)^2}{i!\,((m-i)!)^2\,(M-2m+i)!\,M!} \\ &\approx exp[(2M-2m+1)\ \ln(M-m) - (M-2m \\ &\quad +\frac{1}{2})\ln(M-2m) - (M+\frac{1}{2})\ \ln(M)], \quad \text{if } i = 0 \\ &\approx \frac{1}{\sqrt{2\pi}}\, exp[(2m+1)\ \ln(m) + (2M-2m \\ &\quad +1)\ \ln(M-m) - (i+\frac{1}{2})\ \ln(i) - (2m-2i \\ &\quad +1)\ \ln(m-i) - (M-2m+i+\frac{1}{2})\ln(M \\ &\quad -2m+i) - (M+\frac{1}{2})\ \ln(M)], \quad \text{if } i > 0. \end{aligned}$$

Thus, given the parameter values M, m and q, we can set up the network connectivity $p_{connect}$.

The direct network connectivity probabilities for our scheme are plotted for various values of M = 10000, 20000,30000, 40000, 50000, 60000, 70000, 80000, m = 200, and $q = 1, 2, 3$ in Figure 3. We see from this figure that to gain the better connectivity the key pool size M should be chosen smaller.

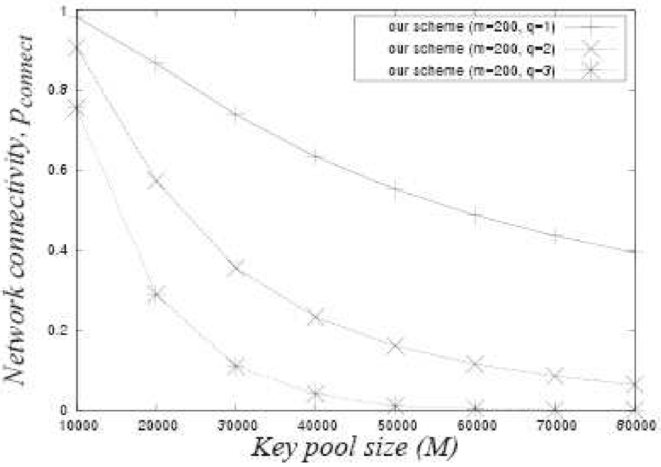

Figure 3: Direct network connectivity $p_{connect}$ of our scheme, with M = 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, m = 200, and q = 1, 2, 3

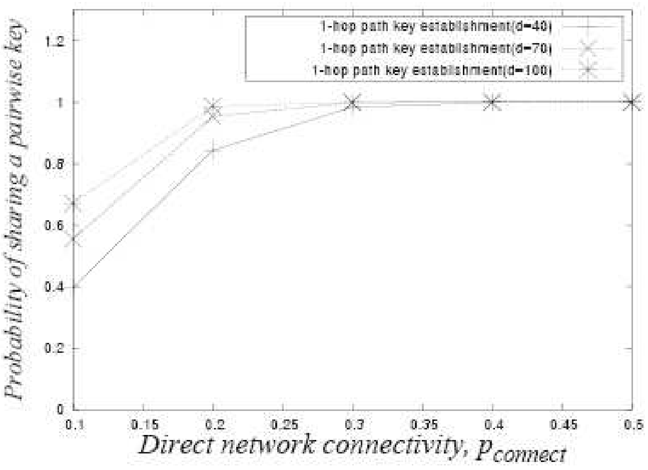

Figure 4: The probability P_s of establishing a pairwise key v.s. the probability $p_{connect}$ that two sensor nodes establish a direct pairwise key, with $d = 40, 70, 100$

3.1.2 Probability of Establishing Keys Using 1-hop Path Key Establishment between Neighbors

Let d be the average number of nodes that each node can contact. Let us consider 1-hop path: $\langle u, u_1, v\rangle$ where u and v be neighbor nodes not sharing any keys currently. The probability that the intermediate node u_1 shares a pairwise key with both the source node u and the destination node v is $p^2_{connect}$. As long as one of the d nodes act as an intermediate node, nodes u and v establish a pairwise secret key. Hence, the probability that two sensor nodes establish a pairwise key using 1-hop path (directly or indirectly) is

$$P_s = 1 - (1-p_{connect})(1-p^2_{connect})^d.$$

The network connectivity probabilities for 1-hop path key establishment are plotted in Figure 4 for various values of

d. It is clear from the figure that one can achieve significantly better connectivity after executing this phase even if the network is initially disconnected with high probability.

3.2 Storage Overhead

We see that a sensor node is given only m keys in its key ring before deployment in key pre-distribution phase by the (key) setup server. Though the key ids are generated by the sensor nodes during the shared key discovery phase, they are deleted from memory as soon as nodes establish pairwise keys between them. As a result, the storage overhead is due to storage of m keys only. In the random schemes [5, 11], each sensor node has to store m keys and the corresponding key ids in its memory. Thus, the storage overhead for our scheme is less than that of the random schemes [5, 11].

3.3 Computational Overhead

For each pair of neighbor nodes, the nodes need to generate the key ids each time by invoking m efficient PRF operations. If d be the average number of physical neighbors of a sensor node, then the node needs to generate the key ids for all d neighbors. Thus, the computational overhead is due to $m \times d$ efficient PRF operations.

Zhu et al. [26] pointed out that due to computational efficiency of pseudo-random function (PRF), the computational overhead is negligible. Hence, we see that though our scheme requires $m \times d$ PRF operations, due to computational efficiency of PRF function the actual computational overhead is low. As a result, our scheme provides a good trade-off for the resource-constrained sensor networks as compared to the existing random schemes [5, 11, 17].

3.4 Communication Overhead

In order to establish a secret key between two neighbor nodes, they need only to exchange the generated key ids residing in their key rings. Thus, the communication overhead is mainly due to the transmission of the key ids residing in a sensor's key ring.

3.5 Security Analysis

In this section, we analyze the security under random node capture attack, selective node capture attack as well as active attack such as node fabrication attack.

3.5.1 Random Node Capture Attack

In the basic random key pre-distribution scheme [11] and the q-composite scheme [5], the security of sensor networks are analyzed on the basis of fraction of the communication links compromised due to captured sensor nodes. In those schemes, the resilience against node capture is measured on the basis of random node capture of sensors. The resilience figures against random node capture for those schemes show that the q-composite scheme is better than the basic random key pre-distribution scheme when the number of nodes captured is small. Since our scheme is closed to the q-composite scheme, so the resilience against random node capture remains same as the q-composite scheme. Thus, if the number of nodes captured is small, our proposed scheme provides better security against random node capture compared to the basic random key pre-distribution scheme.

3.5.2 Selective Node Capture Attack

In the existing random key pre-distribution schemes such as [5, 11], the sensors are captured randomly. However, in practice, the random node capture assumption is too weak. Hence, an attacker can selectively capture sensors from certain pockets of the target field instead of individual nodes randomly over the target field. For the basic random scheme (the EG scheme) [11], in the best case for the attacker, for a key pool of size M and the key ring of size m of each sensor node, the attacker can compromise all communication links by capturing $\lceil \frac{M}{m} \rceil$ sensor nodes in the network. As a result, an attacker can inspect all keys possessed by captured sensors and then find the minimal cover set which contains the minimal number of sensors that can cover the maximum number of keys in the key pool. However, due to the purely random selection of keys in [5, 11], the attacker does not able to gain significantly more information using the selective node capture attack as compared to the random node capture attack. As our proposed scheme is closest to the random q-composite scheme [5], so the gain due to selective node capture attack over random node capture attack is not significant in our scheme also.

3.5.3 Node Fabrication Attack

In this kind of active attack, an attacker can capture some nodes in the network and then fabricate some fake nodes using the information gathered from the captured nodes. Due to lack of *a-priori* knowledge of post-deployment configuration, the fake nodes are not detected as anomalous sensor nodes by the uncompromised sensors in the other parts of the sensor network.

Mehta et al. proposed a scheme called RINK-RKP [17] for key pre-distribution and shared key discovery to prevent active attacks. They showed that their scheme significantly improves the security against node fabrication attack compared to the existing schemes [5, 11].

Our security analysis against node fabrication attack is also similar to that of [17]. Now, in our scheme, in order for a fabricated node to get connect to the network via an uncompromised node, the node needs to satisfy the following two conditions:

1) The fabricated node should share at least q number of keys with the uncompromised node.

2) Given the first condition is true, all the shared secret pairwise keys must be already known to the attacker.

To satisfy the first condition, the security of our scheme depends on the security of the PRF function [12]. In order to share q keys with the uncompromised node, the fabricated node must compute the ids of the keys residing in its key ring to get connect to the network.

Let $P_f(c)$ denote the probability that a fabricated node will satisfy the above two conditions with c number of nodes already captured by the attacker. The probability $P_f(c)$ can be computed as:

$$P_f(c) = \frac{1}{p_{connect}} \sum_{i=q}^{m} p_i \times \left[1 - \left(1 - \frac{m}{M}\right)^c\right]^i$$

where $[1 - (1 - \frac{m}{M})^c]$ is the fraction of keys that are compromised due to capture of c sensor nodes, m the key ring size of a sensor node and M the key pool size. The probabilities $p_{connect}$ and p_i are given in the Equations (2), (3) and (4).

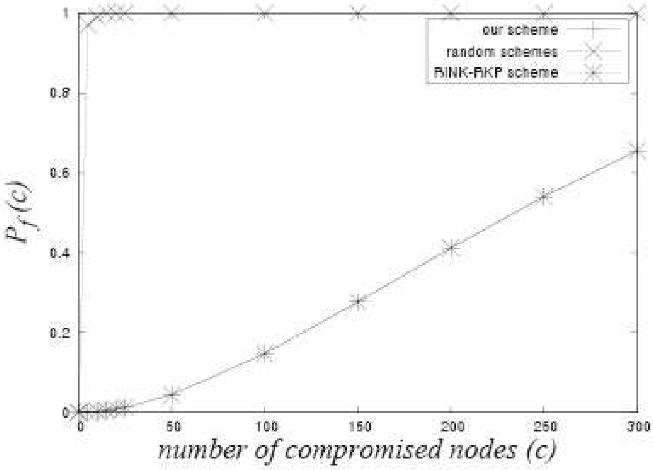

Figure 5: Node fabrication attack, with $M = 40000$, $m = 200$, and $q = 2$ so that $p_{connect} = 0.2329$

In the existing random schemes [5, 11], the attacker can easily fabricate fake nodes with identity of his choice with the same set of key informations of the captured nodes. This is possible because in those schemes there is no defined relationship between the node id and the ids of the keys possessed by each sensor node. But, in our scheme, there is always a relationship between the node id and the ids of the keys generated by each sensor.

Mehta et al. analyzed in their scheme RINK-RKP that in the existing random key pre-distribution schemes such as [5, 11] by capturing only two nodes, an attacker can fabricate and deploy approximately $\begin{pmatrix} 2m \\ m \end{pmatrix}$ fake nodes. Since these fake nodes contain valid key informations, so they can not be detected by the network.

From the Figure 5, we note that our scheme significantly improves the security against node fabrication attack compared to the existing random key predistribution schemes such as the basic random scheme [11] and the q-composite scheme [5]. However, the security of the RINK-RKP scheme [17] remains same as that of our scheme under the node fabrication attack.

4 Comparison with Previous Schemes

In this section, we compare the performances of our proposed scheme with those for the EG scheme [11], the q-composite scheme [5] and the RINK-RKP scheme [17].

Table 1 illustrates the comparison of the existing random key pre-distribution schemes [5, 11, 17] with our proposed scheme with respect to the generation of the key pool, selection of keys from that pool, the relationship between the node ids and the ids of keys residing in each sensor's key ring and the shared key discovery (direct key establishment) procedure. This table shows that for the existing schemes and our scheme, the keys are selected randomly without replacement (RWR) from the big key pool $\mathcal{K}$. Thus, there may be overlap of keys between the key rings of the nodes in the network. For the EG scheme, q-composite scheme as well as RINK-RKP scheme, the same key may be shared between several neighbor nodes in the network. On the other hand, our proposed scheme always guarantees that the key shared by any two neighbor nodes in the network is distinct because that secret key between neighbors is derived using the ids of the neighbor nodes as well as the common keys between the nodes (as derived in Equation (1)).

For the EG and q-composite schemes, there is no defined relationship between the nodes' ids and the ids of the keys possessed by the nodes. Both the RINK-RKP scheme as well as our scheme define a relationship between the node ids and the ids of the keys generated by the nodes. Due to this relationship, we see from the Figure 5 that our proposed scheme has a significant improvement than the EG scheme and q-composite scheme from a serious attack like node fabrication attack. However, the RINK-RKP has the same security compared to that for our scheme.

In the EG and q-composite schemes, both the cleartext broadcasting (CB) and the private shared-key discovery (PSD) procedures are proposed during the shared key discovery (direct key establishment) phase. In cleartext broadcasting, in the EG and q-composite schemes, each node broadcasts its own id and a list of key ids from its key ring. On the other hand, in private shared-key discovery, for each key residing in the key ring of a node, each node broadcasts a list of challenge messages as described in Section 2.1. For secure key establishment procedure both the EG and q-composite schemes need to involve the PSD procedure. However, our proposed scheme and the RINK-RKP do not require to involve the PSD procedure for establishing secret pairwise keys between

Table 1: Comparison of our scheme with the existing random key pre-distribution schemes

Schemes $\Rightarrow$ *Items* $\Downarrow$	*EG scheme [11]*	*q-composite [5]*	*RINK-RKP scheme [17]*	*Our scheme*
Key Pool ($\mathcal{K}$)	unstructured	unstructured	unstructured	unstructured
Key selection (m)	RWR	RWR	RWR	RWR
Node id & key id relation	not defined	not defined	defined	defined
Shared-key discovery	CB/PSD	CB/PSD	CB	CB

RWR: random without replacement

CB/PSD: clear-text broadcasting, private shared-key discovery

Table 2: Comparison of the performances of our scheme with those for the existing random key pre-distribution schemes (the EG, q-composite and RINK-RKP schemes)

Schemes $\Rightarrow$ *Items* $\Downarrow$	*EG scheme [11]*	*q-composite [5]*	*RINK-RKP [17]*	*Our scheme*
Storage overhead	m keys + m key ids	m keys + m key ids	m keys + m key ids	m keys only
Communication overhead	list of key ids (in CB) or list of challenge messages (in PSD) + node's own id + response	same as EG scheme	node's own id + response	list of generated key ids + node's own id + response
Computational overhead	$2d$ encryptions /decryptions for responses + $2(m \times d)$ encryptions /decryptions (in PSD)	same as EG scheme + d hash operations	$m \times d$ hash operations + $d \times q$ XOR operations + $2d$ encryptions /decryptions for responses	$m \times d$ PRF operations + d hash operations + $2d$ MAC operations for responses
Network connectivity	$p_{connect}$ $(q = 1)$	$p_{connect}$ $(q \geq 2)$	$p_{connect}$ $(q \geq 2)$	$p_{connect}$ $(q \geq 1)$
Resilience against random node capture	poor	better than EG scheme	better than EG scheme	better than EG scheme
Resilience against selective node capture	not significant	not significant	not significant	not significant
Resilience against node fabrication attack	very poor	very poor	high	high

neighbor nodes. As a result, we notice that our proposed scheme provides efficient mechanism to establish secret pairwise keys between neighbors compared to the EG and q-composite schemes.

Table 2 shows the comparison of storage requirement, computational overhead, communication overhead, network connectivity and resilience against node capture compared to those for the EG, q-composite and RINK-RKP schemes.

From the Table 2, it is very clear that our scheme requires less storage overhead compared to that for the existing EG, q-composite and RINK-RKP schemes. We note from this table that if the PSD procedure is applied for the EG and q-composite schemes to make the shared key discovery secure, they require more communication overhead than the RINK-RKP and our scheme. We also observe that due to computational efficiency of the PRF functions, the overall computational overhead is less than that for the RINK-RKP scheme. Moreover, if the shared key discovery of the EG and q-composite schemes is applied under PSD procedure, our scheme requires less computational overhead compared to that for those schemes. On the other hand, if the shared key discovery of the EG and q-composite schemes is applied under CB procedure, the computational overhead of our scheme is also comparable with that for those schemes.

From the Table 2, it is also clear that the network connectivity of our scheme remains same as the EG scheme, because both schemes require at least one common key to be shared between two neighbor nodes in order to establish a secret key shared by those nodes. On the other hand, our scheme provides significantly better network connectivity compared to that for the q-composite as well as RINK-RKP schemes. Overall, we conclude our scheme is scalable and efficient in storage, communication and computation.

Let us now compare the security of our scheme with the EG, q-composite and RINK-RKP schemes. We observe from this table that the resilience against random node capture of our scheme is better than the EG scheme, while it has the same security level as the q-composite and RINK-RKP schemes. The resilience against node fabrication attack is very poor for the EG and q-composite schemes compared to the RINK-RKP and our scheme. As a result, our scheme has better security than the EG and q-composite schemes, whereas the security of our scheme retains same as that for the RINK-RKP scheme.

5 Conclusion

In this paper, we propose a new scheme for random key pre-distribution in sensor networks. Our scheme has better trade-off between communication overhead, network connectivity and security against node capture compared to the existing random key pre-distribution schemes. In the existing schemes, the security analysis is based on random capture of sensor nodes. We have analyzed the security based on selective node capture attack. Due to the relationship between the node id and the ids of keys generated by each sensor node, our scheme provides significantly better security against node fabrication attack compared to the existing schemes. Moreover, our scheme supports dynamic node addition after initial deployment of the nodes in the deployment area and also works for any deployment topology.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A Survey,” *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[3] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd, 2002.

[4] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, “Habitat Monitoring: Application Driver for Wireless Communications Technology,” in *Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, pp. 20-41, Apr. 2001.

[5] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” in *IEEE Symposium on Security and Privacy*, pp. 197-213, 2003.

[6] J. Daemen and V. Rijmen, *AES proposal: Rijndael*, AES Round 1 Technical Evaluation, CD-1: Documentation, NIST, Aug. 1998.

[7] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, 1976.

[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)*, pp. 42-51, Oct. 2003.

[9] 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)*, pp. 586-597, Mar. 2004.

[10] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, pp. 469-472, July 1985.

[11] L. Eschenauer and V. D. Gligor, “A Key Management Scheme for Distributed Sensor Networks” in *the 9th ACM Conference on Computer and Communication Security*, pp. 41-47, Nov. 2002.

[12] O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions,” *Journal of the ACM*, vol. 33, no. 4, pp. 792-807, Oct. 1986.

[13] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," in *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 52-61, Oct. 2003.

[14] D. Liu and P. Ning, "Improving key pre-distribution with deployment knowledge in static sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 2, pp. 204-239, 2005.

[15] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Network for Habitat Monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)* pp. 88-97, Sep. 2002.

[16] R. Merkle, "Secure communication over insecure channels," *Communications of the ACM*, vol. 21, no. 4, pp. 294-299, 1978.

[17] M. Mehta, D. Huang, and L. Harn, "RINK-RKP: A scheme for key predistribution and shared-Key discovery in sensor networks," in *The 24th IEEE International Performance Computing and Communications Conference (IPCCC'05)*, pp. 193-197, 2005.

[18] J. M. Patin, "A very short proof of Stirling's formula," *Amer. Math. Monthly*, vol. 96, pp. 41-42, 1989.

[19] J. Pike, *Sound Surveillance System (SOSUS)*, Nov. 2002. (http://www.globalsecurity.org/intell/systems/sosus.htm)

[20] R. L. Rivest, A.Shamir, and L.M.Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[21] R. Rivest, "The RC5 algorithm," *Dr. Dobb's Journal*, vol. 20, no. 1, pp. 146-148, Jan. 1995.

[22] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research Challenges in Wireless Networks of Biomedical sensors," in *ACM SIGMOBILE*, pp. 151-165, 2001.

[23] *Secure Hash Standard*, NIST, U. S. Department of Commerce, Apr. 1995.

[24] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart development problem-solving environments," in *ACM SIGMOBILE*, pp. 132-138, 2001.

[25] W. Stallings, *Cryptography and Network Security: Principles and Practices*, Prentice Hall, 3rd, 2003.

[26] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 62-72, Oct. 2003.

Ashok Kumar Das received the M.Sc. degree in Mathematics from Indian Institute of Technology, Kharagpur 721 302, India, in 1998. He also received the M.Tech. degree in Computer Science from Indian Institute of Technology, Kharagpur 721 302, India, in 2000. He is currently pursuing his Ph.D. degree in Computer Science and Engineering from Indian Institute of Technology, Kharagpur 721 302, India. Prior to join in Ph.D., 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 important projects in the fields of SS7 (Signaling System No. 7) protocol stack, GSM and GPRS. His current research interests include cryptography, information security, network security, and wireless sensor network security.