Improving Identity-based Random Key Establishment Scheme for Large-scale Hierarchical Wireless Sensor Networks*

Ashok Kumar Das

Center for Security, Theory and Algorithmic Research, International Institute of Information Technology
Gachibowli, Hyderabad 500 032, India (Email: iitkgp.akdas@gmail.com)

Abstract

In this paper, we propose a novel identity-based random key pre-distribution scheme called the *identity based key pre-distribution using a pseudo random function* (IBPRF), which has better trade-off between communication overhead, network connectivity and resilience against node capture compared to the other existing key pre-distribution schemes. IBPRF always guarantees that no matter how many sensor nodes are captured, the secret communication between non-compromised sensor nodes are still secure. We then propose an improved version of our scheme in a large-scale hierarchical wireless sensor network. This improved approach has better trade off among network connectivity, security, communication, computational and storage overheads, and scalability than the existing random key pre-distribution schemes. The strength of the proposed IBPRF scheme and its improved approach is establishing pairwise secret keys between neighboring nodes with scantling communication and computational overheads. The improved IBPRF approach further supports a large-scale sensor network for the network connectivity. Through the analysis we show that the improved IBPRF scheme provides better security and lower overheads than other existing schemes.

Keywords: Identity-based key pre-distribution, key management, large-scale hierarchical networks, wireless sensor network

1 Introduction

In a distributed wireless sensor network (DWSN), many tiny computing nodes called sensors, are scattered in an area for 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 computationally well-equipped whereas the sensor nodes are resource-starved. Such networks are used in many applications including tracking of objects in an enemy's area for military purposes, distributed seismic measurements, pollution tracking, monitoring fire and nuclear power plants, tracking patients, engineering and medical explorations like wildlife monitoring, etc. A survey on sensor networks can be found in [1].

Data collected by sensor nodes need be encrypted before transmitting to neighboring nodes and base stations. In order to protect the sensing data and the sensor readings, symmetric cryptographic secret keys should be used to encrypt the exchanged messages between communicating nodes in the network. Due to resource limitations as well as vulnerability to physical capture of nodes, traditional public key security protocols (such as RSA [33], Diffie-Hellman key exchange protocol [14], Elliptic Curve cryptography (ECC) [34, 35], ElGamal cryptosystem [16]) are too complicated and energy-consuming for large-scale wireless sensor networks. Moreover, trusted third-party authentication schemes (e.g., Kerberos [24]) are also infeasible due to the unpredictable network topology, short radio transmission range and the intermittent operations of wireless sensors. As a result, it is not viable to use public-key cryptosystems in most resource constrained wireless sensor networks. Hence, the symmetric cipher such as DES/IDEA/RC5 [34, 35] 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.

The wireless nature of communication among the sensors make sensor networks vulnerable to passive and active attacks. For many applications, the low cost sensors are often deployed in unattended target field which make them physically insecure. The sensors are not considered as tamper-proof devices because of their low-cost design issue. Thus, one of the goals is to design a secure scheme for pairwise key establishment to minimize the effect of physical node capture in sensor networks.

The following issues make secure communication between sensor networks different from 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.

- *Limited life-time of sensor nodes:* Each sensor node is battery-powered. Once the deployed sensor nodes expire, it is necessary to deploy some fresh nodes for continuing the data collection operation.

- *Limited communication abilities of sensor nodes:* Sensor nodes have ability to communicate each other and the base stations by the short range wireless radio transmission at low bandwidth and over small communication ranges.

- *Lack of knowledge about deployment configuration:* In most of the sensor networks applications, the post deployment network configuration is not known 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.

- *Mobility of sensor nodes:* Sensor nodes may be mobile or static. If sensor nodes are mobile, they can change the network configuration at any time.

- *Issue of node capture:* A part of the network may be captured by the adversary. 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.

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., or due to the mobile sensor nodes in the network.

- *Redeployment of additional nodes phase:* Additional sensor nodes can be redeployed at any time to replace the faulty or compromised sensor nodes.

A protocol that establishes cryptographically secure communication links among the sensor nodes is called the *bootstrapping protocol*. Several key management schemes have been proposed for sensor networks (see [3, 4, 37] for surveys of this field), but most existing schemes are not scalable or vulnerable to a small number of captured nodes. Some methods [5, 6, 9, 10, 15, 17, 18, 26, 27, 32, 36, 38, 39] are already proposed in order to solve the bootstrapping problem. Eschenauer and Gligor [18] 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. [6] proposed the q-composite key predistribution and the random pairwise keys schemes. For both the EG and the q-composite schemes, if a small number of sensors are compromised, they may reveal to compromise 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 in supporting the large network size. Liu and Ning's polynomial-pool based key predistribution scheme [27] and the matrix-based key predistribution proposed by Du et al. [15] improve security considerably as compared to that for the EG scheme and the q-composite scheme. Liu and Ning proposed an extended version [25] of the closest pairwise keys scheme [25] for distributed static sensor networks. Their scheme is based on the pre-deployment locations of the deployed sensor nodes and a pseudo random function (PRF) proposed by Goldreich et al. [19]. There is no communication overhead for establishing direct pairwise keys between neighbor nodes and the scheme is perfectly secure against node capture.

The rest of the paper is organized as follows. In Section 2, we discuss the network models in wireless sensor networks. Section 3 gives a brief overview of some existing random key pre-distribution schemes. Section 4 introduces our identity based random key pre-distribution scheme called the identity based key predistribution using a pseudo random function (IBPRF) in static sensor networks. In this section, we provide a theoretical analysis for security and performances of our scheme and compare the performances of our scheme with the existing schemes. In Section 5, we provide an improved version of our basic scheme (IBPRF) for a large-scale hierarchical wireless sensor network. In this section, we discuss the security aspects and performances of this improved scheme and we also compare our improved scheme with the existing related schemes. Finally, Section 6 concludes the paper.

2 Network Models

Basically, there are two types of WSN architectures available for wireless sensor networks. One is the hierarchical architecture and the other is the distributed flat architecture.

Hierarchical wireless sensor networks: A *hierarchical wireless sensor network (HWSN)* is shown in Figure 1. From this figure, we see that there is a hierarchy among the nodes based on their capabilities: base stations, cluster heads and sensor nodes. *Sensor nodes* are inexpensive, limited capability and generic wireless devices. Each sensor has limited battery power, memory size and data

processing capability and short radio transmission range. We assume that after deployment of the sensor nodes, they become static. Sensor nodes form a cluster, communicate each other in that cluster and finally communicate with the cluster head (CH). We further assume that communication between sensor nodes in a cluster exists. *Cluster heads* have more resources than sensors. They are equipped with high power batteries, large memory storage, powerful antenna and data processing capabilities. Cluster heads can execute relatively more complicated numerical operations than sensors and have much larger radio transmission range than sensor nodes. Cluster heads can communicate with each other directly and relay data between its cluster members and the base station. For example, the cluster heads can be PDAs and the sensor nodes are the MICA2-DOT motes [21]. A *base station or sink node* (BS) is typically a gateway to another network, a powerful data processing/storage center, or an access point for human interface. A base station collects sensor readings, perform costly operations on behalf of sensor nodes and manage the network. In some applications, the base station is assumed to be trusted and tamper resistant. Thus, the base station is used as key distribution center (KDC). Sensor nodes are deployed around one or more hop neighborhood of the base station. The base station can reach all the sensor nodes in a network. Depending on the applications, the base station (BS) can be located either in the center or at a corner of the network. Data flow in such networks can be: (1) pairwise (unicast) among sensor nodes, (2) group-wise (multicast) within a cluster of sensor nodes, and (3) network-wise (broadcast) from base station to sensor nodes.
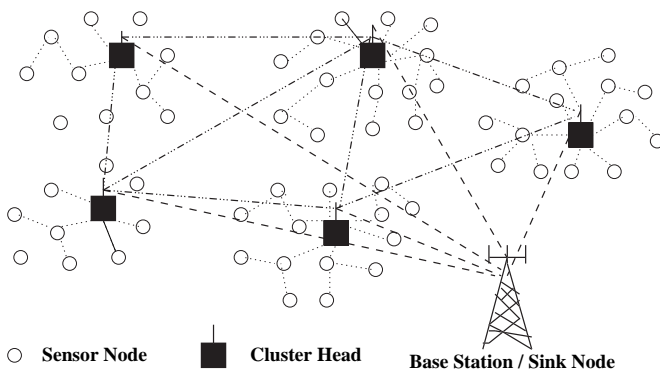

Figure 1: A hierarchical wireless sensor network (HWSN) architecture.

Distributed wireless sensor networks: A *distributed wireless sensor network (DWSN)* is shown in Figure 2. There is no fixed infrastructure and network topology is not known prior to deployment of the sensor nodes in the target field. Sensor nodes are usually deployed all over the target area randomly. After deployment sensor nodes form an infrastructure-less multi-hop wireless communication between them and data is routed back to the base station. Data flow in DWSN is similar to data flow in HWSN with a difference that network-wise (broadcast)

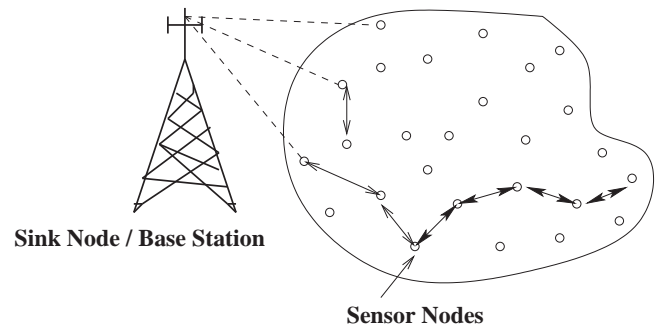flow takes place by every sensor node in the network.

Figure 2: A distributed wireless sensor network (DWSN) architecture.

3 Overview of Existing Random Key Pre-distribution Schemes

In this section, we briefly describe the following existing random key pre-distribution schemes in distributed wireless sensor networks.

Eschenauer and Gligor first proposed a random key predistribution scheme (henceforth referred to as the EG scheme) [18]. Before the sensor nodes are deployed, a key predistribution phase is performed by the key setup server in offline. In this phase, a large pool (set) $\mathcal{K}$ of M keys is generated by the key setup server. Each key can be also assigned a unique short key identifier in the key pool $\mathcal{K}$. For each sensor node, m keys are randomly selected from the key pool $\mathcal{K}$ and stored into the node's memory. This set of m keys is called the node's key ring. The number of keys in the key pool, M, is chosen such that two random subsets of size m will share at least one key with some probability p. After deployment of sensor nodes in a target field, each node performs a direct key establishment phase (also called the shared key discovery phase). In this phase, after locating all physical neighbors in the communication range by each sensor node, they broadcast the list of key identifiers of their key rings to their neighbors. Once nodes discover that they have a shared key in their key rings, they then verify that their neighbor actually holds the key through a challenge-response protocol. The path key establishment phase is an optional phase only applied after the direct key establishment phase. If two neighbor nodes are not able to establish a direct key, they can discover a secure multi-hop path between them. Once the path is discovered, a new randomly generated key is transmitted along that path. Finally, both nodes store this newly established key for their direct communication in future.

An improved alternative of the path key establishment phase is given in [9]. The basic idea behind the improved proposed scheme is that due to the random selection of keys 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. Now, an unused key, say k in the key ring of a sensor node u may help another node v in order to establish a secure link between v and its physical neighbor w with which it does not currently share a secret key. As a result, once a secure $u - v$ path between u and v is established by the initiating node u, then transmitting k securely from u to v along the discovered path achieves this goal. Hence, using this key k, two neighbor nodes v and w can easily establish a new pairwise key k for their future secret communication. This scheme has better trade-off between overheads (communication and computational overheads), network connectivity and also resilience against node compromise than the path key establishment. In this scheme, better connectivity allows one to start with bigger networks and/or bigger key pool sizes, both leading to better security against node capture.

The analysis of the EG scheme shows that the network connectivity depends on the key pool size M for a fixed key ring size m. The network connectivity increases when the key pool size is small. Since the m keys of a key ring for a sensor node are selected from the key pool $\mathcal{K}$ *randomly without replacement*, the same key may be repeated for several pair of neighbor nodes throughout the network. Thus, if the size M of the key pool $\mathcal{K}$ is chosen to be smaller, the network connectivity increases which in turn degrades the resilience against node capture. In this scheme, even if the number of captured nodes is small, the gathered information of those captured nodes reveal a large fraction of total communication in the network when the key pool size is small. Further, the maximum supported network size for this scheme is rather small in order to be resilient against node capture attack.

In order to improve the resilience against node capture, Chan et al. proposed several modifications of the EG scheme. The q-composite scheme is a modification of the EG scheme proposed by Chan et al. [6] which requires q common keys ($q > 1$), instead of just one. In this scheme, a direct key k_{uv} between two neighbor nodes u and v is generated as the hash of all shared keys, that is, $k_{uv} = H(k_1|| \; k_2|| \; \cdots \; ||k_{q'})$, where H is a secure one-way hash function (for example $H = $ SHA-1 [20]) and k_1, k_2, ..., $k_{q'}$ are the q' common keys in their key rings. By increasing the amount of key overlap required for key establishment, the resilience against node capture is improved when compared to the EG scheme when the number of captured nodes is small. However, in this scheme the maximum supported network size is also rather small in order to be resilient against node capture attack.

The random pairwise keys scheme was another modification of the EG scheme proposed by Chan et al. [6]. If m be the size of the key ring of each sensor node and p the probability that any two nodes be able to communicate securely, then in the key predistribution phase, a total of $n = \frac{m}{p}$ unique node identifiers are generated. Here the actual size of the network may be smaller than n. For each sensor node to be deployed, a set of m other randomly selected distinct node IDs and a pairwise key is generated for each pair of nodes. The key is stored in both nodes' key rings along with the ID of the other node that also knows the key. In the direct key establishment phase, each node broadcasts their own IDs to their neighbor nodes in communication ranges. If the ID of a neighbor node is found in a node's key ring, they share a common pairwise key for communication. A cryptographic handshake is then performed between neighbor nodes for mutual verification of the common key. Since the pairwise key between two nodes is generated randomly, no matter how many nodes are captured by an adversary, the other non-compromised nodes communicate with each other with 100% secrecy. Thus, the random pairwise keys scheme provides perfect security against node capture. We note that no computational overhead is required for this scheme in order to establish secret keys between them. Though this scheme provides unconditional security against node capture and requires minimal communication and computational overheads, it does not support a large-scale network in order to achieve a decent network connectivity.

The polynomial-based key pre-distribution scheme proposed by Blundo et al. in [2] which achieves unconditional security and t-collision resistant property 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 and 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 $GF(q)$, 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 $f(u, y) = \sum_{j=0}^{t} g_j \, y^j$, where $g_j = a_{ij} \, u^i$ (mod q). We note that $f(u, y)$ is a t-degree univariate polynomial. The setup server finally loads the coefficients g_j of y^j of $f(u, y)$ in the memory of the sensor node u. In the direct key establishment phase, if a node u wants to establish a secret key with its physical neighbor v, they exchange their own ids. After receiving the id of the node v, u computes the secret key shared with v as $k_{uv} = f(u, v)$. Similarly, v computes the secret key shared with u as $k_{uv} = f(v, u)$. Since $f(u, v) = f(v, u)$, both the nodes u and v store the key k_{uv} 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* [22]. 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 wire-

less sensor networks due to the limited memory in sensors. In order to improve the resilience against node capture of the polynomial-based key pre-distribution scheme, Liu and Ning proposed the polynomial-pool based key pre-distribution scheme [27]. The polynomial-pool based key pre-distribution scheme has significantly better resilience against node capture as compared to that for the polynomial-based key pre-distribution scheme. Moreover, the resilience against node capture for the polynomial-pool based key pre-distribution scheme is much better than that for the EG scheme and q-composite scheme.

An efficient random key distribution scheme using two disjoint key pools approach has been proposed in [10]. In this scheme, the first key pool is used for initial deployment phase and the second one for dynamic node addition phase. This scheme provides high network connectivity and better resilience against node capture as compared to that for the existing schemes [6, 13, 17, 18, 27, 39].

In the existing random pre-distribution schemes [6, 18, 27], the attacker can easily fabricate fake nodes with identity of his choice with the same set of key information 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. In order to improve the resilience against active attacks (such as node fabrication attack), an identity-based efficient random key pre-distribution scheme [8] has been proposed. In this scheme, there is always a relationship between the node id and the ids of the keys generated by each sensor. Due to this property, this scheme achieves significantly better resilience against node fabrication attack as compared to that for the existing random key pre-distribution schemes [6, 18, 27].

The low-energy key management scheme (LEKM) [23] and improved key distribution mechanism (IKDM) [7] have been proposed in hierarchical WSNs. No communication between sensor nodes exist for LEKM and IKDM in the network, whereas the sensor nodes in a cluster directly communicate with the cluster head in that cluster only. These schemes have better performances than the random key distribution schemes [6, 18], because hierarchical structure has used for those schemes. LEKM requires less key storage overhead than the random schemes [6, 18]. The main drawback of LEKM is that once a cluster head in a cluster is captured, all the keys in sensors of that cluster are compromised. Though IKDM requires only two secret keys to be stored in each sensor's memory, once a cluster head in a cluster is captured after the network initialization phase, all the keys stored in sensors in that cluster are directly compromised. Recently, Paterson and Stinson [31] outlined two attacks on IKDM. They showed that their attacks can result in the compromise of most if not all of the sensor node keys after a small number of cluster heads are compromised. The basic problem in LEKM and IKDM is that all the sensors in a cluster communicate directly with the cluster head only.

Liu and Ning proposed an extended version of the closest pairwise keys scheme [25] for distributed static sensor networks which is based on the security of a pseudo-random function (PRF). The basic idea behind their extended scheme is that for each sensor u, the setup server first randomly generates a master key K_u (master key is shared with the base station only), and selects a set $S = \{v_1, v_2, \ldots, v_c\}$ of c other sensor nodes whose expected locations are closest to that of u. Then for each $v \in S$, the setup server generates a pseudo random number $k_{u,v} = PRF_{K_v}(u)$ as the pairwise key shared between u and v, where K_v is the master key for v. The generated c key-plus-id combinations $\{(k_{u,v_i}, v_i), 1 \le i \le c\}$ are loaded into the memory of the node u before its deployment. As a result, for each $v \in S$, node u stores the pairwise key $k_{u,v}$, while node v can compute the same key with its own master key and the ID of node u. This scheme has better network performances when the deployment error between the expected location and the actual location of nodes is small. However, this scheme essentially degrades to a random scheme when the deployment error is significantly large.

The group-based deterministic key distribution mechanism [11] proposed by Das and Sengupta is based on bivariate polynomials. In this scheme, every sensor node in a group can establish a secret key with its neighbor nodes (including its group head). This deterministic key distribution provides very high network connectivity and also unconditional security against node capture. It provides better security against group head node capture as compared to that for LEKM and IKDM. However, there is a limitation on the number of nodes to be deployed in each group in order to make the scheme unconditional security against node capture attack.

4 Identity-based Key Pre-distribution Using a Pseudo Random Function (IBPRF)

The bootstrapping protocol for the random key predistribution schemes [6, 18, 27] incurs much more communication overhead for establishing direct pairwise keys between sensor nodes in a sensor network. Thus, more communication overheads make the resource-constrained sensor networks to spend more energy consumption.

Our main goal is to design an energy-efficient protocol which will substantially reduce communication and computational overheads for establishing direct pairwise keys between neighbor sensors during direct key establishment phase of the bootstrapping. In order to achieve this goal, we introduce a new scheme called the *identity based key predistribution using a pseudo random function* (IBPRF) in a distributed static wireless sensor network (DWSN) as shown in Figure 2. We assume that sensor nodes are *static* after deployment in a target field.

IBPRF is motivated by the following considerations. In the random pairwise keys scheme [6], if we want to add a new sensor node u after initial deployment, the

(key) setup server (i.e., the base station) has to select randomly a set of m existing sensor nodes' ids, say, id_{v_1}, id_{v_2}, ..., id_{v_m}. We note that the existing nodes are already deployed in the sensor network. The setup server then generates a distinct pairwise secret key, say k_{u,v_i} for each pair of the newly deployed node u and the existing node v_i ($i = 1, 2, \ldots, m$). The m key-plus-id combinations are stored in the key ring of the newly deployed node u. Since the m existing nodes do not have these newly generated pairwise keys with the newly deployed node u, we have to load these generated key-plus-id combinations to the randomly chosen existing nodes' key rings. Thus, in this scheme to add a new sensor node after deploying in the sensor network, the setup server has to inform a number of existing sensors in the network for storing the newly generated key-plus-id combination with the newly deployed node about the addition of the new sensor, which may significantly introduce communication overhead. In this paper, to overcome this problem we propose the novel IBPRF scheme which achieves better network performances in the network so that (1) the storage overhead in each sensor node is small and fixed no matter how the sensors are deployed and (2) no extra communication overhead is introduced during the addition of new sensor nodes.

IBPRF has the following interesting properties:

- There is negligible amount of communication overhead during direct key establishment phase for establishing direct pairwise keys between sensors.

- There is negligible amount of communication overhead during the addition of new sensor nodes.

- IBPRF is perfectly resilient against node capture. This means that no matter how many sensor nodes in the network are captured, the non-compromised sensor nodes communicate with each other with 100% secrecy.

IBPRF is based on the following two ingredients:

- An efficient pseudo-random function (PRF) (For example, as in [8] a PRF function proposed by Goldreich et al. in 1986 [19]).

- A master key (MK) shared between each sensor node and the base station (BS).

4.1 Different Phases

The different phases for this scheme are described as follows.

4.1.1 Key Pre-distribution Phase

Let $\mathcal{N}$ be a pool of the ids of n sensor nodes in a sensor network. Assume that each sensor node u is capable of holding a total of $m+1$ symmetric cryptographic keys in its key ring $KeyRing_u$. The key predistribution has the following steps:

Step 1. For each sensor node u, the key setup server randomly generates a master-key MK_u which will be shared with the sensor node u and the base station (BS) only.

Step 2. For each sensor node u, the key set up server also assigns a unique identifier, say, id_u.

Step 3. For each sensor node u, the key setup server selects a set $S = \{id_{v_1}, id_{v_2}, \ldots, id_{v_m}\}$ of m randomly selected ids of sensor nodes from the pool $\mathcal{N}$. For each $id_{v_i} \in S$ ($i = 1, 2, \ldots, m$), the key setup server generates a symmetric key $k_{u,v_i} = PRF_{MK_{v_i}}(id_u \| id_{v_i})$ as the secret pairwise key shared between the nodes u and v_i, where MK_{v_i} is the master key for v_i. The key-plus-id combination (k_{u,v_i}, id_{v_i}) is stored in u's key ring $KeyRing_u$. We note that each node v_i easily computes the same secret key k_{u,v_i} using its own master key MK_{v_i} and the ids of nodes u and v_i.

Finally, the key ring $KeyRing_u$ of each sensor node u is loaded with the following information: (1) the identifier id_u of the node u, (2) its own master key MK_u, and (3) a list of m key-plus-id combinations $\{(k_{u,v_i}, id_{v_i}), i = 1, 2, \ldots, m\}$ calculated in *Step-3*.

4.1.2 Direct Key Establishment Phase

After deployment of sensor nodes in a deployment area (i.e., target field), sensor nodes will establish direct pairwise keys between them. Each sensor node first locates its all physical neighbors. Nodes u and v are called *physical neighbors* if they are within the communication range of one another. They are called *key neighbors* if they establish a secret pairwise key. They are *direct neighbors* if they are both physical neighbors as well as key neighbors. This phase has the following steps:

Step 1. After identifying the physical neighbors by each sensor node u, it can easily verify which ids of its physical neighbors exist in its key ring $KeyRing_u$. If u finds that it has a pre-calculated pairwise key $k_{u,v} = PRF_{MK_v}(id_u \| id_v)$ with its neighbor node v, it informs sensor v that it has such a key. This notification is done by sending a short message containing the id of node u that it has such a key. We note at this point that this message never contains the exact value of the key $k_{u,v}$.

Step 2. On receiving such a notification message by the node v, it easily calculates the secret shared pairwise key $k_{u,v} = PRF_{MK_v}(id_u \| id_v)$ using its own master key MK_v and its own id id_v as well as the id id_u of the node u. Node v stores this key k_{uv} for future secret communication with the node u.

In this way, every node can establish pairwise secret keys with its neighbor nodes in its own communication range.

Remark 1: It is noted that two keys k_{u,v_i} or $k_{v_i,u}$ can be possible between two neighbor nodes u and v_i (as in Step-3 of Section 4.1.1). To tackle this issue during the direct key establishment phase, we use the following strategy. After exchanging the ids id_u and id_{v_i} between the nodes u and v_i, if v_i first sends a notification to u that it is sharing a key with u, then u must compute the key $k_{v_i,u}$ using its own master key MK_u and the id id_{v_i} and stores it along with the id id_{v_i} in its memory. Hence, even the id id_{v_i} is present in the key ring of node u, it must not send any further notification to node v_i for key establishment. A similar situation will be taken care by the node v_i when it first receives the notification from u. This leads no additional computational complexity in such scenario.

4.1.3 Path Key Establishment Phase

This is an optional stage, and if executed, adds to the connectivity of the network. After direct key establishment, if the connectivity is still poor, 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 $\langle u = u_0, u_1, u_2, \ldots, u_{h-1}, u_h = v \rangle$ such that each (u_i, u_{i+1}) ($i = 0, 1, 2, \ldots, h-1$) is a secure link.

Step 2. u generates a random number k' 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 to node u_1.

Step 3. u_1 retrieves k' 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 to u_2.

Step 4. This process is continued until the key k' reaches to the desired destination node v.

Nodes u and v use k' 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 (as stated in [27]) to discover a path between a source node and a destination node is that 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 continues with the intermediate nodes forwarding the request further. 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 and computational 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 restricted.

Remark 2: In sensor networks, each node establishes direct keys with their neighbor nodes only rather than with every other nodes in the network. The time needed to complete the direct key establishment phase is actually short. We may then believe that the sensor nodes can be fairly well protected during the path key establishment phase when it is performed in the network initialization phase. The secure bootstrapping is thus necessary in order to apply the path key establishment phase. On the other hand, if the path key establishment phase is executed after the network initialization phase, compromise of intermediate nodes of a secure path exposes the established path key to an attacker and hence the network resilience against node capture attacks degrades (resilience against node capture attacks for the path key establishment phase is given in Section 4.2.5).

Remark 3: It may be the case that both the neighbor nodes u and v initiate the path key establishment phase concurrently for establishing a common path key between them. To minimize the wastage of resources and reduce the complexity, the following strategy can be employed. In order to establish a path key between two neighbor nodes u and v, they first discover a secure path between them. If both the nodes have discovered secure paths between them, then only the minimum hop path will be considered for path key establishment for secure transmission of the path key. Again if the two paths are of equal length, then one path discovered by a node will be taken into consideration such that the identifier of that node is greater than the other's identifier.

4.1.4 Dynamic Sensor Node Addition Phase

Sometimes nodes can be compromised or damaged. Therefore, it is necessary to redeploy some new fresh nodes in the network to continue the security services.

A centralized node revocation method was proposed by Eschenauer and Gligor [18]. In their method, when the base station detects a misbehaving node, it broadcasts a message to revoke that node. A localized mechanism for sensor network node revocation was proposed by Chan, Perrig and Song [6]. In this approach, nodes can revoke their neighbors. The Sybil attack in sensor network has been analyzed and described by Newsome et al. [29]. Further, a mechanism for distributed detection of node replication attacks in sensor networks was proposed by Parno et al. in [30]. We thus assume that the compromised (captured) nodes can be detected and as a result, the base station knows the ids of the compromised nodes in the network.

In order to add a new sensor node u, the key setup server selects a set S of m randomly selected ids of sensor nodes from the pool $\mathcal{N}$. The key setup server randomly generates a master key MK_u for node u and also assigns a unique id id_u (must be different from the ids of compromised nodes). For each sensor node id $id_v \in S$, the key setup server picks up its master key MK_v and computes

the secret key $k_{u,v} = PRF_{MK_v}(id_u||id_v)$ as the shared secret pairwise key between nodes u and v, and distributes the key-plus-id combination $(k_{u,v}, id_v)$ to the key ring of u. After deployment of sensor node u, it establishes direct pairwise keys using direct key establishment phase of IBPRF with its physical neighbors for which the ids of those neighbors are in u's key ring $KeyRing_u$.

4.2 Analysis of the IBPRF Scheme

In this section, we compute the network connectivity of our scheme during both direct key establishment and path key establishment phases. We then analyze communication overhead, computational overhead and finally security analysis of our scheme.

4.2.1 Probability of Establishing Direct Keys Between Neighbor Nodes

Let p be the probability that two physical neighbors can establish a direct pairwise key. In order to establish a secret pairwise key between two neighbor nodes, both of them will initiate the direct key establishment procedure. For the derivation of p, we note that two physical neighbors u and v can establish a pairwise key if any one of the following conditions is satisfied: (1) the identifier id_v of the node v must be resident in the key ring of node u along with the pre-calculated pairwise key $k_{u,v} = PRF_{MK_v}(id_u||id_v)$, and (2) the identifier id_u of the node u must be resident in the key ring of node v along with the pre-calculated pairwise key $k_{u,v} = PRF_{MK_u}(id_u||id_v)$. Let p' denote the probability that the id of a node will be resident in another node's key ring. The total number of ways to select m ids from the pool $\mathcal{N}$ of size n is $\binom{n}{m}$. For a fixed key ring $KeyRing_u$ of node u, the total number of ways to select $KeyRing_v$ of a node v such that $KeyRing_v$ does not have the id of u is $\binom{n-1}{m}$. Thus, we have,

$$p' = 1 - \frac{\binom{n-1}{m}}{\binom{n}{m}} = \frac{m}{n}.$$

We then have, $p = 1-$ (probability that none of u and v establish a pairwise key). Hence, we obtain,

$$p = 1 - (1 - p')^2 \approx 2\,p', \text{ if } p' \text{ is small.} \qquad (1)$$

We note that p strictly depends on the network size n and the key ring size m. The network connectivity for our scheme (IBPRF) for different values of the key ring sizes is shown in Figure 3. It is clear from this figure that when the network size is small, IBPRF provides better connectivity. Although increasing the size m of the key ring can improve the network connectivity of IBPRF, it is not suitable for wireless sensor networks due to the limited memory size of sensors (a typical example is that a sensor node can store 200 cryptographic keys). Therefore, IBPRF works well when the network size is reasonable.
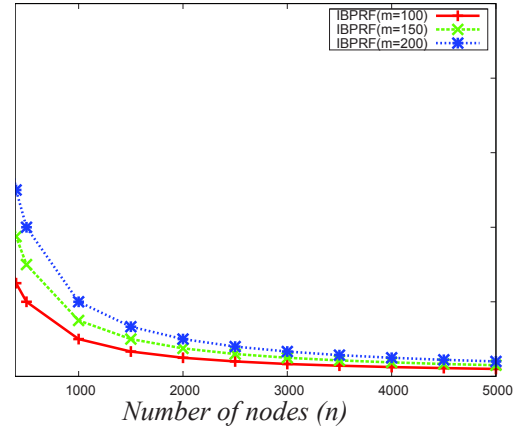

Figure 3: The probability p that two sensors establish a direct pairwise key v.s. the network size n, with $m = 100, 150, 200$.

4.2.2 Probability of Establishing Keys Using h-hop Path Key Establishment

Let d be the average number of neighbor nodes that each sensor node can contact. It follows from the similar analysis in [27] that the probability of two sensor nodes establishing a pairwise key (directly or indirectly) is

$$p_1 = 1 - (1 - p)(1 - p^2)^d.$$

If p_h is the probability that two neighbor sensor nodes can establish a key using a h-hop path key establishment phase, it is easy to deduce that

$$p_h = 1 - (1 - p_{h-1})(1 - p.p_{h-1})^d \text{ for all } h \geq 1, \qquad (2)$$

where $p_0 = p$.

The network connectivity probabilities for path key establishment with h-hop ($h = 1, 2, 3$) are plotted in Figure 4. From this figure it is also clear that one can achieve better connectivity after executing this stage even if the network is almost disconnected initially. Of course, one has to sacrifice some degradation of communication and computational overheads for this case.

4.2.3 Communication Overhead

For establishing a pairwise key between two sensor nodes u and v during the direct key establishment phase, if one of them, say u, has the id of other node v in that node's key ring, then that node sends a request message to node v that its key ring contains the shared key between them. Hence, the communication overhead during the direct key establishment phase involves only one short message for informing the other node that it has a pairwise key.

We now focus on the communication overhead required during the path key establishment phase of our basic scheme, IBPRF. We note that the path key establishment

is a complicated procedure in order to establish a secure h-hop path $\langle u = u_0, u_1, u_2, \ldots, u_{h-1}, u_h = v \rangle$ between two neighbor nodes u and v, such that each (u_i, u_{i+1}) $(i = 0, 1, 2, \ldots, h - 1)$ is a secure link. It is already stated in Subsection 4.1.3 that the path key establishment phase is similar to route discovery phase used to establish a route between two nodes. In this paper, we assume that both nodes u and v know a secure h-hop path, that is, we assume that a secure h-hop path exists between u and v. Here we only compute the number of messages to be transmitted along this secure established h-hop path in order to establish a secret key between u and v. We also assume that no retransmissions of messages are required.
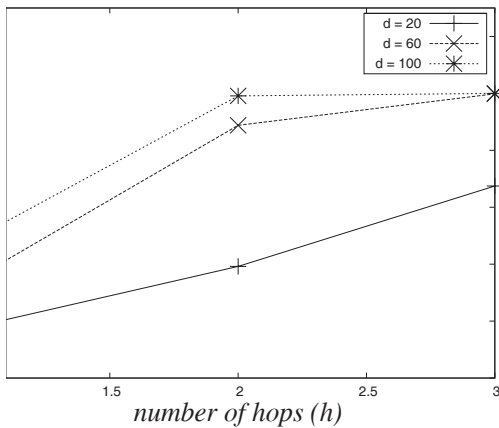

Figure 4: The probability p_h of establishing a pairwise key during h-hop path establishment phase v.s. the number h of hops in the path, with $m = 200$, $n = 5000$ and $d = 20, 60, 100$.

For 1-hop path key establishment, node u can establish a secret key with its neighbor node v via a secure path $\langle u, u_1, v \rangle$. In this case, u first generates randomly a secret key k, encrypts it using the key shared between u and u_1, and sends it to node u_1. Node u_1 then decrypts the encrypted key using the key shared between u and u_1, encrypts the key k using the key shared between u_1 and v, and finally sends to node v. v decrypts the encrypted key using the key shared between u_1 and v.

After this, a cryptographic handshake may be performed between u and v for mutual verification of the common key k. For this purpose, v first sends a challenge message encrypted by the key k to node u. In reply, node u responses with an acknowledgment that it shares the same common key k as v has. Thus, the total number of messages to be transmitted for this is $2 + 2 = 4$. In general, the total number of messages to be transmitted during the h-hop path key establishment phase is $(h+1)+2 = h+3$ and hence the communication overhead due to only transmission of messages along an established secure h-hop path for establishing a secret key between u and v requires $h + 3$ messages.

4.2.4 Computational Overhead

It is clear that for establishing a pairwise key between two sensor nodes during the direct key establishment phase, the computational overhead for a node is due to single efficient PRF operation. For the computational overhead analysis for the path key establishment phase, we assume that already a secure h-hop path exists between two neighbor nodes u and v. We further assume that no retransmissions of messages are required.

In the analysis of computational overhead due to h-hop path key establishment phase, we only consider the number of encryptions and decryptions to be carried out by the nodes along with the secure path. In case of 1-hop path key establishment, node u can establish a secret key with its neighbor node v via a secure path $\langle u, u_1, v \rangle$. We thus note that node u requires one encryption, the intermediate node u_1 requires two encryption/decryption, and finally node v requires one decryption.

If a cryptographic handshake is performed between u and v for mutual verification of their established common key, node v only requires one encryption of a challenge message using the common key. In this way, the total number of encryptions and decryptions required is $4 + 1 = 5$ and hence, in general, the total number of encryptions and decryptions required for h-hop path key establishment phase is $2(h + 1) + 1 = 2h + 3$. As a result, the computational overhead due to only encryptions and decryptions by the nodes along an established secure h-hop path for establishing a secret key between u and v is $2h + 3$.

We now compute the total number of encryptions/decryptions per node on an average during the path key establishment phase. Let p and p_h denote the probabilities that two neighbor nodes can establish a pairwise secret key during the direct key establishment phase and path key establishment phase respectively. The formulas for p and p_h are given in Equations (1) and (2) respectively. Let there be n sensor nodes deployed in the network and each node have in average d physical neighbors in its communication range. Then the network can be model as an undirected graph having n nodes and each node having degree d, and thus the total number of direct communication links in the network is $\frac{\sum_{i=1}^{n} d}{2} = \frac{nd}{2}$. The number of secure links formed during the direct key establishment phase is $\frac{nd}{2} \times p$.

Out of the remaining $\frac{nd}{2} \times (1 - p)$ links, the secure links formed during the path key establishment phase using secure h-hop paths becomes $\frac{nd}{2} \times (1 - p) \times p_h$. We note that for establishing a secure direct link using h-hop secure path between two neighbor nodes is $2h + 3$ encryptions/decryptions.

The total number of encryptions/decryptions required by the nodes in the network for establishing path keys becomes $\frac{nd}{2} \times (1 - p) \times p_h \times (2h + 3)$. Hence, the average number of encryptions/decryptions per node due to h-hop path key establishment phase turns out to be $\frac{\frac{nd}{2} \times (1-p) \times p_h \times (2h+3)}{n} = \frac{d}{2} \times (1 - p) \times p_h \times (2h + 3)$.

4.2.5 Resilience Against Node Capture Attack

The security of IBPRF depends on the followings: (1) the security of PRF [19], and (2) a node's master key MK which is shared with the base station. In this section, we discuss the security of our scheme in the following two cases.

Resilience against node capture attack for the direct key establishment phase: In this case, we calculate the resilience against node capture attack when only the direct key establishment phase is executed by the deployed sensor nodes. Based on the security of the PRF function [19], if a node's master key is not disclosed, no matter how many pairwise keys generated by this master key are disclosed, the task is still computationally difficult for an adversary to recover the master key as well as the non-disclosed pairwise keys generated with different ids of sensor nodes. Since each pre-distributed pairwise key between two sensor nodes is generated using PRF function randomly, no matter how many sensor nodes are captured, the direct pairwise keys between non-captured nodes are still secure. In other words, node compromise does not eventually lead to compromise of the direct pairwise keys between the other non-captured nodes, that is, any two non-captured neighboring nodes communicate with 100% secrecy. In this way, IBPRF provides *perfect security against node capture*, that is, IBPRF is *unconditionally secure against node capture* during the direct key establishment phase. If $P_e(c)_{direct-key}$ is the probability that the adversary can decrypt the secret communications between u and v when c sensor nodes are already compromised during the direct key establishment phase, then we have $P_e(c)_{direct-key} = 0$.

Resilience against node capture attack for the path key establishment phase: We now calculate the resilience against node capture attack if the optional path key establishment phase is executed by the nodes after the direct key establishment phase in order to increase the network connectivity. Consider a secure h-hop path $\langle u = u_0, u_1, u_2, \ldots, u_{h-1}, u_h = v \rangle$ between two neighbor nodes u and v through which u and v can establish a pairwise direct secret key between them. The secure link (u, v) is compromised by an attacker if either of its end points u and v are compromised, or any one of the intermediate nodes $u_1, u_2, \ldots, u_{h-1}$ is compromised. If a fraction f of sensor nodes are captured by an attacker in the network during the path key establishment phase, the probability that the secure link (u, v) is compromised is $1-$ (probability that the link (u, v) is not compromised) $= 1 - (1 - f)^{h+1}$. Let p and p_h denote the probabilities that two neighbor nodes can establish a pairwise secret key during the direct key establishment phase and path key establishment phase respectively. The formulas for p and p_h are given in Equations (1) and (2) respectively. Let there be n sensor nodes deployed in the network and each

node have in average d physical neighbors in its communication range. The total number of secure links in the network is $\frac{nd}{2} \times p + \frac{nd}{2} \times (1-p) \times p_h$. Now, out of these secure links, $\frac{nd}{2} \times p$ links are already secure even if the attacker captures a fraction of f of nodes in the network. Only the secure links formed during the path key establishment phase are affected due to capture of a fraction f of nodes in the network by the attacker. Hence, the attacker can compromise only $\frac{nd}{2} \times (1-p) \times p_h \times (1 - (1-f)^{h+1})$ links in the network and the rest links are secure. As a result, the resilience against node capture during the h-hop path key establishment phase due to capture of a fraction f of sensor nodes in the network can be estimated as

$$
\begin{aligned}
P_e(c)_{path-key} &= \frac{\frac{nd}{2} \times (1-p) \times p_h \times (1 - (1-f)^{h+1})}{\frac{nd}{2} \times p + \frac{nd}{2} \times (1-p) \times p_h} \\
&= (1 - \frac{p}{p + (1-p) \times p_h}) \\
&\quad \times (1 - (1-f)^{h+1}).
\end{aligned} \tag{3}
$$

The resilience against node capture during the path key establishment phase for our scheme in shown in Figure 5. We note that the resilience is good for a small number of captured nodes. However, when the number of capture nodes increases, the resilience also decreases along with the number of hops in the path key establishment phase. Thus, to keep the resilience to be higher we can make a better trade-off between the number of hops applied during path key establishment phase and the resilience.
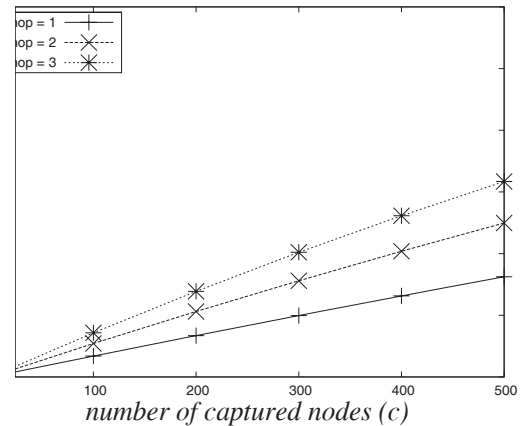

Figure 5: The resilience against node capture v.s. the number of captured nodes during the path key establishment phase for our scheme, with $m = 200$, $n = 5000$, $d = 100$ and $h = 1, 2, 3$.

4.3 Comparison with Previous Schemes

In this section, we compare the performances of our scheme (IBPRF) with the EG scheme [18], the q-composite scheme [6], the polynomial-pool based

scheme [27], the random pairwise keys scheme [6], the identity-based random key pre-distribution scheme [8] and the random scheme with disjoint key pools approach [10].

4.3.1 Network Connectivity

For comparison of network connectivity, we consider the polynomial-pool based and the random pairwise schemes because they are more resilient against node compromise than EG scheme, q-composite scheme, identity-based random key pre-distribution scheme [8] and random scheme with disjoint key pools approach [10]. It is assumed that no path key establishment phase is executed after the direct key establishment phase for the schemes. The schemes [6, 8, 10, 18] do not support large networks of arbitrarily big sizes in order to be resilient against node capture. Moreover, Chan et al. [6] showed that the maximum supported network sizes for the EG and q-composite schemes scale linearly with the size m of the key ring. It is also true for the identity-based random key pre-distribution scheme [8] and the random scheme with disjoint key pools approach [10]. Due to limited memory storage of sensor nodes, the maximum supported network sizes for these schemes are rather small in order to be perfectly resilient against node capture attacks. The relationship between the probability of establishing direct keys and the maximum supported network size for the polynomial-pool based scheme, the random pairwise keys scheme and our scheme (IBPRF) is shown in Figure 6. We assume that each sensor is capable of storing 200 keys in its key ring. From this figure, it is very clear that our scheme (IBPRF) provides better connectivity than the polynomial-pool based scheme and the random pairwise keys scheme in order to be resilient against node compromise.

4.3.2 Resilience Against Node Capture

The comparison of resilience against node capture among different existing schemes and IBPRF is shown in Figure 7. We assume that each sensor node is capable of holding 200 cryptographic keys in its memory. For the EG scheme [18], the q-composite scheme [6] and the identity-based random key pre-distribution scheme [8], it follows that even if the number of captured nodes is small, these schemes may reveal a large fraction of pairwise keys shared between non-compromised sensors when the key pool size is chosen smaller. For the random scheme with disjoint key pools approach [10], we have considered a total of 10000 nodes are deployed in the network, where 9000 nodes are initially deployed and later on 1000 nodes in the network. Due to short time period of the network initialization phase, it is assumed here that no nodes are captured during the network initialization phase, but nodes are captured after the network initialization phase. From the figure, it is evident that it provides much better resilience against node capture attacks as compared to
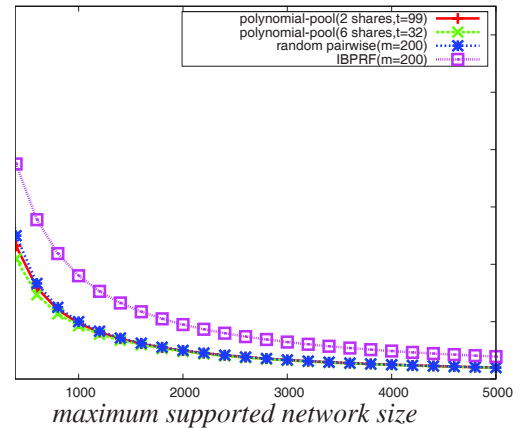

Figure 6: The probability p of establishing a common key v.s. the maximum supported network size n in order to be resilient against node compromise. Assume that each sensor node is capable of holding 200 keys.

that for the schemes [6, 8, 18]. However, the identity-based random key pre-distribution scheme [8] provides significantly better security against node fabrication attack as compared to that for the schemes [6, 10, 18]. The polynomial-pool based scheme [27] shows that this scheme is unconditionally secure and t-collusion resistant. The polynomial pool based scheme has better resilience against node capture compared to that for the EG and the q-composite schemes. However, IBPRF and the random pairwise keys scheme provide perfect security against node capture, that is, they are unconditionally secure.

4.3.3 Communication and Computational Overheads

The path key establishment is a complicated procedure and requires more communication and computational overheads for establishing path keys between neighbor nodes. We only concentrate on the direct key establishment phase of different schemes for communication and computational overheads. For the EG and the q-composite schemes, when a node wishes to establish pairwise keys with its physical neighbor nodes, it needs to broadcast a list of key ids in plaintext or a list of some messages encrypted by keys in its key ring. In the polynomial-pool based scheme, a sensor node needs to broadcast its own identifier in plaintext or a list of some messages encrypted by potential pairwise keys based on its polynomial shares for establishing direct pairwise keys with its physical neighbors. For the random pairwise keys scheme, a sensor node needs to broadcast its own identifier only to its physical neighbors in order to establish pairwise keys with its neighbors. For the identity-based random key pre-distribution scheme [8], the communication
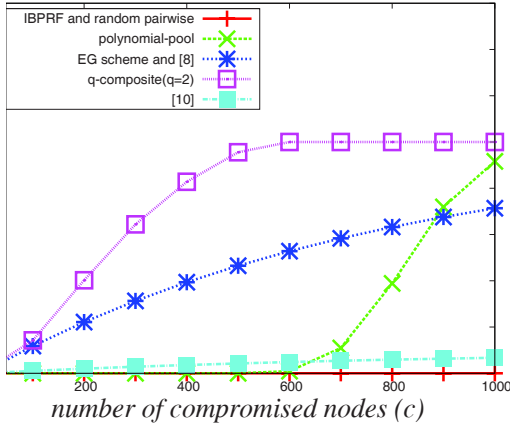
Figure 7: Comparison of resilience against node capture among our scheme (IBPRF), random pairwise, EG, polynomial-pool, q-composite, identity-based random key pre-distribution [8] and random scheme with disjoint key pools approach [10] schemes for the direct key establishment phase. The network connectivity is taken as $p = 0.22$ for all the schemes with suitable choices of the parameters.

overhead is same as the EG scheme and q-composite. In the random scheme with disjoint key pools approach [10], during the network initialization phase a node requires to send a list of m_1 key ids from its first key ring, whereas after the network initialization phase that node requires to send a list of m_2 key ids from its second key ring in order to establish a secret key with its neighbor node. Thus, the communication overhead is on the order of the key ring size for the EG scheme, q-composite scheme, polynomial-pool scheme, identity-based random key pre-distribution scheme and random scheme with disjoint key pools approach. In our proposed scheme (IBPRF), the communication overhead is only due to one short message sent by a node to inform its physical neighbor that it has a pairwise key in its key ring. Hence, IBPRF requires significantly less communication overhead than the EG, the q-composite, and the polynomial-pool based schemes. However, the communication overhead for IBPRF is comparable with that for the random pairwise keys scheme.

Liu et al. [27] reported the communication and computational overheads for direct key establishment phase for different random key pre-distribution schemes [18], [6] and the polynomial-pool based scheme [27]. In the EG scheme and q-composite scheme, the communication overhead is calculated using the size of the list of keys and for the the polynomial-pool based scheme it is calculated using the size of the list of polynomial ids. The communication overhead for the random pairwise keys scheme is negligible, since a node needs to send its own identifier to its neighbor node in order to establish direct key between

them. Now, for the EG scheme, q-composite scheme, identity-based random key pre-distribution scheme and random scheme with disjoint key pools approach, the computational overhead is calculated using the number of comparisons in identifying the common key(s). In case of the polynomial-pool based scheme, the computational overhead is calculated using the number of comparisons in identifying the common polynomial(s) and the number of polynomial evaluation(s) between two neighbor nodes. It is assumed that the ids of keys or polynomials are stored in ascending order in each node's key ring and binary search is performed to locate the id of the common key or polynomial.

The communication and computational overheads for direct key establishment between two neighbor nodes of different schemes are shown in Table 1. M and m denote the key pool size and the key ring size for the EG scheme and q-composite scheme. p_{EG} and $p_{poly-pool}$ denote the probabilities of establishing a direct key between two neighbor nodes during the direct key establishment phase, respectively. For the polynomial-pool based scheme, s is the polynomial pool size, s the number of polynomial shares given to each node, and t the degree of a symmetric bivariate polynomial over a finite field F_q. From this table, we note that due to efficient PRF operation, the computational overhead as well as communication overhead for our scheme (IBPRF) are significantly less than those for the EG scheme, the q-composite scheme, the polynomial-pool based scheme, the identity-based random key pre-distribution scheme and the random scheme with disjoint key pools approach. We also observe that though the random pairwise keys scheme does not require any communication overhead and computational overhead, it has poor network connectivity as compared to that for our scheme (IBPRF) when the network size is large.

5 The Improved Scheme

In this section, we first discuss the motivation behind the development of the improved version of our basic scheme (IBPRF). We then describe the different phases of our improved scheme. Finally, we analyze and compare the performances of the improved scheme with those for the previous existing schemes.

5.1 Motivation

From the analysis of our proposed scheme, IBPRF, it follows that the network connectivity degrades when the network size increases. As a result, IBPRF does not support a large-scale sensor network. However, IBPRF provides perfect resilience against node capture and requires only negligible amounts of communication as well as computational overheads in order to establish direct pairwise keys between neighbor sensor nodes during the direct key establishment phase.

Table 1: Comparison of communication and computational overheads for direct key establishment phase between our scheme (IBPRF), EG scheme, q-composite scheme, random pairwise scheme, polynomial-pool scheme, identity-based random scheme, and random scheme with disjoint key pools approach.

	Communication overhead	*Computational overhead*
EG scheme [18]	$m \log M$ bits	$\frac{2m+p_{EG}-p_{EG} \cdot m}{2} \log m$ comparisons
q-composite scheme [6]	$m \log M$ bits	$m \log m$ comparisons + 1 hash operation
identity-based random scheme [8]	$m \log M$ bits	m PRF operations + $m \log m$ comparisons + 1 hash function
random scheme with disjoint key pools approach [10]	$m_1 \log M$ bits (initialization phase) $m_2 \log M$ bits (dynamic node addition phase)	$m_1 \log m_1$ comparisons + 1 PRF operation (initialization phase) $m_2 \log m_2$ comparisons + 1 PRF operation (dynamic node addition phase)
random pairwise keys scheme [6]	0	0
polynomial-pool scheme [27]	$s' \log s$ bits	$\frac{2s'+p_{poly-pool}-p_{poly-pool} \cdot s'}{2} \log s'$ comparisons + 1 t-degree polynomial evaluation
Our scheme (IBPRF)	one notification message	1 PRF operation

As described in [6], the communication patterns within a sensor network fall into three categories: the first one is the node to node communications (e.g., aggregation of sensor readings), the second one is the node to base station communication (e.g., sensor readings) and the last one is the base station to node communication (e.g., specific requests). As stated in [7], wireless sensor networks are distributed event-driven systems that differ from traditional wireless networks in several ways, for examples, extremely large network size, severe energy constraints, redundant low-rate data, and many-to-one flows. Thus, in many sensing applications, connectivity between all sensor nodes is not necessary. Therefore, data centric mechanisms should be performed to aggregate redundant data in order to reduce the energy consumption and traffic load in wireless sensor networks. As a result, hierarchical heterogeneous network model (shown in Figure 1) has more operational advantages than the distributed flat homogeneous model (shown in Figure 2) for wireless sensor networks due to inherent limitations of sensors on power and processing capabilities.

For a large-scale sensor network of 10,000 sensor nodes, LEKM [23] and IKDM [7] require 100 cluster heads (if each cluster has 100 sensors to be communicated with the cluster head directly). Since a cluster head node is more expensive device than a sensor node, requirement of more cluster heads in a HWSN makes its restricted applicability in practice. We eliminate the problems in LEKM and IDKM by allowing secret communications between the sensor nodes in a cluster and only neighboring sensors

of a cluster head in that cluster will communicate with the cluster head directly. We take the number of sensor nodes in each cluster such that any two neighbor nodes (including the cluster head) communicate secretly with some reasonable probability p.

5.2 Description of our Improved Approach

Based on a three-tier hierarchical network model (shown in Figure 1), we propose an improved version of IBPRF for a large-scale wireless sensor network. In our network model, we partition the deployment field into N_{CH} number of disjoint cells called groups/clusters. A large number of sensor nodes, say, n sensor nodes will be deployed into these clusters as follows. The i-th cluster, say, $cluster_i$, consists of a cluster head (CH_i) and a set of n_i sensor nodes (distinct from sensor nodes of other clusters). Based on the prior deployment knowledge of the nodes, the sensor nodes in a particular cluster will be deployed randomly in that cluster and also a cluster head will be deployed around center of that cluster. Our approach has the following phases.

5.2.1 Key Pre-distribution Phase

This phase has the following steps:

Step 1. The cluster head, CH_i in $cluster_i$ is assigned a unique identifier, say, id_{CH_i} and also a unique master key MK_{CH_i} by the setup server. The purpose of

loading the master key in the memory of CH_i is that the cluster head CH_i will communicate secretly with the base station. The setup server assigns a unique identifier, say, id_u and also a unique master key MK_u to each sensor node u in each cluster $cluster_i$ ($i = 1, 2, \ldots, N_{CH}$). The setup server then forms a pool $\mathcal{N}_i$ of the id id_{CH_i} and the ids of the n_i sensor nodes for the i-th cluster $cluster_i$. We assume that sensor nodes and cluster head nodes are *static* after deployment in the target field.

Step 2. The setup server generates a t-degree symmetric bivariate polynomial $f(x, y) = \sum_{i,j=0}^{t} a_{ij}x^i y^j$, where the coefficients a_{ij} ($0 \le i, j \le t$) are randomly chosen from a finite field $GF(q)$, q is a prime that is large enough to accommodate a symmetric cryptographic key, with the property that $f(x, y) = f(y, x)$. The degree t of the polynomial $f(x, y)$ is so chosen such that $t > N_{CH}$ is satisfied. In this case, even all the cluster heads will be compromised, the polynomial will never be compromised. For each cluster head CH_i in $cluster_i$ to be deployed, the setup server loads the polynomial share $f(id_{CH_i}, y)$ in the memory of CH_i.

Step 3. For each sensor node u to be deployed in the cluster $cluster_i$, the setup server selects a set $S_1 = \{id_{v_1}, id_{v_2}, \ldots, id_{v_m}\}$ of m randomly selected distinct ids from the pool $\mathcal{N}_i$. It is noted that any one of these ids may be the id of the cluster head CH_i. For each $id_{v_i} \in S_1$ ($i = 1, 2, \ldots, m$), the setup server computes a pairwise key between nodes u and v_i as $k_{u,v_i} = PRF_{MK_{v_i}}(id_u || id_{v_i})$, and loads these key-plus-id combinations (k_{u,v_i}, id_{v_i}) in the memory of node u.

Step 4. For the cluster head CH_i in the cluster $cluster_i$, the setup server selects a set $S_2 = \{id_{w_1}, id_{w_2}, \ldots, id_{w_m}\}$ of m randomly selected distinct ids excluding the id of the cluster head CH_i from the pool $\mathcal{N}_i$. For each $id_{w_j} \in S_2$ ($j = 1, 2, \ldots, m$), the setup server computes a pairwise key between the cluster head CH_i and sensor node w_j as $k_{CH_i,w_j} = PRF_{MK_{w_j}}(id_{CH_i} || id_{w_j})$, and loads these key-plus-id combinations (k_{CH_i,w_j}, id_{w_j}) in the memory of the cluster head CH_i.

Each cluster head (CH_i) in the i-th cluster $cluster_i$ is loaded with the following information in its memory before deployment: (1) its own identifier id_{CH_i}, (2) its own master key MK_{CH_i}, (3) a t-degree symmetric polynomial share $f(id_{CH_i}, y)$, and (4) a list of m key-plus-id combinations calculated in Step-4. Each sensor node u in a cluster receives the following information before its deployment: (1) its own identifier id_u, (2) its own master key MK_u, and (3) a list of m key-plus-id combinations calculated in Step-3.

5.2.2 Direct Key Establishment Phase

In this phase, we have the following two sub-phases, called the *inter-cluster pairwise key establishment* and the *intra-cluster pairwise key establishment*.

1) Inter-cluster pairwise key establishment: After deployment, each cluster head broadcasts its own identifier to its neighboring cluster heads. Assume that CH_i and CH_j are two neighboring cluster head nodes. CH_i computes a secret key shared with CH_j as $k_{CH_i,CH_j} = f(id_{CH_i}, id_{CH_j})$. Similarly, CH_j computes a secret key shared with CH_i as $k_{CH_i,CH_j} = f(id_{CH_j}, id_{CH_i})$. Since $f(x, y) = f(y, x)$, both cluster heads CH_i and CH_j store k_{CH_i,CH_j} in their memory for future communication.

2) Intra-cluster pairwise key establishment: Here we consider the following cases.

   a. Node-to-node pairwise key establishment: After deployment of sensor nodes in a cluster, each sensor node broadcasts their own ids to their physical neighbors in communication ranges. Two neighbors then establish a secret key between them as in the direct key establishment phase of IBPRF (see in Subsection 4.1.2).

   b. Node-to-cluster head/Cluster head-to-node pairwise key establishment: A cluster head CH_i in the i-th cluster $cluster_i$ broadcasts its own id to its physical neighboring sensor nodes. Similarly, neighboring sensor nodes of CH_i broadcast their ids to their neighbors. Assume that CH_i and u are two neighbors. CH_i will establish a secret pairwise key with u if the id id_u of sensor node u is resident along with the calculated pairwise key $k_{CH_i,u} = PRF_{MK_u}(id_{CH_i} || id_u)$ in its key ring. Similarly, sensor node u will establish a secret pairwise key with CH_i if the id id_{CH_i} of CH_i is resident along with the calculated pairwise key $k_{u,CH_i} = PRF_{MK_{CH_i}}(id_u || id_{CH_i})$ in its key ring. Assume the id of node u is found in the key ring of CH_i. The cluster head CH_i sends a notification to u that it has a pairwise key shared with u. Node u then computes that pairwise key using its own master key MK_u and its own id id_u as well as the id id_{CH_i} of CH_i.

Remark 4: In IKDM [7], only after the inter-cluster pairwise key establishment procedure, the cluster head establishes the pairwise keys with the sensor nodes in a cluster. We observe from our direct key establishment procedure that there is no need to perform the inter-cluster pairwise key establishment before the intra-cluster pairwise key establishment. Moreover, they will be performed simultaneously in the network.

A special case: It may be possible that a sensor node, say u is not deployed in its own cluster, say $cluster_i$, and

it is deployed in another cluster, say $cluster_j$ due to deployment error. After deployment, node u broadcasts a HELLO message containing its own identifier to the nodes in its communication range. Similarly, neighbor nodes of it also broadcasts HELLO messages containing their ids. In this way, u lists its all neighbors (sensor nodes as well as cluster head) in its communication range. Since the node u does not possess any keying information to establish secret keys with its neighbor nodes in that cluster, $cluster_j$, it can establish a secret key with a neighbor node v in that cluster as follows. We use the following notations for this discussion: $E_k(M)$: a message M encrypted using key k, $MAC_k(M)$: a message authentication code (MAC) for the message M, under the key k, RN_u : a random nonce generated by the sensor node u (Nonce is a one-time random bit-string, usually used to achieve freshness), and $A||B$: data A concatenates with data B. $u \rightarrow v : M$ refers to a message M sent from a node u to another node v.

Step 1. Node u first generates a random nonce RN_u, forms a message containing its own id id_u and the generated nonce RN_u, and a computed message authentication code (MAC) on that message under its own master key MK_u. u then sends the following message to node v:

$$u \rightarrow v : (id_u||RN_u)||MAC_{MK_u}(id_u||RN_u).$$

Step 2. Node v then generates a random nonce RN_v and sends the following message to its cluster head CH_j:

$$v \rightarrow CH_j : (id_u||id_v||RN_u||RN_v)||(MAC_{MK_u}(id_u||RN_u)||MAC_{MK_v}(id_v||RN_v)).$$

Step 3. The cluster head CH_j simply forwards the received message from v to its neighbor cluster head, if required. This message finally reaches to the base station (BS) via cluster heads.

$$CH_j \rightarrow BS : (id_u||id_v||RN_u||RN_v)||$$
$$(MAC_{MK_u}(id_u||RN_u)||MAC_{MK_v}(id_v||RN_v)).$$

Step 4. The BS validates the received message. The BS computes the massage authentication codes on $id_u||RN_u$ using the master key MK_u of node u, and $id_v||RN_v$ using the master key Mk_v of node v. Note that the base station has the master keys of all sensor nodes. If both computed MACs match with the corresponding received MACs, both the nodes u and v are considered as legitimate nodes. After that the BS generates randomly a secret key $k_{u,v}$ to be shared by nodes u and v, prepares two protected copies of it: one is for node u encrypted by MK_u and other for node v encrypted by MK_v, and sends the following message to CH_j:

$$BS \rightarrow CH_j : (E_{MK_u}(id_u \oplus RN_u \oplus k_{u,v}),$$
$$E_{MK_v}(id_v \oplus RN_v \oplus k_{u,v})).$$

Step 5. After receiving the message from the base station BS, the cluster head CH_j forwards this message to the node v:

$$CH_j \rightarrow v : (E_{MK_u}(id_u \oplus RN_u \oplus k_{u,v}),$$
$$E_{MK_v}(id_v \oplus RN_v \oplus k_{u,v})).$$

Step 6. Node v decrypts $E_{MK_v}(id_v \oplus RN_v \oplus k_{u,v})$ using its own master key MK_v and retrieves the key shared with the node u using its own identifier id_v and random nonce RN_v as $k_{u,v} = (id_v \oplus RN_v \oplus k_{u,v}) \oplus (id_v \oplus RN_v)$. v stores this key for future secret communication with node u. v then sends the following message containing the first part of its received message to u:

$$v \rightarrow u : E_{MK_u}(id_u \oplus RN_u \oplus k_{u,v}).$$

Step 7. Similar to node v, after receiving the message from v, node u decrypts $E_{MK_u}(id_u \oplus RN_u \oplus k_{u,v})$ using its own master key MK_u and retrieves the key shared with the node v using its own identifier id_u and random nonce RN_u as $k_{u,v} = (id_u \oplus RN_u \oplus k_{u,v}) \oplus (id_u \oplus RN_u)$. Finally, u stores this key $k_{u,v}$ for future secret communication with the node v.

Due to involvement of the cluster heads and the base station, we have low communication and computational overheads in order to establish a secret key between two neighbor nodes. However, such a special case is unlikely to happen, because the probability of having a smaller deployment error is typically higher than the probability of having a larger one when the nodes are randomly deployed in a cluster in the deployment field.

5.2.3 Sensor Node Addition Phase

In order to add a new sensor node u in a cluster, say, $cluster_i$, the key setup server assigns a unique identifier, id_u (different from the ids of compromised sensor nodes) and a unique master key MK_u. The setup server then performs Step-3 of the key pre-distribution phase described in Subsection 5.2.1. For the sensor node u, the setup server selects a set $S = \{id_{v_1}, id_{v_2}, \ldots, id_{v_m}\}$ of m randomly selected distinct ids (including the id of the cluster head CH_i) from the pool $\mathcal{N}_i$. For each $id_{v_i} \in S$ ($i = 1, 2, \ldots, m$), the setup server calculates a pairwise key between nodes u and v_i as $k_{u,v_i} = PRF_{MK_{v_i}}(id_u||id_{v_i})$, and loads the key-plus-id combination (k_{u,v_i}, id_{v_i}) in the memory of node u. After deployment, node u establishes secret pairwise keys with its neighbor nodes using the intra-cluster pairwise key establishment procedure described in Subsection 5.2.2.

5.2.4 Cluster Head Node Addition Phase

If a cluster head node is compromised by an adversary in the network, it is necessary to redeploy a new cluster head node in order to replace that compromised cluster

head node to continue the security services in the network. In order to replace the compromised cluster head node in a cluster, say, $cluster_i$, the key setup server assigns a unique identifier, say, id_{CH_r} (different from the compromised cluster head nodes in the network) and also a unique master key MK_{CH_r} for the cluster head node CH_r to be deployed. The setup server also replaces the id and master key of the compromised cluster head node, say, CH_i by the newly assigned id and master key of CH_r in the pool $\mathcal{N}_i$ for the cluster $cluster_i$. Similar to Step-4 in Subsection 5.2.1, the setup server selects a set S of m randomly selected distinct ids excluding the id of the cluster head CH_r from the pool $\mathcal{N}_i$ and then loads the m key-plus-id combinations $\{(k_{CH_r,w_j}, id_{w_j}), w_i \in S\}$ in the memory of the cluster head CH_r. In order to establish pairwise keys with other cluster heads, CH_r is to be loaded with the same t-degree polynomial share $f(id_{CH_r}, y)$. After deployment, the cluster head CH_r will establish the pairwise keys with its neighboring sensor nodes and cluster heads as described in Subsection 5.2.2.

5.3 Analysis of the Improved Approach

In this section, we analyze the security and performances of our improved approach.

5.3.1 Network Connectivity

From the inter-cluster pairwise key establishment phase described in Subsection 5.2.2, we note that every cluster head can establish a pairwise secret key with its neighbor cluster heads in the network using its own polynomial share. Let $p_{clusterhead-clusterhead}$ denote the probability that a cluster head can establish a pairwise secret key with its another neighbor cluster head. Since every cluster head establishes a pairwise secret key with 100% with its all neighbor cluster head nodes, we have

$$p_{clusterhead-clusterhead} = 1.$$

We now consider the network connectivity between two neighbor nodes for the intra-cluster pairwise key establishment phase described in Subsection 5.2.2. Let p_i be the probability that any two nodes (including the cluster head) in a cluster $cluster_i$ can establish a secret key between them. We note that each cluster $cluster_i$ consists of a cluster head CH_i and n_i sensor nodes. Then, similar to the analysis of IBPRF, we have,

$$p_i = 1 - (1 - p_i')^2 \approx 2\, p_i', \ \text{if}\, p_i' \text{is small},$$

where $p_i' = 1 - \frac{\binom{n_i}{m}}{\binom{n_i+1}{m}} = \frac{m}{n_i+1}$ is the probability that the id of a node will be resident in another node's key ring, since the node pool is of size $n_i + 1$ and each node is given m key-plus-id combinations before its deployment. Hence, the (average) probability that any two neighboring nodes can establish a pairwise key in a cluster is given by

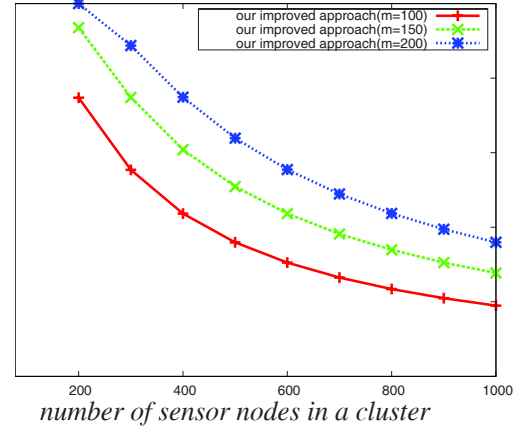$$p = \frac{\sum_{i=1}^{N_{CH}} p_i}{N_{CH}}.$$

Figure 8: The probability of establishing a common key v.s. the number of sensor nodes in each cluster, with $m = 100, 150,$ and 200.

The network connectivity in a cluster versus the number of sensor nodes in each cluster is shown in Figure 8 with different values of the key ring sizes. We note from this figure that one can achieve reasonable network connectivity with the suitable choice of the number of sensor nodes to be deployed in each cluster in a large-scale hierarchical sensor network.

For simulation of network connectivity of our improved approach, we have considered a square deployment field. The target field is partitioned into l clusters/groups CH_i ($i = 1, 2, \ldots, l$), each of equal size. For each cluster, we have deployed a cluster head CH_i around the center of the cluster. The number n_i of sensor nodes is taken to be equal for each cluster. We deploy the n_i sensor nodes randomly in each cluster. The following parameters are considered for our simulation of network connectivity:

- The number of clusters in the target field is $l = 100$.

- The number of sensor nodes deployed in each cluster is ≤ 1000.

- The area of the deployment field is $A = 1000m \times 1000m$.

- The area of each cluster is $100m \times 100m$.

- The communication range of each sensor node is 30 meters.

- The average number of nodes for each node is ≤ 100.

We have simulated the network connectivity for each cluster and then taken the average network connectivity for a cluster. Figure 9 shows the relationship between the simulated average network connectivity in a cluster versus the analytical average network connectivity in that cluster, with $m = 200$. We observe that both the simulation as well as analysis results tally closely.
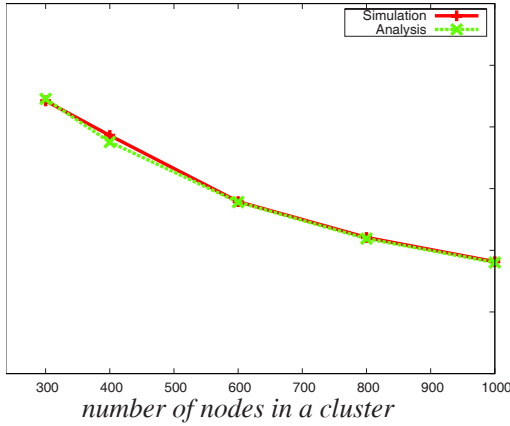
number of nodes in a cluster

number of captured sensor nodes (c)

Figure 9: Average network connectivity of a cluster CH_i, with $m = 200$ and different values of n_i.
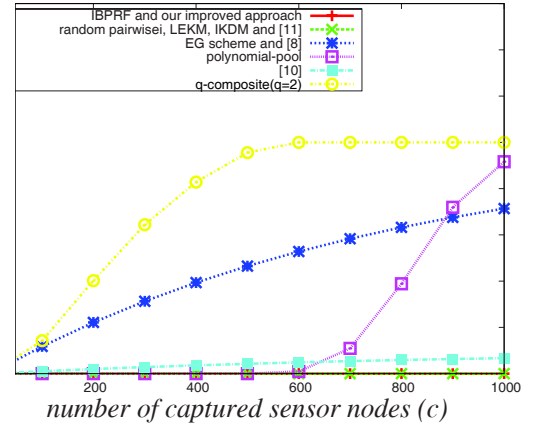
Figure 10: Fraction of compromised keys in non-captured sensor nodes v.s. number of captured sensor nodes, with $m = 200$. $P_e(c)$ denotes the fraction of compromised keys in non-captured sensor nodes after capturing c sensor nodes by an adversary.

5.3.2 Security Analysis

In this section, we compare the resilience against sensor node and cluster head capture for the direct key establishment phase of our improved approach with the existing schemes. The comparison of the fraction of compromised keys in non-captured sensor nodes versus number of captured sensor nodes among our basic scheme (IBPRF), our improved approach, EG scheme, q-composite scheme, polynomial-pool based scheme, random pairwise keys scheme, LEKM, IKDM and deterministic group-based scheme [11] is shown in Figure 10. The network connectivity is taken as 0.22 with suitable choices of the parameters for the EG scheme, q-composite scheme, polynomial-pool based scheme, random pairwise keys scheme, identity-based random key pre-distribution scheme [8] and random scheme with disjoint key pools approach [10]. From this figure, it is clear that our improved scheme is also perfect resilient against sensor node capture attack as our basic scheme (IBPRF), random pairwise keys scheme, LEKM, IKDM and deterministic group-based scheme [11]. It is also clear that our improved approach provides significantly better security against sensor node capture compared to that for the EG scheme, q-composite scheme, polynomial-pool based scheme, identity-based random key pre-distribution scheme and random scheme with disjoint key pools approach.

We now compare the network resilience against cluster head node capture attack during the network initialization phase and also after the network initialization phase among our improved approach, LEKM, IKDM and deterministic group-based scheme [11]. In LEKM and IKDM, we assume that there are 100 sensors in each cluster and 100 cluster heads in the network so that they can support 10,000 sensor nodes. In these schemes, all the sensor nodes will communicate with the cluster head node in a cluster directly. In LEKM, any single cluster head's

capture could compromise the 100 sensors' secret keys. In IKDM, if the cluster head nodes are captured in the network initialization phase, no secret keys in sensors are compromised. As stated in [28, 39], a widely accepted assumption is that an adversary will not launch an attack during few minutes following the network initial deployment and the network initialization is expected to be completed safely. However, in most sensor networks, it is expected that nodes will be captured after the network initialization phase only. Hence, in IKDM when a cluster head node is subsequently captured after the network initialization phase, all the 100 sensors' secret keys are compromised directly (as stated in [11]).

In our improved approach, we assume that each sensor node will have d neighbors (for example, $d = 100$). The network connectivity for each cluster is taken as $p_i \approx 1.00$ for $m = 200$ and $n_i = 220$. In the deterministic group-based scheme [11], the network connectivity for each group also becomes $p_i \approx 1.00$ for $m = 200$ and $n_i = 198$ in order to provide unconditional security against node capture attack. If we assume as in LEKM, IKDM and deterministic group-based scheme that there are 100 cluster heads in the network, our improved approach will support 22,000 sensor nodes whereas LEKM and IKDM support 10,000 sensor nodes only, but deterministic group-based scheme supports 19,800 sensor nodes in the network. The network resilience comparison against cluster head node capture attack during the network initialization phase among our improved approach, LEKM, IKDM and deterministic group-based scheme is shown in Figure 11. From this figure we see that our improved approach, IKDM and deterministic group-based scheme provide significantly better security as compared to that for LEKM. Figure 12 illustrates the network re-
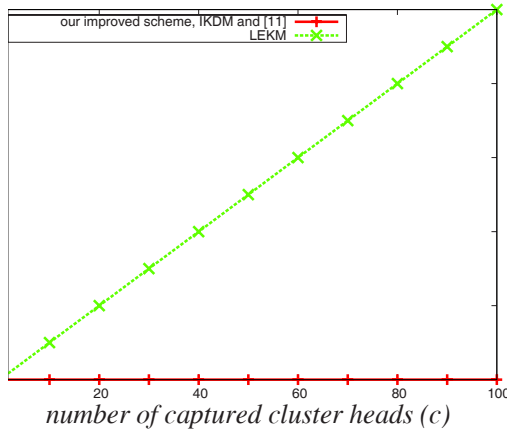
Figure 11: Number of compromised sensor keys v.s. number of the compromised cluster heads in the network initialization phase. $N_{cluster-head}(c)$ denotes the number of compromised keys in sensor nodes after capturing c cluster head nodes. In LEKM and IKDM, there are $10,000$ sensor nodes, in deterministic group-based scheme [11], there are $19,800$ sensor nodes, and in our improved approach, there are $22,000$ sensor nodes in the network.
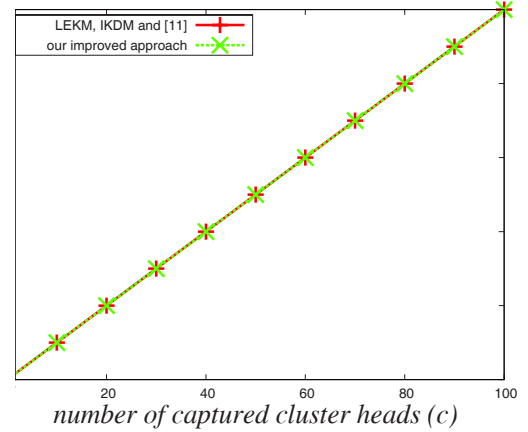

Figure 12: Number of compromised sensor keys v.s. number of the compromised cluster heads after the network initialization phase. $N_{cluster-head}(c)$ denotes the number of compromised keys in sensor nodes after capturing c cluster head nodes. In LEKM and IKDM, there are $10,000$ sensor nodes, in deterministic group-based scheme [11], there are $19,800$ sensor nodes, and in our improved approach, there are $22,000$ sensor nodes in the network.

silience comparison against cluster head node capture attack after the network initialization phase. We note from this figure that in our improved approach, even after capturing 100 cluster head nodes, an adversary can compromise only $10,000$ keys in $22,000$ sensors and in deterministic group-based scheme the adversary can compromise only $10,000$ keys in $19,800$ sensors. On the other hand, the adversary can compromise all $10,000$ keys directly in $10,000$ sensors in LEKM and IKDM. Paterson and Stinson showed in [31] that their attacks on IKDM can result in the compromise of most if not all of the sensor node keys after a small number of cluster heads are compromised after the network initialization phase. As a result, our improved approach and deterministic group-based scheme provide significantly better resilience against cluster head node capture as compared to that for both LEKM and IKDM. Moreover, our improved approach has better resilience against cluster head node capture as compared to that for deterministic group-based scheme.

5.3.3 Overheads

In our improved approach, the communication overhead remains same as that for our basic scheme (IBPRF) during the direct key establishment phase. In LEKM [23], the communication overhead involves in sending a message (in plaintext) to the cluster head by a sensor node in a cluster consisting of the id of that sensor node, whereas in IKDM [7] communication overhead is due to sending a message (in plaintext) to the cluster head by a sen-

sor node consisting of the id of that sensor node and the ids of the cluster heads from which the keys stored in the memory of that sensor node being pre-calculated in the key pre-distribution phase. In deterministic group-based scheme, a node requires communication overhead due to sending a short message consisting of the id of the polynomial share to its neighbor nodes. Thus, we see that the communication overhead for our improved approach is comparable to that for IKDM, LEKM and deterministic group-based scheme. However, our improved approach requires significantly lower communication overhead compared to that for the random key pre-distribution schemes [6, 8, 10, 18, 27].

Our improved approach reduces the computational overhead than the random key pre-distribution schemes [6, 8, 10, 18, 27]. However, the computational overhead for our improved approach is also comparable with that for LEKM, IKDM and deterministic group-based scheme. In many applications, fresh sensor nodes need to be added into an existing network to replace the power exhausted or compromised sensor nodes. Similarly, when the cluster head nodes are compromised, it is required to replace them into an existing network. In random key pre-distribution schemes [6, 18, 27], a fresh node needs to exchange its store information with the existing nodes after it is deployed into the network. Thus, a fresh sensor node addition causes lots of additional communication overheads in a network. In LEKM, fresh sensor node addition is a complicated energy-consuming procedure. In IKDM and deterministic group-based scheme, since they

are based on the polynomial share calculation; there is no additional key re-assignment and re-distribution operations needed as in LEKM, when new sensor nodes are joined into an existing network. Thus, our improved approach, IKDM, LEKM and deterministic group-based scheme have lower communication overhead than the random key pre-distribution schemes [6, 18, 27].

We now consider that a new fresh cluster head node is to be added into an existing network in order to replace a compromised cluster head node. In LEKM, since all the sensor nodes in a cluster communicate directly with the cluster head, capturing of that cluster head leads to compromise of all the keys stored in sensor nodes in that cluster, which means that one has to replace the sensor nodes in a cluster in order to replace a compromised cluster head in that cluster. As in LEKM, similar problem also exists in IKDM for adding a fresh cluster head node in order to replace a compromised cluster head. In our improved approach and deterministic group-based scheme, it is efficient to replace a compromised cluster head node in a cluster by a new fresh cluster head node without affecting the existing sensor nodes in that cluster (see in Subsection 5.2.4).

6 Conclusion

In this paper, we have proposed two new identity-based random key pre-distribution schemes in wireless sensor networks. Our first scheme, IBPRF, which is applied for a distributed wireless sensor network (DWSN) has negligible computation and communication overheads for establishing pairwise secret keys between neighbor sensor nodes during the direct key establishment phase. IBPRF provides perfect security against node capture and reasonable network connectivity during the direct key establishment phase. In addition, IBPRF supports addition of new sensor nodes after initial deployment efficiently compared to the existing random key pre-distribution schemes. Our second scheme which is an improved version of IBPRF supports a large-scale sensor network in a hierarchical architecture. Our improved approach provides better connectivity in the network compared to IBPRF and existing random key pre-distribution schemes. This scheme has also negligible communication and computation overheads as IBPRF has. It provides perfect security against sensor node capture in the network. It is also highly scalable than LEKM, IKDM and deterministic group-based scheme. Moreover, it provides efficiently addition of new sensor nodes and cluster head nodes after initial deployment compared to the existing schemes such as LEKM and IKDM in an HWSN.

Acknowledgments

References

[1] 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, 2002.

[2] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," *Advances in Cryptology (CRYPTO'92)*, LNCS 740, pp. 471-486, 1993.

[3] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey", *Technical Report*, TR-05-07, Rensselaer Polytechnic Institute, March 2005.

[4] A. K. Das, "A survey on analytic studies of key distribution mechanisms in wireless sensor networks", *Journal of Information Assurance and Security*, vol. 5, no. 5, pp. 526-553, 2010.

[5] H. Chan, V. D. Gligor, A. Perrig and G. Murilidharan, "On the distribution and revocation of cryptographic keys in sensor networks", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 233-247, July-Sept. 2005.

[6] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks", *IEEE Symposium on Security and Privacy*, pp. 197-213, USA, 2003.

[7] Y. Cheng and D.P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks", *Ad Hoc Networks*, vol. 5, no. 1, pp. 35-48, 2007.

[8] A. K. Das, "An identity-based random key pre-distribution scheme for direct key establishment to prevent attacks in wireless sensor networks", *International Journal of Network Security*, vol. 6, no. 2, pp. 134-144, March 2008.

[9] A. K. Das, "A location-adaptive key establishment scheme for large-scale distributed wireless sensor networks", *Journal of Computers*, vol. 4, no. 9, pp. 896-904, 2009.

[10] A. K. Das, "An efficient random key distribution scheme for large-scale distributed sensor networks", *Security and Communication Networks*, Published online in Wiley InterScience, DOI: 10.1002/sec.123, June 2009.

[11] A. K. Das and I. Sengupta, "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials", *Third IEEE International Conference on Communication Systems Software and Middleware (COMSWARE 2008)*, pp. 9-16, 2008.

[12] A.K. Das and D. Giri, "An identity based key management scheme in wireless sensor networks", *Proceedings of 4th Asian International Mobile Computing Conference (AMOC 2006)*, pp. 70-76, 2006.

[13] M. Chorzempa, J. -M. Park and M. Eltoweissy, "SECK: Survivable and efficient clustered keying for wireless sensor networks", *IEEE Workshop on Information Assurance in Wireless Sensor Networks, WSNIA '05*, pp. 453-458, 2005.

[14] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, November 1976.

[15] W. Du, J. Deng, Y. S. Han and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", *ACM Conference on Computer and Communications Security (CCS'03)*, pp. 42-51, Washington DC, USA, Oct. 27-31, 2003.

[16] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, July 1985.

[17] M. Eltoweissy, M. Moharram and R. Mukkamala, "Dynamic key management in sensor networks", *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122-130, April 2006.

[18] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks", *9th ACM Conference on Computer and Communication Security*, pp. 41-47, Nov. 2002.

[19] 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.

[20] Secure hash standard, *FIPS PUB 180-1, National Institute of Standards and Technology (NIST)*, U.S. Department of Commerce, April 1995.

[21] Crossbow Technology Inc., *Wireless Sensor networks*, 2010. (http://www.xbow.com)

[22] F. B. Hildebrand, *Introduction to Numerical Analysis*, Second Edition, New York: Dover, 1974.

[23] G. Jolly, M. C. Kuscu, P. Kokate and M. Younis, "A low-energy key management protocol for wireless sensor networks", *Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC'03)*, Kemer-Antalya, Turkey, June 30 - July 3 2003,

[24] J. Kohl and B. Clifford Neuman, "The Kerberos Network Authentication Service (V5)", *RFC 1510*, Sep. 1993.

[25] 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.

[26] D. Liu, P. Ning and W. Du, "Group-based key pre-distribution in wireless sensor networks", *Proceedings of 2005 ACM Workshop on Wireless Security (WiSe 2005)*, pp. 11-20, Sep. 2005.

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

[28] M. Moharrum and M. Eltoweissy, "A study of static versus dynamic keying schemes in sensor networks", *ACM Workshop on performance evaluation of Wireless Ad-hoc, Sensor and Ubiquitous Networks (PE-WASUN 2005)*, pp. 122-129, Montreal, Canada, Oct. 2005.

[29] J. Newsome, E. Shi, D. Song and A. Perrig, "The Sybil attack in sensor networks: Analysis and defenses", *Proceedings of third IEEE International Conference on Information Processing in Sensor Networks (IPSN 2004)*, pp. 259-268, 26-27 Apr. 2004.

[30] B. Parno, A. Perrig and V. Gligor, "Distributed detection of node replication attacks in sensor networks", *IEEE Symposium on Security and Privacy*, pp. 49- 63, 8-11 May 2005.

[31] M. B. Paterson and D. R. Stinson, "Two attacks on a sensor network key distribution scheme of Cheng and Agrawal", *Journal of Mathematical Cryptology*, vol. 2, no. 4, pp. 393-403, 2008.

[32] A. Rasheed and R. Mahapatra, "Secure data collection scheme in wireless sensor network with mobile sink", *Proceedings of 7th IEEE International Symposium on Network Computing and Applications (NCA 2008)*, pp. 332-340, 10-12 July 2008.

[33] 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, no. 2, pp. 120-126, Feb. 1978.

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

[35] D. R. Stinson, *Cryptography Theory and Practice*, Chapman & Hall/CRC, Third Edition, 2006.

[36] Y. Wang, "Robust key establishment in sensor networks", *ACM SIGMOD Record*, vol. 33, no. 1, pp. 14-19, March 2004.

[37] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu and M. Galloway, "A survey of key management schemes in wireless sensor networks", *Computer Communications*, vol. 30, no. 11-12, pp. 2314-2341, 2007.

[38] Y. Zhang, D. Gu and J. Li, "Exploiting unidirectional links for key establishment protocols in heterogeneous sensor networks", *Computer Communications*, vol. 31, no. 13, pp. 2959-2971, August 2008.

[39] S. Zhu, S. Setia and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks", *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500-528, November 2006.

Ashok Kumar Das is currently working as an Assistant Professor in the Center for Security, Theory and Algorithmic Research of the International Institute of Information Technology (IIIT), Hyderabad 500 032, India. Prior to joining IIIT Hyderabad, he held academic position as an Assistant Professor in Department of Computer Science and Engineering of IIIT, Bhubaneswar 751 013, India from July 2008 to May 2010. He received his Ph.D. degree in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, India on April 2009. He received the M.Tech. degree in Computer Science and Data Processing from the Indian Institute of

Technology, Kharagpur, India on January 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., 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. Dr. Das 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 2008) 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. His current research interests include cryptography, wireless sensor network security, proxy ring signature and remote user authentication. He has over 20 publications in international journals and conferences in these areas.