International Journal of Computer Networks and Communications Security

VOL. 2, NO. 12, DECEMBER 2014, 423–442
Available online at: www.ijcncs.org
ISSN 2308-9830

A Survey on key pre-distribution Schemes for security in Wireless Sensor Networks

Samira Akhbarifar[1] and A. M. Rahmani[2]

[1, 2] Research and Science University, Tehran, Iran

E-mail: [1]samira.akhbarifar@aut.ac.ir

ABSTRACT

WSN is the union of set of thousands of small sensor nodes, which have limited the capability of sensing, computing, and channeling the information in the network. Key pre distribution management is the most essential cryptographic in every kind of applications where security is worried. In recent years, there are many schemes proposed for the security, one of the central issues of these schemes is the key pre distribution management. In this paper, we classify the existing solutions and describe the key pre distribution techniques; we distinguish some advantages and disadvantages of those. In addition, we characterized all of the schemes by useful and useless properties of these. In order to categorize them, we draw taxonomy and table charts.

Keywords: *Wireless Sensor Networks, Key Management, Key Pre Distribution.*

1 INTRODUCTION

A WSN is compromised by a large number of tiny sensor nodes with limited computation capacity, storage space, and power resource. The nodes in a WSN should be able to intercommunicate with each other in order to amass data and relay it to a base station in a secure good way. Key management is one of the crucial parts for security. The main goal of the key pre distribution scheme is to provide secure communication between sensor to sensor, a group of sensor and sensor to the base station. These are known as broadcast, multicast and unicast respectively, where sensor nodes are randomly broken up in belligerently environments. Several researchers have categorized the key pre distribution into three categories: probabilistic, deterministic and hybrid. In this paper, we present some key pre-distribution approach in a WSN. We briefly note a few of these at once.

For security problems cryptographic keys should be injected in the sensor nodes, which can carry on communication securely. Hence key management becomes of extreme importance in sensor networks [26, 45]. For instance, in [30] a tree of keys is built for the hierarchical network, where the keys at a certain level are passed out to the comparable division of nodes. The keys at higher levels can be applied to gain the keys at lower levels, but not vice versa. Expedition data collection, fusion and query propagation in hostile environments is the main aim of the hierarchical network.

A good key pre distribution scheme has high scalability, low storage, and low computational and communicational costs.

This article consists of six sections. In section 2, we introduce needful terms and definitions. In section 3, we classify key pre distribution schemes. In section 4, we focus on probabilistic approaches of key pre distribution. In section 5, we concentrate on deterministic schemes and in section 6; we describe hybrid approaches of key pre distribution in WSN. In section 7, we provide performance and security evaluation for all of the schemes by taxonomy diagram and table diagram. In section 8, we gather the conclusion.

2 TERMS AND DEFINITIONS

Sensor networks must arrange several types of data packets, including packets of routing protocols and packets of key management protocols. The key establishment techniques must incorporate the following properties [1, 3]:

1. Availability: Insuring that the service provided by the whole WSN, by any part of it, or by a single sensor node must be available whenever needed.
2. Authenticity: Ability for verifying that the message sent by a node is authentic.
3. Confidentiality: The key establishment technique should protect the disclosure of data to unauthorized parties.
4. Integrity: No misrepresentation of data during transmission.
5. Scalability: The key establishment technique must allow for the variation in the size of the network.
6. Flexibility: Key establishment technique should be useful in multiple applications and allow for adding nodes at any time.
7. Non-repudiation: Preventing malicious nodes to cover their actions.
8. Survivability: Ability to offer service in the event of power failure or attacks.
9. Adaptive security service: Power to change security levels as resource availability changes.

Security protocols in sensor networks have the following constraints and demand:

1. Memory: The number of keys must be as low as possible.
2. Computation overhead: The sum of calculation for key establishment must be as depressed as possible.
3. Communication overhead: For key establishment, the nodes must broadcast as little information as potential.
4. Scalability: The system must be able to add more nodes when the demand comes up.
5. Key connectivity: Probability that two sensor nodes share some common key (or share intermediate nodes, which partake in common keys) and thus communicate with each other must be high.
6. Resistance to node fabrication: The schemes must be capable to resist node replication to guard against Sybil attacks [13].
7. Revocation: At that place must be some efficient means to revoke corrupted nodes.
8. Resilience: Once nodes are captured or compromised, the shock of the compromise on the remainder of the network must be as depressed as possible.

One method to establish secret keys is by using public-key protocols. Though there are examples of such schemes [16, 42] using elliptic curves or RSA, such protocols are rather expensive (especially in computation requirements in sensors) and require control and maintenance of keys by base stations and are hence not applied much in exercise.

Another approach requires the key distribution center (KDC) which is a resource rich center and represents as a trusted arbiter for key establishment. Instances of such scheme include transport layer security (TLS) [12], security protocols for sensor networks-SPINS [28], and Kerberos [6]. Kerberos is an authentication protocol which is based on Needham and Schroeder's protocol [24]. In Kerberos, the trusted server shares long-lived keys with every node in the network and transmits session keys to sensor nodes on request. This method is extremely expensive for message relay so is not suitable for sensor networks.

The third method is to preload the keys in sensor nodes prior to deployment, which is called key pre-distribution. There has been an extensive research on key pre-distribution schemes [6, 29]. In this chapter, we depict the problem of key pre-distribution and discuss different key pre-distribution schemes, pointing out their merits and demerits.

WSNs use symmetric key mechanism for key establishment, which consists of the following three steps.

1. *Key pre-distribution*: Preloading keys in sensor nodes prior to deployment. The keys present in a sensor node constitute the key ring, which is called key chain of the sensor. A connection exists between two nodes if they partake in a common key and therefore can communicate instantly with each other.
2. *Shared-key discovery*: Communication protocol to find shared key(s) between two communicating nodes.

3. *Path-key establishment*: If a common key does not subsist between two communicating nodes, then a route has to be ground between the transmitting nodes.

This path is composed of links among nodes sharing common keys. A path-key S. Ruj et al. is generated and communicated through the established path. The two communicating nodes communicate with each other utilizing the path-key. Key pre-distribution in WSN can be performed in any of the following three ways.

3 CLASSIFICATION OF KEY PRE DISTRIBUTION SCHEMES

There are several approaches in which we can classify key pre distribution schemes in WSN by considering various factors. Key pre distribution schemes in WSN can be classify broadly into probabilistic, deterministic and hybrid solutions.

1. Probabilistic: Key rings are randomly taken out from a key pool and located in the sensor nodes. Two nodes have communication with each other with certain probability.
2. Deterministic: Key chains are placed on sensor nodes following some definite form.
3. Hybrid: Is a combination of the above two approaches.

In discussing key pre-distribution, it is significant to discuss shared-key discovery and path-key establishment, because key pre-distribution is incomplete without the two. A naive approach to pre-distribute keys is to employ a single master key in all the nodes. Thus each node can communicate with every other node in the network using this common key. This strategy is most effective in terms of memory. Nevertheless, each node is a single point of failure that brings the whole network down. Along the other extreme, consider a network of N nodes, each node containing N − 1 keys, sharing one key with each of the other N − 1 nodes in the network. Such nodes are supposed to share pair wise keys. This guarantees that any node is linked to all other nodes in the network. More significantly, the compromise of one or more nodes does not regard the connection between any other uncompromised nodes. However, since sensors have limited storage, this alternative may not be viable for large sensor networks. Therefore, in that respect is a tradeoff between the memory board, key connectivity, and resiliency.

Resilience shows how robust the network is against node capture attacks. In these attacks node capture leads to the revelation of node's key set and all the keys belonging to compromised nodes known keys to eavesdrop communication between non compromised nodes that are using the same keys. More formally, resilience fail(s) is a function of number of compromised nodes and is defined to be the probability that communication between two non-compromised nodes could be eavesdropped when nodes are compromised at random. Communication between couples of nodes could be eaves dropped if among the keys the nodes share at least q keys occurred in the compromised nodes. Scalability characterizes the ability of a scheme to grow. The scheme is considered scalable if its operation does not degrade when the network size increases. A key pre-distribution scheme should be capable to provide enough key rings to support network extension and re-deployment. Ideally, there should be no restrictions on the number of nodes the network could hold.

4 PROBABILISTIC APPROACHES

The master key and pair wise key pre-distribution scheme is slight solutions. In other words, in any those cases, if memory need and key connectivity is considered then resilience would compromise or vice versa. Therefore, to conquer such disadvantages, there is one more solution which assures some probability that any two sensor nodes can communicate utilizing a pair wise key. The scheme does not, nevertheless, ascertain that two nodes are constantly capable to compute a pair wise key for secure communication.

4.1 Random Key Pre-Distribution Scheme [23]

In this scheme, key distribution consists of three stages, namely key pre-distribution, shared-key discovery, and path-key establishment. The key pre-distribution phase of this scheme consists of five off-line steps, namely generation of a large pool of P keys (e.g., 217 - 220 keys) and of their key identifiers; random drawing of k keys out of P without replacement to set the key ring of a sensor; loading of the key ring into the sensor's memory; saving of the key identifiers of a key ring and associated sensor identifier on a trusted controller node; every node has loaded the i-th controller node with the common key with that node. The key pre-distribution phase ensures that only a small number of keys need to be placed on each sensor node's key ring to ensure that any two nodes share (at least) a key with a chosen probability; for instance probability of 0.5, only 75 keys has chosen out of a

pool of 10,000 keys need to be on any key ring. The shared-key discovery phase takes place during DSN initialization in the operational environment where every node discovers its neighbors in wireless communication range with which it shares keys. The simplest way for any two nodes to discover if they share a key is that each node broadcast, it has in mind, the list of identifiers of the keys on their key ring. This approach does not give an adversary any attack opportunity that he does not already have. For example, if an adversary captures a node he can discover which key of that node is used for which link by decrypting communications; and if he does not capture a node, the adversary can mount a traffic analysis attack in the absence of key identifiers.

The shared-key discovery phase establishes the regional anatomy of the sensor array as seen by the routing layer of the DSN. A connection exists between two sensor nodes only if they share a key; and if a link exists between two nodes, all communication has security by link encryption. Notice that it is possible that the same key is shared by more than a couple of sensor nodes, since the key rings including keys pulled out randomly from the same pool. This does not affect a link-security exposure, therefore, in normal mode of operation sensor node trusts each other and, during the revocation phase following node-capture detection, revocation of a captured node's key ring ensures that the small set of (k) keys on that ring are removed network-wide.

The path-key establishment phase assigns a path-key to select pairs of sensor nodes in the wireless communication range that do not have a common key, but are connected by two or more links at the end of the shared-key discovery stage. Path keys need not be generated by sensor nodes. The design of the DSN ensures that, after the shared-key discovery phase is finished, a number of keys on a key ring are left unassigned to any link. For instance, both analysis and simulations indicate that even without special provisioning a substantial bit of keys are left unused on key rings. Provisioning for sufficient ring keys that are left unassigned by the determination of key-ring size (k) can also anticipate both the effects of revocation and those of incremental gain of new sensor nodes, since both may require the carrying out of the path key establishment phase after shared-key discovery. The analysis and simulations presented in the next sections indicate that such provisioning is particularly simple.

A random key pre-distribution scheme was proposed for distributed sensor networks which are dynamic in nature, in the sensory faculty that they allow the addition and deletion of sensor nodes in the network after deployment. This method fundamentally notices on the bootstrapping problem in sensor networks.

4.2 Q-Composite Random Key Pre-distribution Scheme [14]

In the basic schema, any two neighboring nodes need to recover a single common key from their key rings to establish a secure link in the key-setup phase. The procedure of the q-composite key scheme is similar to that of the basic scheme, taking issue only in the size of the key pool S and the fact that multiple keys are applied to establish communications instead of merely one. In the initialization phase, we pick a set S of random keys out of the total key space, where a card (S) is computed. For each client, we select m random keys from S (where m is the number of keys each node can take along its key ring) and store them into the node's key ring. In the key-setup phase, each node must discover all common keys it possesses with each of its neighbors. This can be accomplished with a simple local broadcast of all key identifiers that a node possesses. While broadcast-based key discovery is straightforward to implement, it suffers the disadvantage that a casual eavesdropper can identify the key sets of all the nodes in a network and thus pick an optimal set of clients to compromise in order to identify a large subset of the key pool S. A more secure, only slower, method of key discovery could utilize client puzzles such as a Merkle puzzle [35]. Each node could issue m client puzzles (one for each of the m keys) to each neighboring node. Any node that responds with the right response to the client puzzle is therefore identified as experiencing the associated key. After key discovery, each node can identify every neighbor node with which it shares at least q keys. Let the number of actual keys shared be q', where $q' \geq q$. A new communication link key K is generated as the hash of all shared keys, e.g., K = hash (k1‖ k2…‖ kq'). The keys are hashed in some canonical order, for instance, based on the order they happen in the original key pool S. Key-setup is not performed between nodes that share fewer than q keys. Q common keys chosen from a large key pool instead of one common key in the basic scheme [23] are loaded in each of the sensor nodes. This will increase the resilience of the network against node capture.

4.3 Closest Pair-wise Keys Pre-Distribution Scheme [34, 37]

It is based on pseudo random function (PRF) and a seignior key is shared between each sensor and the setup server where each sensor node share pair wise key with a number of other sensor nodes whose has the closest positions of the sensor node.

The contribution of [34] is twofold. First, we develop a simple location-aware deployment model for static sensor networks, and integrate the location information with the random pair wise key scheme. The resulting scheme keeps the nice property of the random pair wise key scheme, that is, the compromise of sensors does not contribute to the compromise of pair wise keys shared between non-compromised sensors. Yet, unlike the random pair wise key scheme, this scheme does not impose restriction on the network size. Moreover, with the same storage capacity in sensors, the scheme achieves a higher chance to establish pair wise key than the random pair wise key scheme, particularly when the deployment error is minor. The extension to this basic scheme further allows smaller storage overhead and easier deployment of dynamically added sensors. Second, we develop another location-based key pre-distribution scheme by combining a polynomial based key pre-distribution [5] with the location data. This strategy offers further trade-offs between the security against node captures and the probability of establishing pair wise keys between pairs of sensors for a given memory constraint. The analysis also indicates that this scheme offers a higher chance to establish pair wise keys between neighbor sensors and better resistance against node captures than the basic probabilistic scheme [19] and the q-composite scheme [15].

4.4 Random Key Pre-Distribution Scheme Using Node Deployment Knowledge [17]

The most important knowledge for pre-distribution is the cognition of the node that is likely to be neighbors of each sensor node. Deployment knowledge can be exhibited using non-uniform probability density functions (pdfs) which means that the positions of sensor nodes to be in certain arenas.

4.5 Key Pre-Distribution Using Post-deployment Knowledge [37]

The strategy's aim is to improve the pair wise key pre-distribution in static sensor networks. Pre-distributed keys have priority based on post-deployment knowledge, following putting an excessive amount of pre-distributed keys to each sensor node, and eliminate low priority keys to avoid node compromise attacks and return memory to the applications. The contribution of [36] is threefold. First, e acquire a localization-aware deployment model for static sensor webs, and mix the required location information with the random pair wise key scheme [14, 15].

The resulting scheme keeps the nice property of the original scheme, that is, the compromise of sensor nodes does not contribute to the compromise of pair wise keys shared directly between non-compromised nodes. Yet, unlike the original scheme, to attain a desired operation, this scheme only needs a certain network density, but does not impose any limitation on the mesh size. Moreover, with the same storage capacity in sensor nodes, this strategy achieves a higher chance to establish pair wise key than the random pair wise key scheme, particularly when the deployment error is minor. The extension to this basic scheme allows smaller storage overhead and easier deployment of dynamically added sensor nodes.

Second, we build up some other key pre-distribution strategy by combining the random subset assignment [35] with the expected location information. This system offers further trade-offs between the security against node captures and the probability of establishing pair wise keys directly between neighbor nodes given a certain memory constraint.

The analysis denotes that this scheme offers a higher chance to establish pair wise keys than the original scheme in [35] and better resistance against node attacks. Third, we distinguish a novel technique, key prioritization, to improve key pre-distribution using post-deployment knowledge.

5 DETERMINISTIC APPROACHES

Deterministic methods, broadcast only their node identifiers using which the shear keys can be set up. There is no demand to exchange key identifiers.

Key pre-distribution algorithms based on deterministic methods can be characterized as below.

5.1 Polynomial based key pre-distribution

Polynomial-based key pre-distribution system is based on pair wise keys pre-distribution source. Therefore, these schemes overcome some of the probabilistic pre-distribution schemes' disadvantages. These are: (1) Any two sensors can definitely establish a pair wise key when there are no compromised sensors; (2) Even with some nodes

compromised, the others in the network can still establish pair wise keys; (3) A node can see the usual keys to specify whether or not it can establish a pair wise key and thereby help reduce communication overhead.

5.1.1 Polynomial Based Pair-wise Key Pre-Distribution [35]

The scheme utilizes the concept of the protocol in [5] which was developed for group key pre-distribution. In this section, we briefly review the basic polynomial-based key pre-distribution protocol in [5], which is the basis of the new techniques. The protocol in [5] was developed for group key pre-distribution. Since the end is to establish pair wise keys, for ease, we only talk about the extra case of pair wise key establishment in the context of sensor networks.

To pre-distribute pair wise keys, the (key) setup server randomly generates a vicariate t-degree polynomial f(x, y) = $\sum_{j=0}^{t}\sum_{i=0}^{t} a_{ij}x_iy_j$a over a finite field Fq, where q is a prime number that is large enough to accommodate a cryptographic key, such that it has the property of f(x, y) = f(y, x). In the following, was assumed all the vicariate polynomials have this property without explicit statement. It is taken for granted that each sensor node has a unique ID. For each node i, the setup server computes a polynomial share of f(x, y), that is, f (i, y). This polynomial share is pre-distributed to node i. Thus, for any two sensor nodes i and j, the node i can compute the key f(i, j) by evaluating f(i, y) at point j, and node j can compute the same key f(j, i) = f(i, j) by evaluating f(j, y) at point i. As a result, nodes i and j can establish a common key f (i, j). In this approach, each sensor node i needs to store a t-degree polynomial f (i, x), which occupies (t + 1) log q storage space. To establish a pair wise key, both sensor nodes need to evaluate the polynomial at the ID of the other sensor node.

On that point is no communication overhead during the pair wise key establishment process. The security proof in [5] ensures that this scheme is unconditionally secure and t-collusion resistant. That is, the coalition of no more than t compromised sensor nodes knows nothing about the pair wise key between any two non-compromised nodes. It is theoretically possible to apply the general group key distribution protocol in [5] in sensor networks. Nevertheless, the storage cost of a polynomial share is exponential in terms of the group size, making it prohibitive in sensor networks.

5.1.2 Polynomial Pool-Based Pair-wise Key Pre-Distribution [35]

It is a general theoretical account which is based on polynomial based key pre-distribution and key pool [14, 23]. A pool of randomly generated bivariate polynomials is used to establish pair wise keys between sensors. The polynomial pool has two particular events. (1) When there is only one polynomial in the pool, the basic framework degenerates into the polynomial-based key pre-distribution. (2) When there is just 0-degree polynomials, the polynomial pool degenerates into a key pool as in Basic Scheme [23] or the Q-Composite scheme [14].

5.1.3 Random Subset Assignment Key Pre-Distribution Scheme [35, 37]

This scheme is based on polynomial pool-based pair wise key pre-distribution. A random method is used for subset assignment during the setup stage. A random subset of polynomials in F is chosen by the setup server and a polynomial share is split to the sensor node.

5.1.4 Grid-Based Key Pre-Distribution [35]

A strategy was offered, which is grounded on the elements of the general framework designed in [35]. This scheme ensures that any two sensors can establish a pair wise key when there is no compromised sensor, provide that the sensors can communicate with each other.

5.1.5 Location-Based Pair-Wise Keys Scheme Using Bivariate Polynomials [34]

It is based on polynomial-based key pre-distribution technique and closest pair-wise keys scheme. The aim field partitioned into small regions called cells, each of which is linked in with a unique random bivariate polynomial.

5.1.6 Closest Polynomials Scheme [36]

It is a compounding of the expected position of sensor nodes with the random subset assignment scheme in [35] to conquer the limitations of sensor nodes. The sensor node's polynomials are considered based on its expected location instead of random choice as in the original random subset assignment method.

5.1.7 Hypercube-Based Key Pre-Distribution Scheme [37]

It is a generalization of grid-based key pre-distribution scheme [35]. It ensures that any two sensor nodes can establish a pair wise key when there are no compromised sensor nodes, taking for granted that the nodes can communicate with each other.

5.1.8 Random Perturbation-Based (RPB) Scheme [48]

This scheme is based on polynomials to generate pair wise keys. The polynomials are determined over a finite field indicated as Fq, where q is a prime number.

In RPB, we present the concept of perturbation polynomial, which is specified as follows: Given a finite field Fq, a positive integer r ($2r < q$), and a set of node IDs S ($S \subset \{0, \bullet\bullet\bullet, q-1\}$), a polynomial set Φ is a set of perturbation polynomials regarding r and S if any polynomial $\varphi(.) \in \Phi$ has the following limited infection property:

$\forall u \in S, \varphi(u) \in \{0, \bullet\bullet\bullet, 2r-1\}$.

The above definition ensures that the value of a perturbation polynomial will not be greater than $2r-1$; it has at most r bits. This property is exploited in the invention of the RPB scheme. Notice that, adding a r-bit number to a l-bit number (l is the minimal integer such that $q < 2l$), the least significant r bits of the l-bit number are immediately affected, while whether its most significant $l - r$ bits are converted or not depends on if a carry being generated from the least significant r bits in the addition process. For instance, adding $(101000)2$ by $(0101)2$ changes its least significant $r = 4$ bits, but does not change the most significant $l-r = 2$ bits; however, adding it by $(1010)2$ changes both its least significant 4 bits but also the most significant 2 bits.

5.2 Matrix-based key pre-distribution schemes

In matrix-based key pre-distribution schemes, in a network of size n all possible link keys can be typified as a $n \times n$ key matrix, which is based on Bloom's concept [4]. The diminished quantity of information is stored in each sensor node, then that every couple of nodes can compute the corresponding area of the matrix, and utilizes it as the link key.

5.2.1 Grid-Group Deployment Scheme [27]

Sensor nodes are uniformly deployed in a large area instead of randomly distributing keys from a large key pool to each sensor. Private keys are consistently distributed to each sensor from a structured key pool.

5.2.2 Robust Group-Based Key Management Scheme [47]

It employs the concept of group-based key management scheme using sensor deployment knowledge. The sensor area is segmented into hexagonal grids. In this scheme, limiting how to assign B matrices and how many rows stored in each node is very important to reach a highly connected network. The purposes are to guarantee neighboring groups always share some common matrix B and neighboring nodes pick rows from the shared B matrices with high probability.

We first select some groups and specify each of them a distinct matrix B. These selected groups are called basic groups. Then, for every non-basic group, we study all its neighboring basic groups, which is adjacent to it.

We attribute to each non-basic group the B matrices assigned to its neighboring basic groups. Therefore, we can guarantee any two neighboring groups always share some common matrix B.

5.2.3 Multiple-Space Key Pre-distribution Scheme [18]

It is based on Blom's key pre-distribution scheme and combines the random key pre-distribution method [23] with it; which suggests

improved network resilience. In this method, complete graph is transforming to a connected graph, therefore each sensor node needs to carry less key information. This method proposes a new key pre-distribution scheme. The main contributions of this scheme are as follows:

1. Substantially improved network resilience against node capture over existing schemes.
2. Pair wise keys that enable certification.
3. Thorough theoretical analysis of security, and communication and computation overhead analysis.

5.3 Key pre-distribution based on graph theory

5.3.3 Tree-based Key Pre-Distribution Schemes

In tree-based key pre-distribution schemes, sensor nodes are set up in a tree in which each sensor node communicates with its parent node. Thus, the key establishment has done between neighboring nodes on the agglomeration tree. The new node gets two tickets that can be affirmed by two existing nodes randomly chosen by the network administrator, before connecting the network. Later in the deployment of a new node in the network, it generates a pair wise key for its parent node. To ship the key to the parent securely, the new node splits the key into two portions and ships them with its tickets to the nodes chosen by the administrator, which affirms the new node and forwards key materials to the parent of the new node. The desert of a tree-based key distribution is the substantial reduction of the memory cost.

5.3.1.1 ID-Based On-Way Function Scheme [33]

In this scheme a public one way hash function was applied to shorten the number of keys stored in the client. A unique ID is assigned to each sensor node and this ID is applied to compute secret keys.

5.3.1.2 Deterministic Multiple Space Blom's Scheme [33]

The primary idea of this method is to undermine the connection of the network graph to progress the resiliency of the multiple IOSs. The scheme regards the complete twofold graph Km1, m2 instead of a complete graph.

5.4 Combinatorial design-based key pre-distribution scheme [10]

The scheme determines how many chosen keys to designate to each key chain before the sensor network deployment. This scheme is interested in fixing up the components of a finite set into subsets to satisfy certain properties.

5.4.1 BIBD, SBIBD

5.4.1.1 BIBD

Blom's matrix-based method is not arisen for the key distribution in sensor networks. Yet, because many key distribution schemes build upon the use of Blom's method, it is also identified here for completeness. Let D be a symmetric matrix whose dimension is

$(\lambda+1) \times (\lambda + 1)$. Let G be an arbitrary matrix of dimension $(\lambda+1) \times n$. The matrix D works as a secret key and should be maintained as secret. The matrix G is the matrix that can be publicly recognized. Let $A= (D\cdot G)\ T$ and $K= A\times G$, where"$\cdot$" denotes matrix multiplication and $(D\cdot G)\ T$ is the transpose of $(D\cdot G)$. One can easily know that K is a symmetric matrix as follows:

$$A\cdot G = (D\cdot G)\ T \cdot G = GT \cdot D\cdot G = (A\cdot G)\ T \qquad (1)$$

The aforementioned operations are wholly executed in the finite field Fq; that is, the arithmetic is performed modulo q. In [18], for each node si, the ith row vector of A and the ith column vector of G are stored in the node si. Therefore, when two nodes si and sj would like to build their common key, they switch their columns of G in plain text and then apply their private rows of A to calculate Ki, j and Kj, i, respectively. Blom's scheme achieves so-called λ-secure [18], The security can be perfectly kept as long as no more than λ nodes are compromised. Intuitively, the security of Blom's scheme comes from the seclusion of the matrix D, whereas the matrix G acts as public information even for the antagonist. When D is totally recognized by the adversary, Blom's scheme becomes insecure. In spite of such guaranteed security, Blom's scheme cannot be immediately applied to wireless sensor networks because the storage overhead grows rapidly when the security level must be kept up in a network of large size.

5.4.1.2 SBIBD

A set system or pattern is a pair (X, A) where X is a set of v elements (points) and A is a finite set of subsets of X called blocks. The degree of a point x € X is the number of blocks containing x and (X, A) is regular of degree r, if all stages have the same degree, r. The rank of a set system is the size of the largest block and (X, A) is stated to be uniform of rank k if all blocks have the same size k. A Balanced Incomplete Block Design (BIBD) is a (v, k, λ)-BIBD or equivalently (v, b, r, k , λ)-BIBD is an arrangement of v distinct objects into b blocks where each block contains exactly k distinct objects and each object happens in exactly r different blocks such that each pair comes together in exactly λ blocks.

In a (v, k, λ)-BIBD, we have: λ (v − 1) = r (k − 1) and bk = vr. In particular, a BIBD is called symmetric BIBD when b = v and therefore r = k [41]. Symmetric (v, k, λ)-BIBD denoted by (v, k, λ)-SBIBD.

A (q2 + q + 1, q + 1, 1)-BIBD with q ≥ 2 is called a projective plane of order q and a

(q2 + q + 1, q + 1, 1)-BIBD is called an affine plane of order q where q ≥ 2.

The following theorems are well-known in combinatorial design theory (see, for proof, [41]).

Theorem 2: For every prime power q ≥ 2,

• There exists a (q2 + q + 1, q + 1, 1)-SBIBD (i.e., a projective plane of order q).

• There exists a (q2, q, 1)-BIBD (i.e., an affine plane of order q).

A Latin square on q symbols is a q × q array such that each of the q symbols occurs exactly once in each row and in each column. The number q is called order of square.

If A = (aij) and B = (bij) are any two q × q arrays, the join of A and B is a q × q arrays whose (i; j)th element is the pair (aij , bij). Latin squares A and B of order q are orthogonal if all entries of A join B are distinct. Latin square A1, A2, ..., Ar are Mutually Orthogonal Latin Squares (MOLS) if they are orthogonal in pairs.

For prime power q, a set of (q − 1) MOLS of order q can be used to construct affine plane of order q and a affine plane of order q can be converted to a projective plane of order q [2].

5.4.2 Residual design

A number of easy constructions exist which can create new BIBDs once given the existence of a symmetric figure. The relation between affine and projective plains can be extrapolated to other block patterns. If B0 is any block of (v, b, r, k, λ)-BIBD, then any two ingredients that do not belong to B0 must happen together in λ of the remaining blocks, while any two elements of B0 must be together in λ −1 of the remaining blocks. It pursues that the blocks B\B0 form a BIBD when B ranges through the remaining blocks. We shall refer to these as the residual design of the original with respect to the block B0. In general, the replication number of a residual design is not constant, and the block sizes follow no particular pattern. To work out these problems, we can begin with a symmetric balanced incomplete block design [43].

6 HYBRID APPROACHES

6.1 Dynamic key management scheme for dynamic WSN

In the suggested approach, we partition the storage of each sensor node into two pieces. We store α pre-distributed key in the first part and β post-deployment keys in the second division. Each couple of sensor nodes, which are within each other's radio range and receive a coarse pre-distributed out or post-deployment key, can communicate securely. If the two neighboring nodes do not partake in any common key, they can construct a post-deployment key using the procedure which will be reported in the following subsection.

6.2 Key Construction

Take the two neighboring nodes *i* and *j* need to construct a post-deployment key. First, each of these two nodes should generate a bundle with a random number and timestamp and transmit the generated packets to the BS through the nearest sink node. The substance of this message can be decoded simply by the BS. Then, BS sends the random number of node *i* to node *j* and the random number of node *j* to node *i* in an encrypted mode. Ultimately, these two nodes generate the shared key using the random numbers and time stamp. Subsequently, we offer a detailed procedure of the key generation phase of sensor node *i*. It is also worth mentioning that the same procedure is accomplished by sensor node *j*.

1. Sensor node i compute v_i and u_i as follows and transmits them to the sink node.

$K_i = h\ (k_i^1 \| k_i^2 \| \ldots . \| k_i^\alpha \| i \| 0)$

$MK_i = h\ (k_i^1 \| k_i^2 \| \ldots . \| k_i^\alpha \| i \| 1)$

$u_i = E_{k_i}\ \{i \| j \| S_1 \| R_i \| TS_i\}$

$v_i = MAC_{MK_i}\ \{i \| j \| S_1 \| u_i\}$

$N_i \rightarrow S_1$: $i \parallel S_1 \parallel u_i \parallel v_i$ (1)

Where α is pre-loaded key-chain size of each sensor node, and K_i^j is the j-th key in i-th sensor node's key-chain. Additionally, S_1 is the nearest sink node to the sensor node i, and R_i is a random number which is generated by sensor node i. The time stamp of node i is ascertained by TS_i and h is a ECDSA hash function [38].

2. Node S_1, sends the received message to the BS directly or through a multi-hop track (via other sink nodes). We concatenate a message authentication code (MAC) to each message.

$S_1 \rightarrow BS$: $i \parallel S_1 \parallel u_i \parallel v_i \parallel$
$MAC_{K_{S_1}}(i \parallel S_1 \parallel u_i \parallel v_i)$ (2)

3. The BS is aware of the keys which are stored in sensor node i, thus it can decrypt u_i after authenticating the message. After authenticating v_i, BS decrypts u_i and achieves R_i. Similarly, node j accomplishes the aforesaid steps. BS pulls out R_j in the same way and checks the accuracy of TS.

Then, it generates two encrypted messages v_i and u_i as follows and sends them to S_1:

$u'_i = E_{k_i}\ \{i \| j \| S_1 \| R_i \| TS_{ij}\}$

$v'_i = MAC_{MK_i}\ \{i \| j \| S_1 \| u'_i\}$

$BS \rightarrow S_1$: $BS \parallel i \parallel S_1 \parallel u'_i \parallel v'_i$

$S_1 \rightarrow N_i$: $i \parallel S_1 \parallel u'_i \parallel v'_i$ (3)

Node S_1 transmits the message to sensor node i subsequently.

4. After picking up the message, node i verifies v'_i and decrypts u'_i to obtain R_j . Next, it computes K_{ij}^{TS} according to the following Equation and puts it in the post-deployment key list.

$K_{ij}^{TS_{ij}} = \mathrm{KGF}\ (R_i \| R_j \| TS_{ij})$ (4)

Where, KGF is a key generation function which is similar to a hash function. If the number of keys in post-deployment key list is more than β, the key with minimum amount of TS, will be deleted from the list.

5. Finally node i sends ACK to the sink node [22].

$v_i'' = MAC_{K_{ij}}^{TS_{ij}}(i \| j \| ACK \| R_i \| R_j \| TS_{ij})$

$N_i \rightarrow S_1$: $i \parallel S_1 \parallel \mathrm{ACK} \parallel v_i''$ (5)

6.3 Key pre-distribution using combinatorial designs for Grid-group deployment scheme

For each region keys are pre-distributed in the nodes independently of the other region using some existing pre-distribution scheme. There are many randomized [23, 35, 36, 14], deterministic [1, 7, 8, 9, 33], and hybrid [7, 8, 12] schemes of pre-distribution available in the literature. We apply the deterministic method because we would like to insure that all nodes within a region can communicate instantly with each other. Probabilistic methods can't support this fact. Since deterministic designs have a template, discover common key and path-key establishment is effective [33]. We select the symmetric design as given in [7, 8, 10]. Any other combinatorial method which insures direct communication can be used like that given in [1]. Each of the smaller regions consists of p^2+p-2 nodes, each containing $p+1$ keys, where p is a prime power. We do not use the other designs using generalized quadrangles given in [7, 8]. Though these methods result in large network size (of the order of 3 in the number of keys), the connectivity is very depressed. These designs also have a large size of key pool (also of the order of 3 in the number of keys), nevertheless, result in very low connectivity. A low connectivity results in the higher computation for path-key deployment and that's why may decrease the battery power quickly. A large key pool size has the advantage that the resiliency is high. Hence there is a challenge between resiliency and connectivity. So, depending upon the application, we can select the key pre-distribution schemes. Here we select the symmetric design. Here the number of sensors is not of the form $p^2 + p + 1$ for some prime power p, then we select p such that $n \leq p^2 + p + 1$ and distribute keys to only n sensors. First n is adjudicated upon. Grounded along the value of n the parameter p is selected. A maximum of $r^2\ (p^2 + p + 184^*)$ sensor nodes (nodes and agents) can be tolerated.

If $r = 23$, $p = 17$, then 162403 sensor nodes can be tolerated. We distribute the keys according to Algorithm 1 given in [7, 8]. Let us denote the set of keys assigned to nodes in the region $S_{i,j}$ by $P_{i,j}$. Each of the regions has a discrete set of p^2+p+1 keys. So the whole size of the key pool is $r^2\ (p^2 + p + 1)$. Since $P_{i,j} \cap P_{i',j'} = 0$, for $(i, j) \neq (i', j')$, it can be assured that even if a few nodes (or all nodes) within a region are compromised, none of the nodes (or link between nodes) in the other regions is affected. If two nodes share a common key, then anIntralink is said to survive between the nodes.

Algorithm 1. *Key Pre-distribution Using PG(2, p)*
1: ***for*** *Each element b in GF(p)* ***do***
2: ***for*** *Each element c in GF(p)* ***do***
3: ***for*** *Each element y in GF(p)* ***do***
4: $x = -(c + by)$
5: *Assign key (x, y, 1) to node (1, b, c)*
6: ***end for***
7: *Assign key (−b, 1, 0) to node (1, b, c)*
8: ***end for***
9: ***end for***
10: ***for*** *Each element c in GF(p)* ***do***
11: ***for*** *Each element x in GF(p)* ***do***
12: *Assign key (x, −c, 1) to node (0, 1, c)*
13: ***end for***
14: *Assign key (1, 0, 0) to node (0, 1, c)*
15: ***end for***
16: ***for*** *Each element x in GF(p)* ***do***
17: *Assign key (x, 1, 0) to node (0, 0, 1)*
18: ***end for***
19: Assign key (1, 0, 0) to node (0, 0, 1)

Algorithm 2. *Shared Key Discovery Using Symmetric Design*
Require: *: (a_i , b_i , c_i) and (a_j , b_j , c_j), the identifiers of nodes i and j respectively.*
1: ***if*** $a_i = 0$ *and* $b_i = 0$ *and* $c_i = 1$ ***then***
2: ***if*** $a_j = 0$ *and* $b_j = 1$ ***then***
3: *Identifier of the common key = (1, 0, 0)*
4: ***else***
5: *Identifier of the common key =* $(-b_j, 1, 0)$
6: ***end if***
7: ***else if*** $a_i = 0$ *and* $b_i = 1$ ***then***
8: ***if*** $a_j = 0$ *and* $b_j = 1$ ***then***
9: *Identifier of the common key = (1, 0, 0)*
10: ***else***
11: *Identifier of the common key =* $(b_j c_i - c_j , -c_i, 1)$
12: ***end if***
13: ***else*** *{When* $(a_i , b_i , c_i) = (1, b_1, c_1)$ *and* $(a_j , b_j , c_j) = (1, b_2, c_2)$*}*
14: *Identifier of the common key =* $(-c_1 + b_1 \frac{c1-c2}{b1-b2}, \frac{c2-c1}{b1-b2}, 1)$
15: ***end if***

6.4 Signal based key pre-distribution

This scheme, proposes a sharing key algorithm for virtual nodes in the network.
i. A pool of *S* random keys is generated.
ii. An initial virtual sensor node *A* drawing randomly *m* distinct keys from the pool to make a key ring of *A*. The coordinate of home cell of *A* is (i_C, i_R).
iii. Begin loop *1*
a. For each node that is in the cells adjacent to home cell of A, set-up server selects k keys from A and other (m-k) keys from key pool to set apart for the node.
b. For each node *B* that is three cells far away from home cell of *A*, set up server selects *k* keys from *A* and other (*m-k*) keys from key pool to assign for *B*.
c. *A*←*B*.
iv. End loop *1* if (i, i_R) = finish with $i = 1..C$ and (i_C, j) = finish with $j = 1..R$.
v. Begin loop *2* (partitioned area *1*: co-ordinate (i,j) with $i= i_C+2..C, j = i_R +2..R$)
a. For each node *X* that has home cell co-ordinate $(i_C + 3, i_R+ 3)$, set-up server withdraws $(2k– m)$ common keys between two nodes that have home cell at $(i_C + 3, i_R)$ and
$(i_C , i_R + 3)$ to assign for *X*. To achieve *m* key of *X,* set up server takes (*m-k*) keys from the corpse of the node at $(i_C +3, i_R)$ and (*m-k*) keys from the corpse of the node at $(i_C , i_R+ 3)$.
b. For each node that is in the cells adjacent to home cell of *X,* set up server selects *k* keys from *A* and other (*m-k*) keys from key pool to set apart for the node.
c. $(i_C + 3, i_R + 3) \leftarrow (i_C + 6, i_R + 3)$ or $(i_C + 3, i_R + 6)$.
vi. End loop *2* if partitioned area *1* = finish.
vii. Loops *3, 4, 5* are similar with the other regions.

6.5 Zone-based key pre-distribution

In parliamentary law to increase the resiliency of EG scheme without reducing secure connection, Du et al. This scheme (Zo-RKP) using deployment knowledge in key pre-distribution [17]. The sensor field is carved up into zones and nodes that are to be deployed over these zones are grouped in masses.

Each zone has its own key pool and the key pools of neighboring zones share keys. Before the deployment, the nodes of each group are stored random keys that are chosen from the corresponding zone's key pool. Since the nodes of a particular zone are likely to be neighbors after the deployment, same point of secure connectivity is achieved by using less number of keys per node as compared to EG scheme. Since the nodes need to salt away the least number of keys in their key rings, less data is discovered to an attacker in case of node captures. Therefore, the resiliency increases.

6.5.1 Key Pre-distribution in Zo-RoK [31]

In the Zo-RoK scheme, the sensor field is divided into a two dimensional grid of zones/regions1 as in [17]. Each zone has its own forward and backward key pools. The forward and backward key pools of each zone are selected from global forward and

backward key pools. Moreover, the neighboring zones' forward and backward key pools share keys.

6.5.1.1 Generation of Forward and Backward Regional Key Pools:

This scheme uses different sharing factors for the key pools of horizontal/vertical and diagonal neighboring zones. As shown in Figure 1, vertical and horizontal neighbor zones share n· S keys, diagonal neighbor zones share d · S keys, where 4 (n+d) =1. The original method adopted this to multiphase networks in Zo-RoK as will be detailed below.

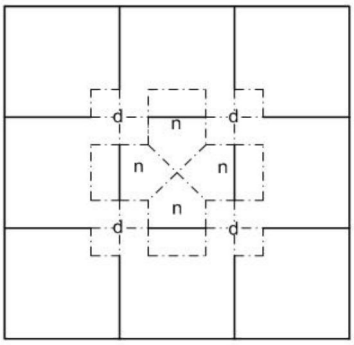

Fig. 1. Key sharing among neighboring zones

The sizes of the global forward and backward key pools are P/2 each. For each region, backward and forward key pool size is S/2. Generalize key pool should have shared and regional key pool generation mechanisms for a square sensor area with t *t zones. For each row, t-1 horizontal forward sub key pools are shared between neighboring zones. For each column, t- 1 vertical forward sub key pools are shared between neighboring zones. The entire number of horizontally and vertically shared forward sub key pools becomes 2 · t (t-1), each has distinct n · S/2 forward keys drawn from the global forward key pool. The entire number of horizontally and vertically shared backward sub key pools is 2 · t (t-1), each has distinct n · S/2 backward keys drawn from the global backward key pool. On that point are also distinct shared diagonal forward and backward sub key pools in this setting. All of diagonally shared forward sub key pools are 2 · $(t-1)^2$, each has d · S/2 distinct forward keys drawn from global forward key pool. Similarly, the total number of diagonally shared backward sub key pools are 2 · $(t-1)^2$, each has d · S/2 distinct backward keys drawn from the global backward key pool.

Each shared sub key pool has distinct keys drawn from the corresponding global key pool (back or ahead). When a key is assigned to a shared key pool, it is erased from the global one so that it is not reused in another shared pool. Each zone sets up its key pool using the above mentioned horizontally, vertically and diagonally have a common sub key pools. For each horizontally neighboring zone pair, the keys in a horizontally shared forward sub key pool are set apart to the forward key pools of these neighboring zones. A shared key pool used for a zone pair is not reused for other neighboring pairs. The same process is applied to the backward keys. Likewise, the vertical and diagonal neighbor pairs undergo the same process using vertically and diagonally shared sub key pools. In this mode, all shared sub key pools are utilized. The identities of the individual keys are specified during the assignment of shared sub key pools to zone key pools. More formally, a forward key or a backward key is identified using three tuples as $fk_{x,y,i}$ and $b_{x,y,i}$, where *x* and *y* are the indices of two neighboring zones. The index *i* is the orderliness of the key in the shared forward or backward sub key pool, where i= 1, 2,…, n·S/2 for horizontal and vertical neighbors, i= 1, 2, … , d · S/2 for diagonal neighbors.

The above mentioned process of zone key pool establishment assigns S/2 keys of non-boundary zones; therefore, the key pool establishment of these zones is filled in.

Nevertheless, this process puts less than S/2 keys in the key pools of the boundary zones since they do not have enough neighbors to share keys. In order to match the key pool sizes for all zones, boundary zones should fill up the remaining keys from the backward and forward global key pools. The entire number of missing forward keys for each of the four corner zones is (1-2n-d) · S⁄2. The total figure of missing backward keys is also the same. Other than these four corner zones, there are 4 · (t-1) side zones. The number of missing keys of the forward key pool for each of these side zones is (1-3n-2d)· S⁄2. The number of missing backward keys is also the same. The identities of those non-shared keys are designated after their assignments to the regional key pools. For the sake of standardization, again 3-tuple identification is utilized, yet the second zone index is set to 0, meaning that this key is not shared between zones. Such non-shared forward and backward keys are indicated as $fk_{x,0,I}$ and $bk_{x,0,i}$, where *x* is the index of the owning zone and *i* is the order of passing from the corresponding global key pool. The range of *i* depends on whether *x* is a corner or side zone; i= 1, 2,…, (1-2n-d) · S⁄2 for corner zones, and i= 1, 2,…, (1-3n-2d) · S⁄2 for side zones.

Equally can be understood from the concluding analysis, the backward and forward global key pools must be passed for the shared sub key pools and for the missing keys of the boundary zones. Moreover, in that respect is the same quantity of keys for both backward and forward keys in each class. Therefore, the size of regional backward and forward key pools, S/2, is computed as follows.

$$S/2=\frac{P/2}{2nt(t-1)+2d(t-1)2\ +4(1-2n-d)+\ 4(t-1)(1-3n-2d)} \quad (1)$$

RoK [3] scheme, have used the *generation* concept in Zo-RoK. Thus, the regional backward and forward key pools are to be produced for each generation. Initialize the forward key pools generation to 0. For generation-0 forward key pool of each zone, $S/2$ keys are chosen from the global forward key pool as described previously. For region z, the initial forward key pool is officially expressed as below:

$$FKP_z^o = \left\{ fk_{\eta,\xi,i}^o \,\middle|\, \begin{array}{c} (\eta=z\vee\xi=z)\wedge(\xi\leftrightarrow\xi\vee\eta\updownarrow\xi) \\ i=1,2,\ldots,n\cdot S/2,\ \eta=1,2,\ldots,Z,\ \xi=1,2,\ldots,z \end{array} \right\}$$

$$\bigcup \left\{ fk_{\eta,\xi,i}^o \,\middle|\, \begin{array}{c} (\eta=z\vee\xi=z)\wedge(\eta\uparrow\xi) \\ i=1,2,\ldots,d\cdot S/2,\ \eta=1,2,\ldots,Z,\ \xi=1,2,\ldots,z \end{array} \right\}$$

$$\bigcup \left\{ fk_{\eta,\xi,i}^o \,\middle|\, \begin{array}{c} i=1,2,\ldots,(1-2n-d)\cdot S/2, \text{ if } z \text{ is corner zone} \\ i=1,2,\ldots,(1-3n-2d)\cdot S/2, \text{ if } z \text{ is a side zone} \\ \varnothing, \text{ if } z \text{ is a non-boundary zone} \end{array} \right\}$$

$$\bigcup$$

Where $z=1,2,\ldots,z$ (2)

In parliamentary law to update the keys for the other generations, we apply the same approach employed in RoK. At each generation, the keys are updated with the assistance of an irreversible hash function. Each key of the forward key pool is hashed to generate the key pool of the following generation. The forward key pool of zone *z* in generation *j* is shown as sticks with:

$$FKP_z^j = \left\{ fk_{\eta,\xi,i}^j \,\middle|\, fk_{\eta,\xi,i}^j = H\left(fk_{\eta,\xi,i}^{j-1}\right),\ \forall \left| fk_{\eta,\xi,i}^{j-1} \in FKP_z^{j-1} \right. \right\},$$

Where $z=1,2,\ldots,Z \quad and \quad j=1,2,\ldots,m$ (3)

Backward key pools for different generations are made similar to forward key pools with one difference such that the preparations should start with the last generation, *m*. Because one-way hash chains [11] are used for each of the backward keys and they suffer to be applied from the terminal. Thus, the first regional backward key pools are the generation-*m* pools, which are officially expressed as below:

$$BKP_z^m = \left\{ bk_{\eta,\xi,i}^m \,\middle|\, \begin{array}{c} (\eta=z\vee\xi=z)\wedge(\eta\leftrightarrow\xi\vee\eta\updownarrow\xi) \\ i=1,2,\ldots,n\cdot S/2,\ \eta=1,2,\ldots,Z,\ \xi=1,2,\ldots,z \end{array} \right\}$$

$$\bigcup \left\{ bk_{\eta,\xi,i}^m \,\middle|\, \begin{array}{c} (\eta=z\vee\xi=z)\wedge(\eta\uparrow\xi) \\ i=1,2,\ldots,d\cdot S/2,\ \eta=1,2,\ldots,Z,\ \xi=1,2,\ldots,z \end{array} \right\}$$

$$\bigcup \left\{ bk_{\eta,\xi,i}^m \,\middle|\, \begin{array}{c} i=1,2,\ldots,(1-2n-d)\cdot S/2, \text{ if } z \text{ is corner zone} \\ i=1,2,\ldots,(1-3n-2d)\cdot S/2, \text{ if } z \text{ is a side zone} \\ \varnothing, \text{ if } z \text{ is a non-boundary zone} \end{array} \right\}$$

$$\bigcup$$

Where $z=1,2,\ldots,z$ (4)

Each key of these key pools is the first element of a one-way hash chain. The keys of generation $m-1$ are the second elements of the chains, and hence along. The backward key pool of zone *z* in generation *j* is represented as sticks with.

$$BKP_z^j = \left\{ bk_{\eta,\xi,i}^j \,\middle|\, bk_{\eta,\xi,i}^j = H\left(bk_{\eta,\xi,i}^{j+1}\right),\ \forall bk_{\eta,\xi,i}^{j+1} \in BKP_z^{j+1} \right\},$$

Where

$z=1,2,\ldots,Z \quad and \quad j=m-1,m-2m,\ldots,0$ (5)

Here one should note that the subscript triplets of a forward key pool are just the same as the subscript triplets of the corresponding backward key pool. This is especially important for key ring generations and session key establishment that will be discussed in subsequent parts.

6.5.1.2 Generation of Key Rings

Here, the process of key assignments to the nodes was described. Each node in Zo-RoK has forward and backward key rings, as in RoK. Each node

picks its keys from regional key pools which is different from RoK and randomly was selected.

In parliamentary law to facilitate the description of the key ring assignment process, we feign that the keys of both forward and backward key pools of each zone are placed by their subscript triplets and each key is put an implicit sequence number in the range of $[1,2,...,S/2]$. The grading is done in a manner that the implicit sequence number of a particular forward key $fk^{j}_{\eta,\xi,i}$ is the same as its backward counterpart $bk^{j}_{\eta,\xi,i}$. The ordering function $f_1(\eta,\xi,i)$ gets the subscript triplet as parameter and gives back the implicit sequence number (in the range of $1,2,...,S/2$) of that key in the corresponding forward and backward key pools. There are amount of k keys on a key ring. Half of it is for forward, the other half is for backward keys. A number generator function $f_2(.)$ was applied that gives back no repeating pseudorandom sequence of *k*/2 numbers, $r_i, i = 1,2,...,k/2$ and $0 < r_i \le S/2$. These values are then applied to define the random keys chosen from the regional key pools. For each node, we utilize this function with the generation, zone and node IDs in order to set a unique random sequence for that node. More officially, for node *A* of zone *z* at generation *j*, the random index values of forward and backward key rings are specified as below:

$$f_2(id_A \| z \| j) = (r_{id_A,z,j,i} \mid i = 1,2,...k/2,\ 0 < r_{id_A,z,j,i} \le S/2) \quad (6)$$

First, forward key ring of this node is specified by picking the forward keys with implicit sequence numbers of $r_{id_A,z,j,i}$ from the corresponding forward key pool FKP^{j}_{z}. The forward key ring of node *A* of zone *z* at generation *j*, $FKP^{j}_{id_A,z}$, is defined as below:

$$FKR^{j}_{id_A,z} = \left\{ fk^{j}_{\eta,\xi,\delta} \,\middle|\, fk^{j}_{\eta,\xi,\delta} \in FKP^{j}_{z} \wedge f_1(\eta,\xi,\delta) \equiv r_{id_A,z,j,i},\ i = 1,2,...\frac{k}{2} \right\} \quad (7)$$

The pseudorandom number sequence $r_{id_A,z,j,i}$, $i = 1,2,...,k/2$, is defined using Equation

Second, the backward key ring is defined similarly the indexing mechanism for the backward key ring. Same $f_2(.)$ pseudorandom sequence (eq. 6) is used to define the index of the backward keys to match up the forward and backward keys in the pair-wise key establishment phase. The only difference in backward key ring generation is that backward key ring of a node at generation *j* includes keys in key pools of generation $GW + j - 1$, where *GW* is a system parameter called *Generation Window*. The primary reason behind using generation window concept is to restrict the amount of generations that a particular key becomes useful in order to provide self-healing.

The use of generation window concept in Zo-RoK is borrowed from RoK scheme and will be explained in the following subdivision. The backward key ring of node *A* of zone *z* at generation *j*, $BKR^{j}_{id_A,z}$, is defined as below:

$$BKR^{j}_{id_A,z} = \left\{ bk^{GW+j-1}_{\eta,\xi,\delta} \,\middle|\, bk^{GW+j-1}_{\eta,\xi,\delta} \in BKP^{GW+j-1}_{z} \wedge f_1(\eta,\xi,\delta) \equiv r_{id_A,z,j,i},\ i = 1,2,...k/2 \right\} \quad (8)$$

The pseudorandom number sequence $r_{id_A,z,j,i}$, $i = 1,2,...,k/2$, is defined using Equation 6, as in the forward key ring. It means, the sequence numbers, and consequently the subscript triplets, of the forward keys are the same the ones of the backward keys. For model, if the forward key ring contains $fk^{j}_{4,7,22}$, then backward key ring also controls $bk^{GW+j-1}_{4,7,22}$.

Both forward and backward key rings are stacked away in the memory of a sensor node before the deployment. Thus, total of $\frac{k}{2} + \frac{k}{2} = k$ keys is stored in each sensor node.

7 TABLE AND TAXONOMY

The main purpose of this report is to point out and introduce of Key pre distribution schemes. The summarization of the paper is followed by the below Table of Contents:

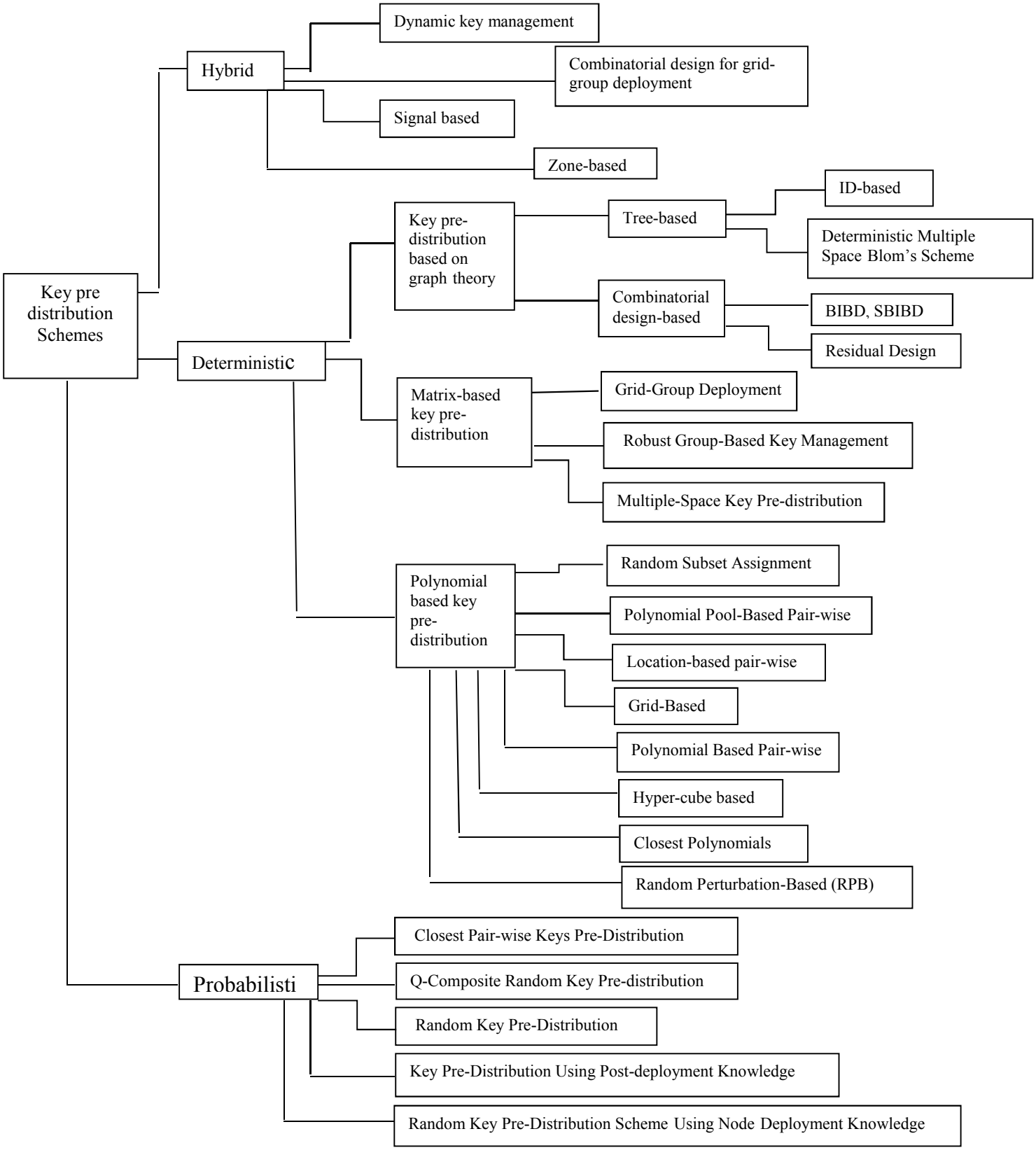
Key pre distribution Schemes
Hybrid
Dynamic key management
Combinatorial design for grid-group deployment
Signal based
Zone-based
Deterministic
Key pre-distribution based on graph theory
Tree-based
ID-based
Deterministic Multiple Space Blom's Scheme
Combinatorial design-based
BIBD, SBIBD
Residual Design
Matrix-based key pre-distribution
Grid-Group Deployment
Robust Group-Based Key Management
Multiple-Space Key Pre-distribution
Polynomial based key pre-distribution
Random Subset Assignment
Polynomial Pool-Based Pair-wise
Location-based pair-wise
Grid-Based
Polynomial Based Pair-wise
Hyper-cube based
Closest Polynomials
Random Perturbation-Based (RPB)
Probabilisti
Closest Pair-wise Keys Pre-Distribution
Q-Composite Random Key Pre-distribution
Random Key Pre-Distribution
Key Pre-Distribution Using Post-deployment Knowledge
Random Key Pre-Distribution Scheme Using Node Deployment Knowledge

Table 1: A Table that compares the different probabilistic key pre distribution scheme

Schemes	Type	Scalability	Key Connectivity	Resiliency	Key Storage	Overhead	Computation
Blom[3, 4]	probabilistic	Not Scalable	1	t-secure	t+1	t+1	t+1
Blundo[5]	probabilistic	Scalable	1	t-secure	(t+1) log q	$O(\log N)$	t+1
ES[7]	probabilistic	Scalable	$\frac{((\lvert X\rvert - k)!)^2}{(\lvert X\rvert - 2k)!\,\lvert X\rvert!}$	K/|X|	K	O(Klog |X|)	Klog |K|
Q-Composite[14, 15]	probabilistic	Limited Scalability	Given in[28]	$\binom{k}{q}$	K	O(Klog |X|)	Klog |K|
Chan-Perrig-Song[15, 16]	probabilistic	Scalable	$\frac{((\lvert X\rvert-k)!)^2}{(\lvert X\rvert-2k)!\lvert X\rvert!}$	k/|X|	Npc	O(Klog |X|)	Klog |K|
Zhu et al[62, 63]	probabilistic	Scalable	1	Given in[182]	k	O(log N)	n, n number of shares
Sadi-Kim-park[53]	Probabilistic	Not Scalable	2/(√N +1)	t-secure	O(τtlog q)	O(τ log N)	O(τ log√Nω)
Mohaisen-Maeng-Nyang[48]	probabilistic	Not Scalable	3/ $\sqrt[3]{n}$+1	Given in[110]	O($\sqrt[3]{n}$)t log q	O(log N)	O(τ log τ)
Huang et al[29, 30]	probabilistic	Scalable	Given in[69,70]	Given in[69,70]	τ(t +1)	O(τ)	ωN
SPINS[51]	probabilistic	Scalable	1	0	K	Low	Low
LEAP[62]	Probabilistic	Not Scalable	1	Prevent Sybil attack	Given in[181]	O(d^2/N)	O(N)
Jolly et al[62]	Probabilistic	NotScalable	S-S communicate Through the gateways G-G communicate Through the nodes	Given in[74](Simulation)	Sensor:2 Gateway: |S| +1 Command node: |G| + |S|	Sensor O(log(|S|/|G|)) Gateway O(log|G|)	O(1)
Cheng-Agrawal[19]	Probabilistic	Scalable	”	t-secure	Sensor:2 CH: t log q Sink node: N + m	Sensor:log⌈N/m⌉ CH:log m	O(1)
SecLEACH[49, 50]	Probabilistic	Scalable	Given in[117]	Given in[117]	Sensor: m	⌈N/m⌉	O(H), H :time to compute hash function
Du et al[22]	Probabilistic	Scalable	Given in[52]	Given in[52]	L:K H >>k	k log|X|	k log k
HERO[46]	Probabilistic	Scalable	Given in[97]	Given in[97]	N_n :≤ k S_n : k	k	k log k

Table 2: A Table that compares the different deterministic key pre distribution scheme

Scheme	Type	Scalability	Key Connectivity	Resiliency	Key Storage	Overhead	Computation
PIKE[16]	Deterministic	Not Scalable	1/√N	1/√N	O(√N)	O(log N)	O(1)
Kalidindi et al[35]	Deterministic	Not Scalable	Given in[75]	1- $\frac{12m-6}{N}$	6(√N/l)	O(log N)	1
Du et al [23](Hash-key)	Deterministic	Not Scalable	Given in[95]	Given in[95]	$(\frac{(m+n)/2}{t})$log q	O(t +1)	O(t +1)
Martin-Paterson-Stinson[47]	Deterministic	Not Scalable	k/n if in the same group [(n-1)/(	Givenin[104]	(m +1)(t +1)	O(log N)	O(1)

			$n^2/\lambda-1)]+[(n^2/\lambda-n)/ n^2/\lambda-1]$ mn if in different group				
Camtepe and Yener[7- 10] (Symmetric)	Deterministic	Not Scalable	1	$\frac{q-1}{q2+q+1}$	O(√N)	O(log N)	O(1)
Lee-Stinson[38]	Deterministic	Not Scalable	k/r+1	r-2/b-2	K	O(log N)	O(1)
Dong et al[21]	Deterministic	Not Scalable	≈0.5	Given in[44]	$O(\sqrt[3]{n})$	1	$O(\sqrt[3]{n})$
Liu-Ning[42]	Deterministic	Not Scalable	Given in[44]	Given in[44]	O((t +1)log q)	O(1)	O(t)
Yu-Guan[58, 59]	Deterministic	Not Scalable	1	t-secure	N(t +1)	t log t	Nt log t
Simonova et al[54]	Deterministic	Scalable	Given in[149]	Given in[149]	O(pN/k)	O(log p′)	O(1)
Das-Sengupta[20]	Deterministic	Scalable	”	t-secure	Sensor: t log q CH: t log q Sinknode: ⌈N/m⌉+ m	Sensor:log⌈N/m⌉ CH:logm	O(1)
SHELL[57]	Deterministic	Scalable	1	Given in[174]	Sensor: c + k	k log|X|	O(1)

Table 3: A Table that compares the different hybrid key pre distribution scheme

Scheme	Type	Scalability	Key Connectivity	Resiliency	Key Storage	Overhead	Computation
Liu-Ning-Li[39-45]	Hybrid	Scalable	Given in[91]	t-secure	s′(t +1)log q	s′ log|F|	t +1
Liu-Ning-Du[45](Polynomial)	Hybrid	Scalable	Given in[95]	Given in[95]	$\frac{(m+n)/2}{t}$ log q	O(t +1)	O(t +1)
Chakrabarti,MaitraandRoy[13]	Hybrid	Not Scalable	Given in[23,24]	Given in[23,24]	zk-$\binom{z}{2}$k/r+1	O(z log N)	z log|X|
Du et al[24]	Hybrid	Scalable	Given in[47]	t-secure	ωτ log q	τ log ω	(t +1)ω log ω
Zhou et al[13] SectionVIIB) Givenin[13](SectionVIIA)	Hybrid	Not Scalable	Given in[179]	Given in[179]	Four types of nodes	O(N)	Depends on the type of node
Erfani et al[26]	Hybrid	Scalable	1	$\binom{n}{q}$	N	1	K log K
Zone-based[36]	Hybrid	Not Scalable	-	n^2	Ns	n^2 log n	nlog n

8 CONCLUSION

Key pre distribution for WSNs is one of the main concerned areas in terms of providing security. In this paper, we provide taxonomy of key management schemes for WSNs with their advantages and disadvantages. We haven’t judged any precedence schemes by tables I, II, III because each of them has some positive and dispositive properties. By attention to them, we can select one of the schemes.

9 REFERENCES

[1] M. Al-Shurman and S.M. Yoo. Key pre-distribution using mds codes in mobile ad hoc networks. In ITNG, p.p 566–567. IEEE Computer Society, 2006.

[2] I. Anderson, E. Horwood Combinatorial designs: Construction Methods. Chicester, U.K. , 1990.

[3] R. Blom. An optimal class of symmetric key generation systems. In Pro- ceeding of EUROCRYPT, p.p 335–338, 1984.

[4] R. Blom Theory and application of cryptographic techniques. Eurocrypt 84 workshop on advances in cryptology. Springer, Berlin, 1984.

[5] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, Perfectly-Secure Key Distribution for Dynamic Conferences. In Proceedings of Crypto 92', 1992.

[6] D. Boneh and M. K. Franklin. An efficient public key traitor tracing scheme. In Michael J. Wiener, editor, CRYPTO, volume 1666 of Lecture Notes in Computer Science, p.p 338–353. Springer, 1999.

[7] S. A. Camtepe, and B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks. In Proceedings of the European Symposium on Research in Computer Security (ESORICS'04). P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, Eds. Lecture Notes in Computer Science, vol. 3193. Springer, 293–308, 2004.

[8] S. A. Camtepe, and B. Yener, Key distribution mechanisms for wiless sensor net works: A survey. Tech. rep. TR-05-07, Computer Science DePartment, Rensselaer Polytechnic Institute, 2005.

[9] S. A. Camtepe, B. Yener, and M. Yung, Expander graph based key distribution mechanisms in wireless sensor networks. In Proceedings of the IEEE International Conference on Communications. 2262–2267, 2006.

[10] S. A. Camtepe, and B. Yener, Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks IEEE/ACM Transactions on Networking, 15(2), 346-358, 2007.

[11] C. Castelluccia, and A. Spognardi, Rok: A robust key pre-distribution protocol for multi-phase wireless sensor networks. SecureComm, Third International Conference on Security and Privacy in Communication Networks, , 2007.

[12] D. Chakrabarti. Applications of combinatorial designs in key pre- distribution in sensor networks. Ph. D. Thesis, Indian Statistical Institute, India, September, 2007.

[13] D. Chakrabarti, S. Maitra, and B. Roy, A key pre- distribution scheme for wireless sensor networks: Merging blocks in combinatorial design. In Jianying Zhou, Javier Lopez, Robert H. Deng, and Feng Bao, editors, ISC, volume 3650 of Lecture Notes in Computer Science, p.p 89–103. Springer, 2005.

[14] D. Chakrabarti and J. Seberry, Combinatorial structures for design of wireless sensor networks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, ACNS, volume 3989 of Lecture Notes in Computer Science, p.p 365–374, 2006.

[15] H. Chan, A. Perrig, and D. Song, Random Key Pre- Distribution Schemes for Sensor Networks. IEEE Symposium on Research in Security and Privacy, 2003.

[16] H. Chan, A. Perrig PIKE: Peer Intermediaries For Key Establishment in Sensor Networks. 24th annual joint conference of the IEEE Computer and Communications Societies (INFOCOM '05), Miami, FL, USA, 2005.

[17] J. Charles Colbourn and J.H. Dinitz, The CRC Handbook of Combinatorial Designs. CRC Press, 1995.

[18] M. Chen, W. Cui, V. Wen, and A. Woo. Security and deployment issues in a sensor. Ninja Project: A Scalable Internet Services Architecture, Berkeley, 2000.

[19] Y. Cheng and D. P. Agrawal. An improved key distribution mecha- nism for large-scale hierarchical wireless sensor networks. Ad Hoc Networks, 5(1):35–48, 2007.

[20] A. K. Das and Indranil Sengupta. An efective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials. In COMSWARE, pages 9–16. IEEE, 2008.

[21] J. Dong, D. Pei, and X. Wang. A key predistribution scheme based on 3-designs. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, Inscrypt, volume 4990 of Lecture Notes in Computer Science, pages 81–92. Springer, 2007.

[22] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. IEEE INFOCOM, 2004.

[23] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, and A. Khalili, A pairwise key pre-distribution scheme for wireless sensor networks. ACM Transactions on Information and System Security (TISSEC) 8(2): 228–58, 2005.

[24] X. Du, Y. Xiao, M. Guizani, and H. H. Chen. An effective key management scheme for heterogeneous sensor networks. Ad Hoc Networks, 5(1):24–34, 2007.

[25] B. Dutertre, S. Cheung, and J. Levy, Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust. Tech. Rep. SRI-SDL-04-02, System Design Laboratory, 2004.

[26] S. H. Erfani, H. H. S. Javadi and A. M. Rahmani, A dynamic key management scheme

for dynamic wireless sensor networks, Security and Communication Networks, DOI: 10.1002/sec.1058, 2014.

[27] L. Eschenauer, and V.D. Gligor, A Key-Management Scheme for Distributed Sensor Networks. 9th ACM conference on Computer and Communications Security, P.P 41-47, 2002.

[28] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, Comparing elliptic curve cryptography and RSA on 8-bit cpus. In Marc Joye and Jean-Jacques Quisquater, editors, CHES, volume 3156 of Lecture Notes in Computer Science, p.p. 119–132. Springer, 2004.

[29] D. Huang, M. Mehta, D. Medhi, and L. Harn, Location-aware key management scheme for wireless sensor networks. ACM workshop on security of ad hoc and sensor networks (SASN04). Washington, DC, USA, 2004.

[30] D. Huang and D. Medhi. Secure pairwise key establishment in large- scale sensor networks: An area partitioning and multigroup key predistribu- tion approach. TOSN, 3(3):16:1–16:34, 2007.

[31] S. Hussain, F. Kausar, and A. Masood. An eficient key distribution scheme for heterogeneous sensor networks. In Mohsen Guizani, Hsiao-Hwa Chen, and Xi Zhang, editors, IWCMC, p.p. 388–392. ACM, 2007.

[32] F. Ian, Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. Computer Networks, 38(4):393– 422, 2002.

[33] J. Jang, T. Kwon, and J. Song, A Time-Based Key Management Protocol for Wireless Sensor Networks. ISPEC 2007, LNCS 4464: 314–328, 2007.

[34] L. Jooyoung, D. R. Stinson, A combinatorial approach to key predistribution for distributed sensor networks. In IEEE Wireless Communications and Networking Conference, WCNC 2005, New Orleans, LA, USA, pages 1200–1205, 2005.

[35] R. Kalidindi, R. Kannan, S.S. Iyengar, and A. Durresi. Sub-grid based key vector assignment: A key pre-distribution scheme for distributed sensor networks. Journal of Pervasive Computing and Communications, 2(1):35–43, 2006.

[36] K. Kalkan, S. Yilmaz, O.Z. Yilmaz, and A. Levi. A highly resilient and zone-based key pre distribution protocol for multiphase wireless sensor networks. In Q2SWinet'09, p.p. 29-36, October, 2009.

[37] L. Lamport, 1981. Password Authentication with Insecure Communication. Commun, of the ACM, 24(11), pp. 770-772, November 1981.

[38] J. Lee, and D. R. Stinson, Deterministic key pre-distribution schemes for distributed sensor networks. ACM symposium on Applied Computing 2004, Lecture notes in computer science, 3357: 294–307,Waterloo, Canada, 2004.

[39] D. Liu, and P. Ning, Location-Based Pair-wise Key Establishment for Static Sensor Networks. 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003a.

[40] D. Liu, and P. Ning. Establishing Pair-wise Keys in Distributed Sensor Networks. 10th ACM Conference on Computer and Communications Security CCS'03, 2003b.

[41] D. Liu, and P. Ning, Improving key pre-distribution with deployment knowledge in static sensor networks. ACM Transactions on Sensor Networks, 1(2): 204–39, 2005a.

[42] D. Liu, P. Ning, R. Li, Establishing Pairwise Keys in Distributed Sensor Networks. ACM Transactions on Information and System Security, 8 (1): 41–77, 2005b.

[43] A. Liu, P. Ning, Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor net- works. Information Processing in Sensor Networks, IPSN'08. International Conference on, IEEE, p.p. 245-256, 2008.

[44] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, ACM Conference on Computer and Communications Security, pages 52–61. ACM, 2003.

[45] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In Sanjeev Setia and Vipin Swarup, editors, SASN, pages 72–82. ACM, 2003.

[46] B. Maala, Y. Challal, and A. Bouabdallah. HERO: Hi- erarchical key management protocol for heterogeneous wireless sensor net-works. Wireless Sensor and Actor Networks II, 264:125–136, 2008.

[47] K. M. Martin, M. B. Paterson, and D. R. Stinson. Key predis- tribution for homogeneous wireless sensor networks with group deployment of nodes, 2008. Available at ePrint Cryptology archive 2008/412.

[48] A. Mohaisen, Y. Maeng, and D. Nyang. On grid-based key pre-distribution: Toward a better connectivity in wireless sensor net-work. In Takashi Washio, Zhi-Hua Zhou,

Joshua Zhexue Huang, Xiaohua Hu, Jinyan Li, Chao Xie, Jieyue He, Deqing Zou, Kuan-Ching Li, and M´ario M. Freire, editors, PAKDD Workshops, volume 4819 of Lecture Notes in Com- puter Science, pages 527–537. Springer, 2007.

[49] L. B. Oliveira, A. C. Ferreira, M. Aur´elio Vilac¸a, H. C. Wong, M. W. Bern, R. Dahab, and A. A. Fer- reira Loureiro. SecLEACH - On the security of clustered sensor networks. Signal Processing, 87(12):2882–2895, 2007.

[50] L. B. Oliveira, H. C. Wong, M. W. Bern, R. Dahab, and A. A. F. Loureiro. SecLEACH - A random key distribu- tion solution for securing clustered sensor networks. In NCA, pages 145–154. IEEE Computer Society, 2006.

[51] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar. SPINS: security protocols for sensor netowrks. In MOBICOM, pages 189–199, 2001.

[52] R. Pietro, L. Mancini, Y. Law, S. Etalle, and P. Havinga, LKHW: a directed diffusion- based secure multicast scheme for wireless sensor networks. 1st International Workshop on Wireless Security and Privacy (WiSPr 03), 2003.

[53] M. G. Sadi, D. S. Kim, and J. S. Park. GBR: Grid based random key predistribution for wireless sensor network. In ICPADS (2), pages 310–315. IEEE Computer Society, 2005.

[54] K. Simonova, A. C. H. Ling, and X. S. Wang. Location- aware key predistribution scheme for wide area wireless sensor networks. In Sencun Zhu and Donggang Liu, editors, SASN, pages 157–168. ACM, 2006.

[55] D. Stinson, Combinatorial designs: Construction and Analysis. Springer, 2004.

[56] H. Willard, Clatworthy. Tables of Two-Associate-Class Partially Balanced Designs. NBS Applied Mathematics Series, vol. 63, 1973.

[57] M. F. Younis, K. Ghumman, and M. Eltoweissy. Location-aware combinatorial key management scheme for clustered sensor networks. IEEE Trans. Parallel Distrib. Syst., 17(8):865–882, 2006.

[58] Z. Yu, Y. Guan, A robust group-based key management scheme for wireless sensor networks. IEEE Wireless Communications and Networking Conference (WCNC 2005), New Orleans, LA USA, 2005.

[59] Z. Yu and Y. Guan. A key management scheme using deployment knowledge for wireless sensor networks. IEEE Transactions of Parallel and Distributed Systems, 19(10):1411–1425, 2008.

[60] W. Zhang, M. Tran, S. Zhu, and G. Cao, A Random Perturbation-Based Scheme for Pairwise Key Establishment in Sensor Networks. MobiHoc'07, Montréal, Québec, Canada, 2007.

[61] L. Zhou, J. Ni, and C. V. Ravishankar. Supporting secure com- munication and data collection in mobile sensor networks. In INFOCOM. IEEE, 2006.

[62] S. Zhu, S. Setia, and S. Jajodia. LEAP: eficient security mechanisms for large-scale distributed sensor networks. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, ACM Conference on Com- puter and Communications Security, pages 62–72. ACM, 2003.

[63] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach. In ICNP, pages 326–335. IEEE Computer Society, 2003.