M.Tech. (Computer Science) Dissertation Series

Grid-Based Key Agreement Protocols For Wireless Sensor Network

M.Tech. Dissertation Report

a dissertation submitted in partial fulfillment of the requirements for the M.Tech.(computer Science) degree of the Indian Statistical Institute

By

Ashish Singh
M.Tech-Computer Science
Roll No. - CS0719

under the supervision of

Prof. Rana Barua
Stat-Math Unit
ISI, Kolkata

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road,
Kolkata-700108, West Bengal, India.

Acknowledgments

With great pleasure and sense of obligation I express my heartfelt gratitude to my guide **Prof. Rana Barua** (Stat-Math Unit) . I am highly indebted to him for his invaluable guidance and his readiness for anytime help. Their persisting encouragement, everlasting patience and excellent expertise in subject have benefited to an extent, which is beyond expression.

I also want to thank to Shashank Singh, Vishnu Vadhan Reddy. B and Manoj Kumar Nanda(MTech, ISI Kolkata) for their motivation and encouragement.

Without wasting this valuable chance, I want to thank my family members, classmates, and friends for their consistent support.

And lastly I want to thank my parents and God for their consistent flow of energy by which I am able to do anything world.

Ashish Singh
July, 2009

Contents

Chapter 1

Introduction to Security in Sensor Network

In sensor network security, an important challenge is to design the protocols for secure communications of sensor key from a collection of sensor nodes, which may have been pre-loaded with some secret information data but have no prior direct communication with each other. Also protocol should allow nodes deployed at a later time to join the network securely. The difficulty of designing such protocols increases due to numerous limitations of sensor networks. We discuss these limitations in detail in section 1.2. some of them are due to inability to utilize existing public key cryptosystems (since the expensive computations involved could expose the power-constrained nodes to a denial-of-service attack), the inability to pre-determine which nodes will be neighbours after node deployment in sensor network, and the inability of any node to put absolute trust in its neighbour (as nodes are not tamper resistant and are vulnerable to physical capture).

Wireless sensor networks and key distribution protocols have few requirement to fulfil due to constrained on sensor nodes such as,

1. *Scalability* - WSNs and key distribution protocols must be able to support a larger network and must be flexible against substantial increase in the size of the network even after node deployment in sensor network.

2. *Efficiency* - Key distribution protocols must be able to fulfil storage, processing and communication limitations of sensor nodes.

3. *Key connectivity* - Probability that two (or more) sensor nodes are able to compute direct communication key, gives connectivity in the network. Enough key connectivity must be provided for a WSN to perform its intended functionality.

4. *Resilience* - key distribution protocols should be highly resistive against node capture. Usually higher resilience means lower number of compromised links.

1.1 Sensor network architecture

A typical sensor network has hundreds to several thousand sensor nodes. Each sensor node is typically low-cost, limited in computation and information storage capacity, highly power constrained, and communicates over a short-range wireless network interface. Most sensor networks have a base station that acts as a gateway to associated infrastructure such as data processing computers. Individual sensor nodes communicate locally with neighbouring sensors, and send their sensor readings over the peer-to-peer sensor network to the base station. Sensors can be deployed in various ways, such as physical installation of each sensor node, or random aerial scattering from an air-plane.

In general, sensor nodes communicate over a wireless network. A typical sensor network forms around one or more base stations, which connect the sensor network to the outside network.
The communication patterns within a sensor network fall into four categories:

1. Node to node communication,

2. Node to base station communication,

3. Base station to node communication and

4. Base station to base station communication.

Size of sensor network and deployment density of sensor nodes in the network depends on application. In this report, sensor network under consideration is very large and nodes have high connectivity in the network.

1.2 Limitations of WSNs

In the following paragraph, we will discuss the limitation of wireless sensor network in detail. These limitations of sensor network makes it very difficult and highly challenging to design a secure key establishment protocol for sensor network.

- *Impracticality of public key cryptosystems* - The limited computation, storage and power resources of sensor nodes makes it infeasible and impractical to use public-key cryptosystem, such as Diffie-Hellman key agreement protocol,and RSA signatures scheme.

- *Vulnerability of nodes to physical capture* - In many applications, sensor nodes need to be deployed in hostile environment that may cause various types of attacks on sensor nodes. Furthermore, the large number of sensor nodes that are deployed in the network makes it impractical and uneconomical that each sensor node are tamper-resistant. This exposes sensor nodes to physical attacks by an adversary. And an adversary may obtain the keying material stored in the node.

- *Lack of a-priori knowledge of post-deployment configuration* - If a sensor network is deployed via random scattering (e.g. from an air plane), the sensor network protocols cannot know beforehand which nodes will be within communication range of each other after deployment.Even if the nodes are deployed by hand, the large number of nodes involved makes it costly to pre-determine the location of every individual node. Hence, a security protocol should not assume prior knowledge of which nodes will be neighbours in a network

- *Limited storage resources* - Storage memory of sensor node usually includes flash memory and RAM. Flash memory is used for storing downloaded application code and RAM is used for storing application programs, sensor data, and intermediate computations. The amount of available key-storage memory in sensor nodes is usually low. it does not possess enough resources to establish unique keys with every one of the other nodes in the network.

- *Limited bandwidth and transmission power* - Typical sensor network platforms have very low bandwidth. Therefore, low transmission reliability makes communication of large blocks of information data very expensive. Also communication range of sensor nodes is limited by the need to conserve energy.

- *Over-reliance on base stations exposes vulnerabilities* - In a sensor network, base stations are very less in numbers and expensive. so it may be tempting to rely on them as a source of trust. However, this invites attack on the base station and limits the application of the security protocol.

- *Limited Computation Power* - The processors embedded in sensor nodes are usually not as powerful as those in nodes of a wired or ad hoc network. So these less powerful processors cannot be used for complex cryptographic algorithms in WSNs.

1.3 Attack Model in Sensor Network

Sensor networks have many characteristics that make them more vulnerable to attack than conventional computing equipment. In WSNs, it is usually assumed that an attacker may know the security mechanisms that are deployed in a sensor network. Attacker may be able to compromise a node or even physically capture a node. It is economically infeasible to deploy tamper resistant sensor nodes in sensor network, so generally, sensor nodes are non-tamper resistant. Also, once a node is compromised, attacker is capable of accessing the key materials stored within that node. Base stations in WSNs are usually considered as trustworthy. Most protocols focus on secure routing between sensors and the base station.

Attacks in sensor networks can be classified into the following categories:

1. *Node capture attack* - We assume that an adversary can have physical access on a sensor node after it is deployed and can get secret information from its memory. Then adversary can use this information to compute the secret stored in other sensor nodes or the secret key used for secure communication by other non-compromised node.

2. *Node replication attack* - In this attack model an adversary can insert additional hostile nodes into the network after getting some secret information (e.g. through node capture or infiltration). This is a serious attack since the compromise of even a single node might allow an adversary to populate the network with clones of the captured node to such an extent that legitimate nodes could be outnumbered and the adversary can thus gain full control of the network.

3. *Outsider versus insider attacks* - Outside attacks are defined as attacks from nodes which do not belong to a sensor network. Inside attacks occur when legitimate nodes of a sensor network behave in unintended or unauthorized ways.

4. *Passive versus active attacks* - Passive attacks include eavesdropping on or monitoring packets exchanged within a WSN. Active attacks involve some modifications of the data steam or the creation of a false stream.

5. *Mote-class versus laptop-class attacks* - In mote-class attacks, an adversary attacks a sensor network by using a few nodes with similar capabilities to the network nodes. In laptop class attacks, an adversary can use more powerful devices such as a laptop to attack a sensor network. These devices have greater transmission range, processing power, and energy reserves than the network nodes.

1.4 Security Measure for a Key Establishment Protocol

The aim of security services in sensor network, is to secure the message information, keying material and resources from attacks and misuse. The security requirements in WSNs include:

1. *Availability:* This ensures that the desired network services are available even in the presence of denial of service attacks.

2. *Authorization:* This security measure ensures that only authorized sensor nodes can be involved in providing information to network services.

3. *Authentication:* This security measure ensures that the communication from one node to another node is genuine. That is, a malicious node cannot masquerade as a trusted network node.

4. *Confidentiality:* This ensures that a given message cannot be understood by anyone other than the desired recipients.

5. *Integrity:* This ensures that a message sent from one node to another is not modified by malicious intermediate nodes.

6. *Non-repudiation:* This security measure defines that a node cannot deny sending a message it has previously sent.

7. *Freshness:* This ensures that the data is recent and ensures that no adversary can replay old messages.

8. *Forward secrecy:* This security parameter ensures that a sensor node should not be able to read any future messages after it leaves the network.

9. *Backward secrecy:* This security parameter ensures that a newly joining sensor node should not be able to read any previously transmitted message.

Chapter 2

Related Works

2.1 Key Pre-distribution in WSNs

In this report, we develop a key pre-distribution protocol to deal with the above problems. In order to study the new key distribution protocol. We first present a few protocols for pairwise key establishment,

1. Polynomial Based key Pre-distribution Scheme,
2. Polynomial Pool-Based Key Pre-distribution Scheme,
3. Grid-Based key Pre distribution Scheme, and
4. Multivariate Symmetric Polynomial base Key Pre-distribution Scheme.

2.1.1 Polynomial Based Key Pre-distribution

Here in this section, we briefly discuss the Polynomial-based key pre-distribution protocol of [2]. This protocol in [2] was developed for group key pre-distribution. We only discuss the pair-wise key establishment protocol in the context of sensor networks.
To pre-distribute pairwise keys, the (key) set-up server randomly generates a bivariate t-degree polynomial

$f(x, y) = \sum_{i=1}^{t} \sum_{j=1}^{t} a_{i,j}\, x^i y^j$ over a finite field F_q,

where q is a prime number and is large enough to accommodate a cryptographic key, also f(x,y) is symmetric in x and y i.e.

$f(x, y) = f(y, x)$

It is assumed that each sensor node has a unique ID.

For each node i (i is the ID of sensor node), the set-up 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, 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 $f(i, y)$, which occupies $(t + 1)\, log\, q$ storage space. To establish a pairwise key, both sensor nodes need to evaluate the polynomial at the ID of the other sensor node. There is no communication overhead during the pairwise key establishment process. The security proof in ref. [2] ensures that this scheme is unconditionally secure and t-collusion resistant. That is, a coalition of no more than t compromised sensor nodes does not know anything about the pairwise key between any two non-compromised nodes.

2.1.2 Polynomial Pool-based Key Pre-distribution

The polynomial pool- based key pre-distribution is inspired by the studies in refs.[3] and [4]. The basic idea can be considered as the combination of the polynomial-based key pre-distribution and the key pool idea used in refs. [3] and [4]. Polynomial pool- based key pre distribution scheme has three phases to establishment pairwise key:

1. Setup,
2. Direct key establishment, and
3. Path-key establishment

The setup phase is performed to initialize the nodes by distributing polynomial shares to them. Direct key establishment phase is performed if two sensor nodes need to establish a pairwise key. Path-key establishment is performed if two sensor nodes are not able to establish direct key, then they try to establish a pairwise key with the help of other sensor nodes.

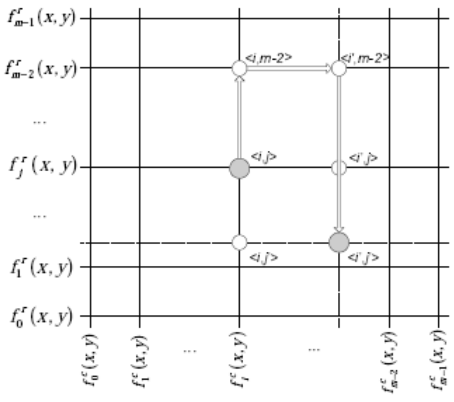
- **Phase 1: Setup -** The setup server randomly generates a set F of bivariate t-degree polynomials over the finite field F_q. To identify different polynomials, the setup server may assign each polynomial a unique ID. For each sensor node i, the setup server picks a subset of polynomials $F_i \subseteq F$, and assigns the shares of these polynomials to node i. The main issue in this phase is the subset assignment problem, which specifies how to pick a subset of polynomials from F for each sensor node.
- **Phase 2: Direct Key Establishment** A sensor node starts phase 2 if it needs to establish a pairwise key with another node. If both sensor nodes have shares on the same bi-variate polynomial, they can establish the pairwise key directly using the polynomial-based key pre-distribution. The main issue in this phase is the polynomial share discovery problem, which specifies how to find a common bivariate polynomial, of which both nodes have polynomial shares. For convenience, we say two sensor nodes have a secure link if they can establish a

pairwise key through direct key establishment. A pairwise key established in this phase is called a direct key.

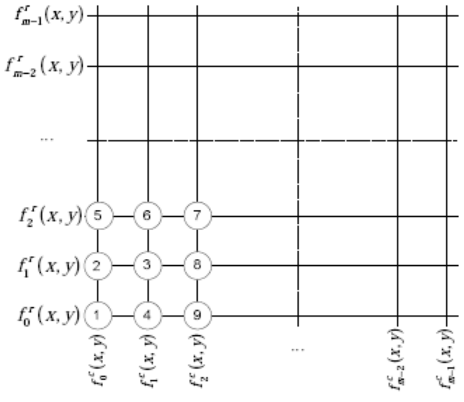
- **Phase 3: Path-Key Establishment -** If direct key establishment fails, two sensor nodes need to start phase 3 to establish a pairwise key with the help of other sensor nodes. To establish a pairwise key with node j, a sensor node i needs to find a sequence of nodes between itself and node j such that any two adjacent nodes in this sequence can establish a direct key. Such a sequence of nodes is called key path (or simply a path), since the purpose of such a path is to establish a pairwise key. Then either node i or j initiates a key establishment request with the other node through the intermediate nodes along the path. A pairwise key established in this phase is called an indirect key. A subtle issue is that two adjacent nodes in the path may not be able to communicate with each other directly. This framework assumes that they can always discover a route between themselves so that the messages from one node can reach the other. The main issue in this phase is the path discovery problem, which specifies how to find a path between two sensor nodes.

2.1.3 Grid-Based Key Pre-distribution

This scheme is based on a generalized Key Pre-distribution scheme in ref. [2], [12]. This scheme consider that a sensor network has at most N sensor nodes. The grid-based pre-distribution scheme constructs an $m \times m$ grid and generates 2m - bi-variate polynomials $\{f_i^c(x,\, y), f_i^r(x,\, y)\}_{i=0,1,2,...,m-1}$, where $m = \lceil \sqrt{N} \rceil$. As shown in the Figure , each row i in the grid is associated with a polynomial $f_i^r(x,\, y)$, and each column j is associated with a polynomial $f_j^c(x,\, y)$. The set-up server assigns each sensor node in the network to a unique non-occupied (i, j) coordinate in this grid. For the node at the coordinate (i, j), the set-up server distributes the polynomial shares of $f_i^c(x,\, y)$ and $f_j^r(x,\, y)$ to this node. As a result, sensor nodes can perform share discovery and path discovery based on this information.



(a) The grid



(b) An example order of node assignment

The ID's constructed from the coordinate (i, j) are represented as $\langle i,\, j \rangle$. This schema works in three Phases:

1. Subset Assignment
2. Polynomial Share Discovery
3. Path Discovery

Now we will describe these three phases in details in the following paragraphs.

Phase 1: **Subset Assignment-** The set-up server randomly generates 2m t-degree bi-variates symmetric polynomials $\{f_i^c(x, y), f_i^r(x, y)\}_{i=0,1,2,\ldots,m-1}$, over F_q. For each sensor node the set-up server chooses an unoccupied coordinate (i, j) in the grid and assigns it to the node with its polynomial share. So each sensor node with ID = $\langle i, j\rangle$ stores polynomial share $\{ID, f_i^c(j, y), f_j^r(i, y)\}$.

Phase 2: **Polynomial Share Discovery-** Let us consider that sensor nodes with ID's $\langle i_1, j_1\rangle$ and $\langle i_2, j_2\rangle$ want to establish pairwise key. So they check whether $i_1 = i_2$ or $j_1 = j_2$.

Case 1 : If $i_1 = i_2 = i(say)$ then both the sensor nodes have polynomial share $f_i^c(j_1, y)$ and $f_i^c(j_2, y)$ respectively of the symmetric bi-variate polynomial $f_{i_1}^c(x, y)$.
Then, sensor nodes can use the Polynomial-based key pre distribution scheme to establish the pairwise key directly between them.

Case 2 : If $j_1 = j_2$ then, similar to case 1, both the sensor nodes have polynomial share of $f_{j_1}^r(x, y)$.
And sensor nodes can use the Polynomial-based key pre-distribution scheme to establish the pairwise key directly between them similar to case 1.

Case 3 : If neither $i_1 = i_2$ nor $j_1 = j_2$ then both the sensor nodes use path discovery to establish a pairwise key.

Phase 3: **Path Discovery-** Nodes $\langle i_1, j_1\rangle$ and $\langle i_2, j_2\rangle$ need to do path discovery if neither $i_1 = i_2$ nor $j_1 = j_2$.

Sensor nodes $\langle i_1, j_1\rangle$ and $\langle i_1, j_2\rangle$ can establish a pairwise key using method similar to Case 1. Whereas node $\langle i_1, j_2\rangle$ and $\langle i_2, j_2\rangle$ can establish a pairwise key using method similar to Case 2.
In other words, node $\langle i_1, j_2\rangle$ can establish pairwise key with both the nodes $\langle i_1, j_1\rangle$ and $\langle i_2, j_2\rangle$. Similarly the node $\langle i_2, j_1\rangle$ can also establish the pairwise key with them.
It is guaranteed that there exist at least one node that can be used as an intermediate node between any two sensor node if there is no corrupted node in the network.

2.1.4 Multivariate Symmetric Polynomial base Key Pre-distribution Scheme

This scheme is based on a *t*-degree multivariate symmetric polynomial in ref. [7] & [8].

A t-degree (k + 1)-variate polynomial is defined as

$$f(x_1,\, x_2,\, \cdots\, x_k,\, x_{(k+1)}) = \sum_{i_1=0}^{t}\sum_{i_2=0}^{t}\cdots\sum_{i_k=0}^{t}\sum_{i_{k+1}=0}^{t} a_{i_1,i_2,\cdots i_k,i_{k+1}}\, x_1^{i_1} x_2^{i_2}\cdots x_k^{i_k} x_{k+1}^{i_{k+1}}$$

At first, every node should have k credentials, which are positive and pairwise different integers. Suppose node u has credentials $(u_1, u_2, ..., u_k)$ and node v has credentials $(v_1, v_2, ..., v_k)$. Before node deployment, setup server assign a polynomial share $f(u_1, u_2, ..., u_k, x_{k+1})$ to u and another share $f(v_1, v_2, ..., v_k, x_{k+1})$ to v. Assigning polynomial shares to sensor nodes means that the coefficients of t-degree uni-variate polynomials $f(u_1, u_2, ..., u_k, x_{k+1})$ and $f(v_1, v_2, ..., v_k, x_{k+1})$ are loaded into memory of nodes *u* and *v*, respectively.

If the credentials of node u and node v have only one element different, i.e.,

1. for some $i \in [1, k],\ u_i \neq v_i$, and
2. for j = 1, 2, . . . , k, $\ j \neq i,\ u_j = v_j = c_j(say)$,

then node u and node v can have a shared key. Node u can take v_i as the input to its own share $f(u_1, u_2, ..., u_k, x_{k+1})$, and node v can also take u_i as the input to its share $f(v_1, v_2, ..., v_k, x_{k+1})$. Due to the polynomial symmetry, the desired shared key between nodes u and v has been established as

$$\begin{aligned} K_{uv} &= f(c_1,\, c_2,\, \cdots,\, c_{i-1},\, u_i, c_{i+1}\cdots,\, c_k,\, v_i) \\ &= f(c_1,\, c_2,\, \cdots,\, c_{i-1},\, v_i, c_{i+1}\cdots,\, c_k,\, u_i) \end{aligned} \tag{2.1}$$

Here, node *u* and *v* achieve the key agreement by a t-degree bi-variate symmetric polynomial, i.e.,

$$f_i(x_i,\, x_{k+1},) = f(c_1,\, c_2,\, \cdots,\, c_{i-1},\, x_i, c_{i+1}\cdots,\, c_k,\, x_{k+1})$$

where $i \in \{1,\, 2,\, \cdots,\, k\}$

Chapter 3

2-Dimensional Grid-Based Key Establishment Protocol For WSNs

Now we will describe our key exchange scheme. This scheme uses trivariate symmetric polynomial for share pre-distribution. Symmetric polynomial ensures extra connectivity in the network. Let we consider a tri-variate symmetric t-degree polynomial given as

$$f(x_1,\, x_2,\, x_3) = \sum_{i_1=0}^{t}\sum_{i_2=0}^{t}\sum_{i_3=0}^{t} a_{i_1,i_2,i_3}\, x_1^{i_1} x_2^{i_2} x_3^{i_3}, \tag{3.1}$$

where all the polynomial coefficients are chosen from a finite field F_q, and q is either a prime number or a prime power, large enough to accommodate a cryptographic key.
Here in this chapter unless otherwise stated, all the calculations are done over the finite field F_q.
Now if we choose all the coefficients of the polynomial such that

$$a_{i_1,i_2,i_3} \;=\; a_{i_{\sigma(1)},i_{\sigma(2)},i_{\sigma(3)}} \tag{3.2}$$

for any permutation σ of $\{1, 2, 3\}$, where $\sigma\ :\{1,\ 2,\ 3\} \longmapsto \{1,\ 2,\ 3\}$ is a bijection, then we will obtain the symmetric polynomial i.e.

$$f(x_1,\, x_2,\, x_3) \;=\; f(x_{\sigma(1)},\, x_{\sigma(2)},\, x_{\sigma(3)}) \tag{3.3}$$

Let us consider that the sensor network has *N* sensor nodes. Also consider a two dimensional grid with *u* rows and *v* column. where *u* and *v* are integers such that $u \cdot v = N$, where $u < \sqrt{N}$ and $v > \sqrt{N}$.

The set-up server will assign each sensor node in the network to a unique non-occupied *(i, j)* co-ordinate in this grid. where *i* and *j* are row and column number in the grid respectively, such that $1 \leq i \leq u$ and $1 \leq j \leq v$. The ID of the sensor node associated with the coordinate (i,j) is represented by $\langle i,\ j\rangle$.

Now consider a set of credentials (positive integers) *C*, (usually, $C = \{1, 2, \cdots, v\}$. Below we will present a method to generate *C*) such that, *C* $= \{c_1, c_2, ... c_v\}$ and $|$*C*$|$ $= v$. We form a set S_1 from first *u* elements of *C* and another set $S_2 =$ *C* i.e.

$S_1 = \{c_1, c_2, ... c_u\}$ and $S_2 = \{c_1, c_2, ... c_v\}$.

Consider that x_1 takes values from the set S_1 and x_2 takes values from the set S_2 .

Now we will compute a bivariate symmetric polynomial for each row in the grid from the tri-variate symmetric t-degree polynomial i.e. each row *i* in the grid is associated with a $(k + i).t$ - degree bivariate symmetric polynomial $\{f(c_i, x_2, x_3)\}^{(k+i)}$, where *k* is a suitably choosen positive integer, such that $(k + 1).t \geq v$ and $(k + u).t$ is not very large integer.

Each sensor node $\langle i, j\rangle$ in the sensor network has a pair of credential, which are positive integers and denoted by (c_i, c_j), where $(c_i, c_j) \in S_1 \times S_2$. Before node deployment in the sensor network, a polynomial share $\{f(c_i, c_j, x_3)\}^{(k+i)}$ is distributed to each sensor node $\langle i, j\rangle$. By distributing the polynomial share to sensor node, we mean that for each sensor node $\langle i, j\rangle$ we store the coefficient of *(k + i).t* - degree univariate polynomial $\{f(c_i, c_j, x_3)\}^{(k+i)}$ into node memory.

Our key establishment schema works in Two Phases:

1. Polynomial Share Pre-distribution.
2. Key Establishment Mechanism
   (a) Direct Key Establishment.
   (b) Indirect Key Establishment.

3.1 Polynomial Share Pre-distribution

Polynomial Share Pre-distribution phase is performed prior to network deployment by a trusted set-up server. The set-up server generates a global tri-variate symmetric t-degree polynomial and a set *C* of credentials as described above. The set-up server will use this global polynomial and set of credentials to calculate the polynomial share for each sensor node.

Since each sensor node $\langle i, j\rangle$ has a pair of credentials (c_i, c_j), which are positive integers such that,

$$(c_i, c_j) \in S_1 \times S_2, \quad \text{where} \quad S_1 = \{c_1, c_2, c_3, \ldots, c_u\} \quad \text{and} \quad S_2 = C$$

The elements of set *C* can be preloaded into sensor nodes before deployments but it causes extra memory overhead. Therefore,we will generate the credentials by using a bijection Φ between nodes ID's and credentials. So that credentials can be derived from the node ID's. The function Φ

is defined as,

$$\begin{aligned}\Phi : \{1,\, 2,\, 3,\, ...,\, v\} &\longmapsto \{c_1, c_2,\, c_3,\, \ldots, c_v\} \\ s.t. \qquad c_i &= \Phi(i) \\ &= u + i \end{aligned} \tag{3.4}$$

where *u* and *v* are the row and column number in the grid respectively. Therefore each node needs to store only *u* instead of the pair of credentials.

Note : In the definition of function Φ, if we consider $u = 0$ then $c_i = i$ and then we do not need to store *u* also in each sensor node.

Now each sensor node $\langle i, j\rangle$ in the sensor network has a pair of credential $(c_i,\, c_j) = (u + i,\, u + j)$ and the polynomial share assigned to it by set-up server is computed as follows :

$$\begin{aligned}\{f(c_i,\, c_j,\, x_3)\}^{(k+i)} &= \{f(u+i,\, u+j,\, x_3)\}^{(k+i)} \\ &= \{\sum_{i_1=0}^{t}\sum_{i_2=0}^{t}\sum_{i_3=0}^{t} a_{i_1,i_2,i_3}\,(u+i)^{i_1}(u+j)^{i_2}\,x_3^{i_3}\}^{(k+i)}\end{aligned} \tag{3.5}$$

Therefore, every node in the sensor network is storing a *(k + i).t*-degree univariate polynomial having (*(k + i).t* + 1) coefficients over the finite field F_q. Before nodes deployment these coefficients are preloaded in the sensor nodes and are used for computing communication key during key establishment process.

3.2 Key Establishment

After node deployment in the sensor network two nodes can establish a communication key using their polynomial share. In our model there are two ways to establish a communication key between sensor nodes.

1. Direct Key Establishment.

2. Indirect Key Establishment.

Now we will present these two key establishment methods in details in following paragraphs.

3.2.1 Direct Key Establishment :-

Let us consider that sensor node $\langle i_1,\, j_1\rangle$ wants to establish a communication key with the sensor node $\langle i_2,\, j_2\rangle$. These two nodes can establish a direct communication key if they have a common credential. The credential associated with sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$ are $(c_{i_1},\, c_{j_1})$ and $(c_{i_2},\, c_{j_2})$ respectively.

We divide direct key communication process in two cases.

- **Case 1 -** Both sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ are in same row of the grid i.e. $i_1 \;=\; i_2$, therefore $c_{i_1} \;=\; c_{i_2} \;=\; c\,(say)$.

  Sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ has polynomial share $\{f(c,\, c_{j_1},\, x_3)\}^{(k+i_1)}$ and $\{f(c,\, c_{j_2},\, x_3)\}^{(k+i_1)}$ respectively. So sensor node $\langle i_1,\, j_1 \rangle$ calculates the credential c_{j_2} and communication key K_1 as follows,

  $$\begin{aligned} c_{j_2} &= \Phi(j_2) = u + j_2 \\ K_1 &= \{f(c,\, c_{j_1},\, c_{j_2})\}^{(k+i_1)} \end{aligned} \tag{3.6}$$

  Similarly, sensor node $\langle i_2,\, j_2 \rangle$ calculates the credential c_{j_1} and communication key K_2 as follows,

  $$\begin{aligned} c_{j_1} &= \Phi(j_1) = u + j_1 \\ K_2 &= \{f(c,\, c_{j_2},\, c_{j_1})\}^{(k+i_1)} \end{aligned} \tag{3.7}$$

  Since $f(x_1,\, x_2,\, x_3)$ is a symmetric tri-variate polynomial, therefore, from equation (3.6) & (3.7), we have

  $$\{f(c,\, c_{j_1},\, c_{j_2})\}^{(k+i_1)} \;=\; \{f(c,\, c_{j_2},\, c_{j_1})\}^{(k+i_1)} \Longrightarrow K_1 \;=\; K_2 \;=\; K\,,(say)$$

  Now using this key *K* sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ can communicate with each other securely.

- **Case 2 -** Both sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ are from different rows of the grid i.e. $i_1 \;\neq\; i_2$, therefore, $c_{i_1} \;\neq\; c_{i_2}$. We further divide this case in three sub-cases.

  1. **Case 2.1 -** Both sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ are from same column of the grid i.e. $i_1 \;\neq\; i_2,$ but $j_1 \;=\; j_2,$ therefore $c_{i_1} \;\neq\; c_{i_2}$ but $c_{j_1} \;=\; c_{j_2} \;=\; c\,(say)$.

     Sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ has polynomial share $\{f(c_{i_1},\, c,\, x_3)\}^{(k+i_1)}$ and $\{f(c_{i_2},\, c,\, x_3)\}^{(k+i_2)}$ respectively. So sensor node $\langle i_1,\, j_1 \rangle$ calculates the credential c_{i_2} and communication key K_1 as follows,

     $$\begin{aligned} c_{i_2} &= \Phi(i_2) = u + i_2 \\ K_1' &= \{f(c_{i_1},\, c,\, c_{i_2})\}^{(k+i_1)} \\ K_1 &= \{K_1'\}^{(k+i_2)} \;=\; \{\{f(c_{i_1},\, c,\, c_{i_2})\}^{(k+i_1)}\}^{(k+i_2)} \\ &= \{f(c_{i_1},\, c,\, c_{i_2})\}^{(k+i_1).(k+i_2)} \end{aligned} \tag{3.8}$$

Similarly, sensor node $\langle i_2,\, j_2\rangle$ calculates the credential c_{i_1} and communication key K_2 as follows,

$$\begin{aligned}
c_{i_1} &= \Phi(i_1) = u + i_1 \\
K_2' &= \{f(c_{i_2},\, c,\, c_{i_1})\}^{(k+i_2)} \\
K_2 &= \{K_2'\}^{(k+i_1)} = \{\{f(c_{i_2},\, c,\, c_{i_1})\}^{(k+i_2)}\}^{(k+i_1)} \\
&= \{f(c_{i_2},\, c,\, c_{i_1})\}^{(k+i_2).(k+i_1)} \qquad (3.9)
\end{aligned}$$

Since $f(x_1,\, x_2,\, x_3)$ is a symmetric tri-variate polynomial, therefore from equation (3.8) & (3.9), we have

$$\{f(c_{i_1},\, c,\, c_{i_2})\}^{(k+i_1).(k+i_2)} = \{f(c_{i_2},\, c,\, c_{i_1})\}^{(k+i_2).(k+i_1)}$$

$$\implies K_1 = K_2 = K\, , (say)$$

Now again as in case 1, using this key *K* sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$ can communicate with each other securely.

2. **Case 2.2 -** Sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$ are such that $i_1 \neq i_2$, but $i_1 = j_2, \implies c_{i_1} \neq c_{i_2}$ but $c_{i_1} = c_{j_2} = c\,(say)$.

   Sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$ has polynomial share $\{f(c,\, c_{j_1},\, x_3)\}^{(k+i_1)}$ and $\{f(c_{i_2},\, c,\, x_3)\}^{(k+i_2)}$ respectively. So, sensor node $\langle i_1,\, j_1\rangle$ calculates the credential c_{i_2} and communication key K_1 as follows,

$$\begin{aligned}
c_{i_2} &= \Phi(i_2) = u + i_2 \\
K_1' &= \{f(c,\, c_{j_1},\, c_{i_2})\}^{(k+i_1)} \\
K_1 &= \{K_1'\}^{(k+i_2)} = \{\{f(c,\, c_{j_1},\, c_{i_2})\}^{(k+i_1)}\}^{(k+i_2)} \\
&= \{f(c,\, c_{j_1},\, c_{i_2})\}^{(k+i_1).(k+i_2)} \qquad (3.10)
\end{aligned}$$

   Similarly, sensor node $\langle i_2,\, j_2\rangle$ calculates the credential c_{j_1} and communication key K_2 as follows,

$$\begin{aligned}
c_{j_1} &= \Phi(j_1) = u + j_1 \\
K_2' &= \{f(c_{i_2},\, c,\, c_{j_1})\}^{(k+i_2)} \\
K_2 &= \{K_2'\}^{(k+i_1)} = \{\{f(c_{i_2},\, c,\, c_{j_1})\}^{(k+i_2)}\}^{(k+i_1)} \\
&= \{f(c_{i_2},\, c,\, c_{j_1})\}^{(k+i_2).(k+i_1)} \qquad (3.11)
\end{aligned}$$

   Since $f(x_1,\, x_2,\, x_3)$ is a symmetric tri-variate polynomial, therefore, from equation (3.10) & (3.11), we have

$$\{f(c,\, c_{j_1},\, c_{i_2})\}^{(k+i_1).(k+i_2)} = \{f(c_{i_2},\, c,\, c_{j_1})\}^{(k+i_2).(k+i_1)}$$

$$\implies K_1 \,=\, K_2 \,=\, K\,, (say)$$

Similar to case 1, using this key *K* sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ can communicate with each other securely.

3. **Case 2.3 -** Sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ are such that $i_1 \,\neq\, i_2,$ but $j_1 \,=\, i_2,$ therefore $c_{i_1} \,\neq\, c_{i_2}$ but $c_{j_1} \,=\, c_{i_2} \,=\, c\,(say).$

   Sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ has polynomial share $\{f(c_{i_1},\, c,\, x_3)\}^{(k+i_1)}$ and $\{f(c,\, c_{j_2},\, x_3)\}^{(k+i_2)}$ respectively. So, sensor node $\langle i_1,\, j_1 \rangle$ calculates the credential c_{j_2} and communication key K_1 as follows,

$$\begin{aligned} c_{j_2} &= \Phi(j_2) = u + j_2 \\ K_1' &= \{f(c_{i_1},\, c,\, c_{j_2})\}^{(k+i_1)} \\ K_1 &= \{K_1'\}^{(k+i_2)} \,=\, \{\{f(c_{i_1},\, c,\, c_{j_2})\}^{(k+i_1)}\}^{(k+i_2)} \\ &= \{f(c_{i_1},\, c,\, c_{j_2})\}^{(k+i_1).(k+i_2)} \end{aligned} \tag{3.12}$$

   Similarly, sensor node $\langle i_2,\, j_2 \rangle$ calculates the credential c_{i_1} and communication key K_2 as follows,

$$\begin{aligned} c_{i_1} &= \Phi(i_1) = u + i_1 \\ K_2' &= \{f(c,\, c_{j_2},\, c_{i_1})\}^{(k+i_2)} \\ K_2 &= \{K_2'\}^{(k+i_1)} \,=\, \{\{f(c,\, c_{j_2},\, c_{i_1})\}^{(k+i_2)}\}^{(k+i_1)} \\ &= \{f(c,\, c_{j_2},\, c_{i_1})\}^{(k+i_2).(k+i_1)} \end{aligned} \tag{3.13}$$

   Using symmetric property of the polynomial $f(x_1,\, x_2,\, x_3)$, and equation (3.12) & (3.13), we have

$$\{f(c_{i_1},\, c,\, c_{j_2})\}^{(k+i_1).(k+i_2)} \,=\, \{f(c,\, c_{j_2},\, c_{i_1})\}^{(k+i_2).(k+i_1)}$$

$$\implies K_1 \,=\, K_2 \,=\, K\,, (say)$$

   Similar to case 1, using this key *K* sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ can communicate with each other securely.

4. **Case 2.4 -** Sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ are such that $i_1 \,\neq\, i_2,$ but $i_1 \,=\, j_2,$ and $j_1 \,=\, i_2,$ therefore $c_{i_1} \,\neq\, c_{i_2}$ but $c_{i_1} \,=\, c_{j_2}$ and $c_{j_1} \,=\, c_{i_2}.$

   This case is an special case of Case 2.2 and Case 2.3, therefore, using either method sensor nodes $\langle i_1,\, j_1 \rangle$ and $\langle i_2,\, j_2 \rangle$ can establish the communication key for secure communication.

3.2.2 Indirect Key Establishment :-

Let we consider that sensor node $\langle i_1,\, j_1\rangle$ wants to establish a communication key with the sensor node $\langle i_2,\, j_2\rangle$ where $i_1 \neq i_2, i_1 \neq j_2, j_1 \neq i_2$, and $j_1 \neq j_2$. Therefore, sensor node $\langle i_1,\, j_1\rangle$ cannot establish a direct key with sensor node $\langle i_2,\, j_2\rangle$. So sensor node $\langle i_1,\, j_1\rangle$ search for some other sensor node $\langle i',\, j'\rangle$ in the sensor network such that sensor node $\langle i_1,\, j_1\rangle$ and sensor node $\langle i_2,\, j_2\rangle$ can directly communicate with the sensor node $\langle i',\, j'\rangle$. Therefore, using sensor node $\langle i',\, j'\rangle$ as an intermediate node, the two sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$ can establish a communication key.

It is easy to show that if there is no compromised node in the network then for any pair of sensor nodes there will always exist at-least one intermediate node for indirect communication. We will show that there are eight nodes for any pair of sensor nodes which can be used as an intermediate node. And if there are compromised nodes in the network then by using more than one non-compromised intermediate nodes, it is always possible to establish a communication key.

Now we determine the condition on a sensor node $\langle i',\, j'\rangle$ for playing the role of intermediate node for sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$.

1. Either $i_1 = i'$, or $i_1 = j'$, or $j_1 = i'$, or $j_1 = j'$, and
2. Either $i_2 = i'$, or $i_2 = j'$, or $j_2 = i'$, or $j_2 = j'$.

Corresponding to each of the four choices in condition (a), there are two choices in condition (b). Therefore, there are total eight choices for intermediate nodes between sensor nodes $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$.

If node $\langle i',\, j'\rangle$ satisfies one of the conditions given in (a) then sensor node $\langle i',\, j'\rangle$ can directly communicate to sensor node $\langle i_1,\, j_1\rangle$ using method for direct key establishment. Similarly if node $\langle i',\, j'\rangle$ satisfies one of the conditions given in (b) then node $\langle i',\, j'\rangle$ can directly communicate to sensor node $\langle i_2,\, j_2\rangle$ using method for direct key establishment. Therefore sensor node $\langle i',\, j'\rangle$ can be used as an intermediate node for sensor node $\langle i_1,\, j_1\rangle$ and $\langle i_2,\, j_2\rangle$.

3.2.3 Network model

Our key agreement protocol is a deterministic key agreement model i.e. using deterministic method, it can establish a communication key between any pair of sensor nodes. And not only that, using deterministic method a sensor node can determine the ID's of other sensor nodes to which it can directly establish the communication key. There are pairs of sensor nodes in the network that cannot compute direct communication key. So if two nodes in the network cannot calculate direct key, then they search one intermediate node to establish an indirect key. If no node of the sensor network is corrupted then it can be guaranteed that for each pair of sensor node there is at-least one intermediate node. Here we assume that the underlying routing protocol can correctly route key establishment messages over multi-hop paths between peer nodes.

In fact even if there are some corrupted node in the network then also using more than one intermediate node, any non-corrupted pair of sensor node can establish an indirect communication key.

3.2.4 Adversary Model

In a sensor network, radio communications are of broadcast type in nature. So an adversary can easily tamper any message broadcast over the air between the sensor nodes. Also adversary may have easy physical access to the sensor network. In that case an adversary can capture sensor nodes in the network and tamper the nodes polynomial share. Though it is possible to use tamper resistant hardware for storing polynomial share to reduce the risk, but this increases the cost and energy consumption of each sensor node. Therefore, it is infeasible and un-economical to use tamper resistant hardware to secure the polynomial share for each sensor node. Even if we use tamper resistant hardware then also it may not provide perfect security.
An adversary can use the information obtained from a compromised node to compute other node's shares. Therefore, node compromise attack is not avoidable. So our aim is to reduced the impact of node compromised attack. For this we will try to reduced the probability of exposing the polynomial share of non-compromised nodes when some nodes have already been compromised.

3.3 Security Analysis and Choice of Parameters

In this section we will analyze the security performance of our model and compute memory cost for each sensor node, node resilience, compromise attack, in the network and computation energy cost.

3.3.1 Node Compromise Attack

Because of hostile nature of deployment area, sensor nodes may have to face wide variety of malicious attack. Also if an adversary gets physical access to sensor nodes then he/she may tamper the node and may get the share information of that node. Using this share information he/she may try to discover the secret communication key of other pair of sensor nodes. Here we will analyze the effect of node compromise attack at row level, column level and network level.

Row Level Node Compromise Attack

Let us consider row level compromise attack in the network. Any pair of nodes in row *i* can establish communication key by a symmetric bi-variate polynomial of degree *(k + i).t*.

$$\{f(c_i,\, x_2,\, x_3)\}^{(k+i)},$$

where c_i is the common credential between the pair of nodes.
Now to compromise the pairwise key without compromising the pair of nodes from *ith*$\{i = 1,\, 2,\, 3, \cdots,\, u\}$ row adversary needs to compromise the shared polynomial $\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}$

between the nodes. Also, $\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}$ is a *(k + i).t*-degree bi-variate symmetric polynomial, therefore, an adversary needs to compromise at-least *((k + i).t + 1)* nodes. And each row in the grid holds *v* nodes. So to ensure that pairwise key between any pair of nodes from any row is unsolvable by other *(v - 2)* nodes from same row, degree of the polynomial must satisfy,

$$0 \leq (v-2) \leq (k+1).t \tag{3.14}$$

Now we consider that in a row level attack an adversary may compromises all *v* sensor nodes of i^{th} row. So, adversary can construct $\dfrac{v.(v+1)}{2}$ equations, given by

$$\begin{array}{rcl}
\{f(c_i,\, c_1,\, c_1)\}^{(k+i)} & = & K_{i,\,(1,1)} \\
\{f(c_i,\, c_1,\, c_2)\}^{(k+i)} & = & K_{i,\,(1,2)} \\
\vdots & \vdots & \vdots \\
\{f(c_i,\, c_1,\, c_v)\}^{(k+i)} & = & K_{i,\,(1,v)} \\
\\
\{f(c_i,\, c_2,\, c_2)\}^{(k+i)} & = & K_{i,\,(2,2)} \\
\{f(c_i,\, c_2,\, c_3)\}^{(k+i)} & = & K_{i,\,(2,3)} \\
\vdots & \vdots & \vdots \\
\{f(c_i,\, c_1,\, c_v)\}^{(k+i)} & = & K_{i,\,(2,v)} \\
\vdots & \vdots & \vdots \\
\{f(c_i,\, c_v,\, c_v)\}^{(k+i)} & = & K_{i,\,(v,v)},
\end{array}$$

where $K_i,\, (J_1,\, J_2)\, s.t.\, J_1 \neq J_2$ is secret communication key between node $J_1,\, and\, J_2$ of i^{th} row of the grid. By solving these $\dfrac{v.(v+1)}{2}$ equations, an adversary may determine the bi variate polynomial share $\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}$ for first row of the grid i.e. $i = 1$, where as for other values of *i* i.e. for $i > 1$, we have, $0 \leq (v-2) < (k+i), t$ so adversary cannot compute the bi-variate symmetric polynomial share associated with i^{th} row s.t $i > 1$,

But to get the base polynomial share $f(c_i,\, x_2,\, x_3)$ by the adversary. He/She must have to find $(k+i)^{th}$ root of the polynomial $\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}$,

Column Level Node Compromise Attack

Now we will consider column level compromise attack in the network. Each sensor node in i^{th} column is from different row of the grid, so each sensor node in i^{th} column has polynomial share from different bi-variate symmetric polynomial.

Example : consider the sensor nodes $\langle i_1,\, j\rangle$ and $\langle i_2,\, j\rangle$ from j^{th} column of the grid. Node $\langle i_1,\, j\rangle$ has polynomial share from bi-variate polynomial $\{f(x_{i_1},\, c_j,\, x_3)\}^{(k+i_1)}$ whereas, node $\langle i_2,\, j\rangle$

has polynomial share from bi-variate polynomial $\{f(x_{i_2},\, c_j,\, x_3)\}^{(k+i_2)}$.

Therefore, compromising some nodes in a column adversary cannot affect the pairwise key for the other non-compromised pairs of nodes from the same column.

Now if an adversary compromises one node in j^{th} column (say) $\langle i,\, j\rangle$ then he/she can construct *v* equation given by,

$$\begin{aligned} \{f(c_i,\, c_j,\, c_1)\}^{(k+i)} &= K_{i,(j,1)} \\ \{f(c_i,\, c_j,\, c_2)\}^{(k+i)} &= K_{i,(j,2)} \\ \vdots \quad & \vdots \ \vdots \\ \{f(c_i,\, c_j,\, c_v)\}^{(k+i)} &= K_{i,(j,v)}, \end{aligned}$$

where $K_{i,(j,j_1)}$ $s.t.$ $j \neq j_1$ is the direct key between sensor nodes $\langle i,\, j\rangle$ and $\langle i,\, j_1\rangle$. Using these equations, he/she may form the uni-variate polynomial $\{f(c_i,\, c_j,\, x_3)\}^{(k+i)}$. But he/she cannot determine uni-variate polynomial $f(c_i,\, c_j,\, x_3)$. As to construct the uni-variate polynomial $f(c_i,\, c_j,\, x_3)$, he/she has to compute $(k+i)^{th}$ root of the polynomial $\{f(c_i,\, c_j,\, x_3)\}^{(k+i)}$. If we assume that adversary can compute the $(k+i)^{th}$ root of the polynomial then also he/she has to compromise all the node of i_{th} row of the grid in order to get bi-variate polynomial $f(c_i,\, x_2,\, x_3)$.

Also if we consider that adversary has captured all the nodes of one column then he/she can construct $\dfrac{u.(2v-u+1)}{2}$ equations. But since these equation are from *u* different bi-variate shared polynomials, therefore, adversary cannot determine them.

The number of distinct coefficient of a *t*-degree bivariate symmetric polynomial is $\binom{t+2}{2}$. So to compute those *u* bi-variate symmetric polynomial adversary need to compute on an average $u.\binom{(k+\lceil u/2\rceil).t+2}{2}$ coefficients, i.e. adversary needs $u.\binom{(k+\lceil u/2\rceil).t+2}{2}$ equations, so we will choose the variables *k, t* and *u* are such that

$$\binom{(k+\lceil u/2\rceil).t+2}{2} \geq \frac{(2v-u+1)}{2} \qquad (3.15)$$

Network Level Node Compromise Attack

Now we will consider network level node compromise attack. Consider that an adversary can compromises all the nodes of the sensor network. Total number of sensor nodes in the network is

$$N = u.v \qquad (3.16)$$

Therefore, total number of equation T_e that an adversary can construct is,

$$\begin{aligned} T_e &= u.\frac{v.(v-1)}{2} + v.\frac{u.(2v-u+1)}{2} \\ &= \frac{u.v}{2}(3v-u+2) \end{aligned} \tag{3.17}$$

Using these equation adversary needs to construct *u*- bivariate symmetric polynomial share namely,

$$\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}, \quad \forall i \,=\, 1,\, 2\, \cdots\, ,\, u$$

The number of distinct coefficient of a *t*-degree bivariate symmetric polynomial is $\binom{t+2}{2}$.

Therefore, to find these *u* -bivariate symmetric polynomial share, adversary needs to find the coefficient of these polynomials. So on an average adversary has to find $u.\binom{(k+\lceil u/2\rceil).t+2}{2}$ distinct coefficients.

So, adversary need $u.\binom{(k+\lceil u/2\rceil).t+2}{2}$ equations to compute all those *u* -bivariate symmetric polynomial shares. Therefore, for the security of these *u* bivariate symmetric polynomial shares, we will choose parameters *k, t, u* and *v* such that,

$$\begin{aligned} u.\binom{(k+\lceil u/2\rceil).t+2}{2} &\geq \frac{u.v}{2}(3v-u+2) \\ \implies \binom{(k+\lceil u/2\rceil).t+2}{2} &\geq \frac{v}{2}(3v-u+2) \\ \implies ((k+\lceil u/2\rceil).t+2).((k+\lceil u/2\rceil).t+1) &\geq v.(3v-u+2), \end{aligned}$$

so we will choose parameters k, t, u and v such that,

$$((k+\lceil u/2\rceil).t)^2 \geq 3.v^2 \tag{3.18}$$

3.3.2 Choice of parameters *k*, *t*, *u*, and *v*

In this protocol we have considered the parameters *k, t, u,* and *v*. where *u and v* are such that $u \cdot v = N$ also $u < \sqrt{N}$ and $v > \sqrt{N}$. We will do analysis for possible values of these parameters.

Security requirement for the protocol and storage limitation imposes following restriction.

1. Sensor nodes corresponding to first row of the grid are storing a uni-variate polynomial of degree $(k+1).t$. So from equation (3.14) we have $0 \leq (v-2) \leq (k+1).t$
2. Sensor nodes corresponding to last row of the grid are storing a uni-variate polynomial of degree $(k+u).t$. So we need to choose *u*, *k* and *t* such that $(k+u).t$ should not be very large.

Using these restriction we will choose the values of *u, v, k* and *t* such that they fulfill the security and storage requirements.

1. If we choose *u* and *v* such that $u \cdot v = N$ also $u < \sqrt{N}$ and $v > \sqrt{N}$. Then it can be shown that,

$$u + v \ > 2.\sqrt{N}$$

as, if $u \neq v$, then

$$\begin{aligned} (\sqrt{u} - \sqrt{v})^2 \ &> \ 0 \\ \implies \ u + v - 2.\sqrt{u.v} \ &> \ 0 \\ \implies \ (u+v) \ &> \ 2.\sqrt{u.v} \ = \ 2.\sqrt{N} \end{aligned}$$

2. The secrecy of the bi-variate polynomial $\{f(c_1,\, x_2,\, x_3)\}^{(k+1)}$ requires that, $0 \leq (v-2) \leq (k+1).t$, therefore, we will choose *k* and *t* such that

$$(k+1).t \ \geq \ v \ \implies \ t \ \geq \ \lceil \frac{v}{(k+1)} \rceil \ \approx \ \lceil \frac{v}{k} \rceil \tag{3.19}$$

3. We will choose *u* and *k* such that, $u \ = \ k \ = \ O(\sqrt{N})$ i.e. if we choose $u \ = \ k \ = \ \lceil \frac{\sqrt{N}}{C} \rceil$, where *C* $(4 \leq C \leq 8)$ is a small integer. Since $u.v = N$, therefore, $v \approx C.\sqrt{N}$ and $t \approx C^2$.

By choosing the parameters as described above, we have

(a) The security of the bi-variate polynomial $\{f(c_1,\, x_2,\, x_3)\}^{(k+1)}$ is ensured by,

$$(k+1).t \ \geq \ v$$

(b) Maximum storage requirement is bounded above as follows,

$$\begin{aligned} (k+u).t \ &= \ (k+u).\lceil \frac{v}{k} \rceil \ \leq \ (k+u).(\frac{v}{k}+1) \\ &= \ v + \frac{u.v}{k} + (k+u) \\ &< \ (v + \frac{N}{k}) + (2.\sqrt{N}) \quad \text{as,} \ \ u = \ k \ < \ \sqrt{N} \end{aligned}$$

Since, $(v + \frac{N}{k}) \ < \ N$, as if

$$\begin{aligned} (v + \frac{N}{k}) \ &\geq \ N \implies v \geq \ N - \frac{N}{k} \\ \implies v \ &\geq \ N(1 - \frac{1}{k}) \ = \ u.v(1 - \frac{1}{k}) \\ \implies 1 \ &\geq \ u.(1 - \frac{1}{k}) \ since, \ u, \, k \ > \ 2 \\ \implies 1 \ &\geq \ \frac{u}{2} > 1 \end{aligned}$$

Which is not possible, therefore $(v + \frac{N}{k}) \; < \; N$, Hence,

$$(k + u).t \quad < \quad N + (2.\sqrt{N}) \tag{3.20}$$

This equation (3.20) shows that maximum storage requirement for a sensor node is of order $O(N)$.

3.4 Performance Evaluation

3.4.1 Memory Cost

Each sensor node in the i^{th} row of the grid is storing the partial information of the bi-variate symmetric polynomial $\{f(c_i,\, x_2,\, x_3)\}^{(k+i)}$, its node ID and the integer *u* to compute the function Φ. In other words, each node $\langle i,\, j\rangle$ stores the uni-variate symmetric polynomial $\{f(c_i,\, c_j,\, x_3)\}^{(k+i)}$, where degree of this polynomial is $(k + i).t$. So each node is storing $((k + i).t + 1)$ coefficient of the polynomial.

The coefficients of the share polynomials are from the finite field F_q. So, each coefficient needs $\log_2 q$ bits of storage. And each node is storing $((k + i).t + 1)$ coefficients of the polynomial share, So total storage requirement for polynomial share is $((k + i).t + 1).\log_2 q$ bits per node.
Sensor node ID has two coordinates. First coordinate requires $log_2 u$ bits at max, and second coordinate requires $log_2 v$ bits at max. So total number of bits required for node ID is

$$log_2 u + log_2 v = log_2 u.v = log_2 N \;\; bits\; per\; node,$$

Therefore, total storage requirement for a sensor node is

$$((k + i).t + 1).\log_2 q + log_2 N + log_2 u \;\; bits,$$

Sensor nodes are capable of determining the ID's of those sensor nodes to which they can establish a direct communication key. Therefore, we do not need to store the ID's of these sensor nodes. Also it is possible to determine the IDs of intermediate nodes for indirect key establishment.

3.4.2 Computation Overhead for Sensors

Our scheme is based on symmetric key cryptography. Here a *t*-degree tri-variate symmetric polynomial is used to distribute a polynomial share. Each sensor nodes associated with i^{th} row of the grid can calculate the communication key using ((k+i).t)-degree uni-variate polynomial, which is derived from the share of the global polynomial. To calculate the communication key each sensor node needs to compute (2.(k+i).t - 1) modular multiplication over F_q. Where ((k+i).t - 1) for $x^2,\, x_3,\, \cdots,\, x_{((k+i).t)}$ and ((k+i).t) for $b_1.x,\, b_2.x^2,\, b_3.x^3,\, \cdots,\, b_{((k+i).t)}.x^{((k+i).t)}$ and then again on an average $\frac{3.k}{2}$ modular multiplication.

Chapter 4

Extension to 3-Dimensional Grid-Based Key Agreement Protocol

Let we consider a 4-variate symmetric t-degree polynomial given as

$$f(x_1,\, x_2,\, x_3,\, x_4) = \sum_{i_1=0}^{t}\sum_{i_2=0}^{t}\sum_{i_3=0}^{t}\sum_{i_4=0}^{t} a_{i_1,i_2,i_3,i_4}\, x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4} \tag{4.1}$$

Where all the polynomial coefficients are chosen from a finite field F_q, and q is either a prime number or prime power, large enough to accommodate a cryptographic key.
Here in this chapter unless otherwise stated, all the calculation are done over the finite field F_q.
Consider that the 4-tuple permutation is a bijection and can be defined as,

$$\sigma\; : \{1,\, 2,\, 3,\, 4\} \longmapsto \{1,\, 2,\, 3,\, 4\} \tag{4.2}$$

Now if we choose all the coefficients of the polynomial such that

$$a_{i_1,i_2,i_3,i_4} \;=\; a_{i_{\sigma(1)},i_{\sigma(2)},i_{\sigma(3)},i_{\sigma(4)}} \tag{4.3}$$

for any permutation σ of $\{1, 2, 3, 4\}$, then we will obtain the symmetric polynomial i.e.

$$f(x_1,\, x_2,\, x_3,\, x_4) \;=\; f(x_{\sigma(1)},\, x_{\sigma(2)},\, x_{\sigma(3)},\, x_{\sigma(4)}) \tag{4.4}$$

Let we consider that the sensor network has *N* sensor nodes. So we will consider a three dimensional grid with dimension $u \times v \times v$. where *u* and *v* are integers such that $u \cdot v^2 = N$

The set-up server will assigns each sensor node in the network to a unique non-occupied $(i,\, j,\, k)$ coordinate in this grid. where *i* , *j* and *k* are such that $1 \,\leq\, i \,\leq\, u$, $1 \,\leq\, j \,\leq\, v$ and $1 \leq k \leq v$. The ID of the sensor node associated with the coordinate $(i,\, j,\, k)$ is represented by $\langle i,\, j,\, k\rangle$.

Let we consider a set of credentials (positive integers) *C*(usually, $C = \{1, 2, \cdots, 2v\}$. In upcoming section *Polynomial Share Pre-distribution* we will present a method to generate *C*) such that *C* $= \{c_1, c_2, \ldots c_{2.v}\}$ and $|C| = 2.v$. Now we form three sets, first set S_1 from first *u* elements of *C*, the second set S_2 from first *v* elements of *C* and the last set S_3 from last *v* elements of set *C* i.e.

$S_1 = \{c_1, c_2, \ldots c_u\}$, $S_2 = \{c_1, c_2, \ldots c_v\}$ and $S_3 = \{c_{(v+1)}, c_{(v+2)}, \ldots c_{2.v}\}$.

Consider that x_1 takes values from the set S_1 , x_2 takes values from the set S_2 and x_3 takes values from the set S_3 in the 4-variate symmetric polynomial $f(x_1, x_2, x_3, x_4)$.

Now we will compute a uni-variate polynomial share for each cell in the grid from the 4-variate symmetric t-degree polynomial i.e. each cell (i, j, k) in the grid is associated with a $(\kappa + i).t$ - degree uni-variate polynomial share $\{f(c_i, c_j, c_k, x_4)\}^{(\kappa+i)}$. Where κ is a suitably choosen positive integer.

Each sensor node $\langle i, j, k\rangle$ in the sensor network has three credential, which are positive integers choosen from set *C* and denoted by (c_i, c_j, c_k) where $(c_i, c_j, c_k) \in S_1 \times S_2 \times S_3$. Before node deployment in the sensor network, a polynomial share $\{f(c_i, c_j, c_k, x_4)\}^{(\kappa+i)}$ is distributed to each sensor node $\langle i, j, k\rangle$. By distributing the polynomial share to sensor node, we mean that for each sensor node $\langle i, j, k\rangle$ we store the coefficient of $(\kappa + i).t$- degree uni-variate polynomial $\{f(c_i, c_j, c_k, x_4)\}^{(\kappa+i)}$ into node memory.

4.1 3-Dimensional Grid-Based Key Establishment Protocol For WSNs

Now we will describe our key exchange scheme. This scheme uses tri-variate symmetric polynomial which ensures extra connectivity in the network. Our schema works in Two Phases:

1. Polynomial Share Pre-distribution.
2. Key Establishment Mechanism
   (a) Direct Key Establishment.
   (b) Indirect Key Establishment.

4.1.1 Polynomial Share Pre-distribution

Polynomial Share Pre-distribution phase is performed prior to network deployment by a trusted set-up server. The set-up server generates a global 4-variate symmetric t-degree polynomial and a set *C* of credentials as described above . The set-up server will use this global polynomial and set of credentials to calculate the polynomial share for each sensor node.

Since each sensor node $\langle i,\, j,\, k\rangle$ has three credentials $(c_i,\, c_j, c_k)$. Which are positive integers such that

$(c_i,\, c_j, c_k) \in S_1 \times S_2 \times S_3$, Where $S_1 = \{c_1,\, c_2,\, ...\, c_u\}$, $S_2 = \{c_1,\, c_2,\, ...\, c_v\}$ and $S_3 = \{c_{(v+1)},\, c_{(v+2)},\, \cdots c_{2.v}\}$.

The elements of set *C* can be preloaded into sensor nodes before deployments but it causes extra memory overhead. Therefore we give a method to derive the credentials from node ID's by using a bijection Φ between node ID's and credentials. So that credentials can be derived from the node ID's. The function Φ is defined as,

$$\begin{aligned}\Phi : \{1, 2, ..., u\} \times \{1, 2, ..., v\} \times \{1, 2, ..., v\} &\longmapsto S_1 \times S_2 \times S_3 \\ s.t. \qquad (c_i,\, c_j,\, c_k) &= \Phi\{(i,\, j,\, k)\} \\ &\overset{\text{def}}{=} (\mu + i,\, \mu + j,\, \mu + v + k) \qquad (4.5)\end{aligned}$$

Since each node can compute the value of credentials used for computing the polynomial share, so there is no need to store the credentials $(c_i,\, c_j, c_k)$ in the sensor nodes, but each node needs to store μ in order to compute Φ.

Note : In the definition of function Φ, If we consider $\mu = 0$ then $(c_i,\, c_j,\, c_k) = (i,\, j,\, v + k)$ and then we do not need to store μ in each sensor node.

Now Each sensor node $\langle i,\, j,\, k\rangle$ in the sensor network has a three credentials $(c_i,\, c_j,\, c_k) = (\mu + i,\, \mu + j,\, \mu + v + k)$ and the polynomial share assigned to it by set-up server is computed as follows :

$$\begin{aligned}\{f(c_i,\, c_j,\, c_k,\, x_4)\}^{(\kappa+i)} &= \{f(\mu + i,\, \mu + j,\, \mu + v + k,\, x_4)\}^{(\kappa+i)} \\ &= \{\sum_{i_1=0}^{t}\sum_{i_2=0}^{t}\sum_{i_3=0}^{t}\sum_{i_4=0}^{t} a_{i_1,i_2,i_3,i_4}\, c_i^{i_1} c_j^{i_2} c_k^{i_3}\, x_4^{i_4}\}^{(\kappa+i)} \qquad (4.6)\end{aligned}$$

Therefore, Every node $\langle i,\, j,\, k\rangle$ in the sensor network is storing a $(\kappa + i).t$-degree univariate polynomial having $((\kappa + i).t + 1)$ coefficients over the finite field F_q. Before nodes deployment these coefficients are preloaded in the sensor nodes and are used for computing communication key during key establishment process.

4.1.2 Key Establishment

After node deployment in the sensor network two nodes can establish a communication key using there polynomial share. In our model there is two ways to establish a communication key.

1. Direct Key Establishment.
2. Indirect Key Establishment.

Direct Key Establishment :-

Let we consider that sensor node $\langle i_1,\, j_1,\, k_1\rangle$ wants to establish a communication key with the sensor node $\langle i_2,\, j_2\, k_2\rangle$. These two nodes can establish a direct communication key if they have a common credential. The credential associated with sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2\, k_2\rangle$ are $(c_{i_1},\, c_{j_1},\, c_{k_1})$ and $(c_{i_2},\, c_{j_2},\, c_{k_2})$ respectively.

We divide direct key communication process in two cases.

- **Case 1 -** Both sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ are in plane $x_3 = k$ of the grid i.e. $k_1 = k_2 = k(say)$, so $c_{k_1} = c_{k_2} = c_k \ (say)$.
  In this case, sensor nodes can use 2-dimensional grid based key establishment method for establishing communication key between them.

- **Case 2 -** Both sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ are from different x_3 plane of the grid i.e. $k_1 \neq k_2$, so $c_{k_1} \neq c_{k_2}$. We further divide this case in two sub-cases.

1. **Case 2.1 -** Both sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ are from different x_3 plane of the grid i.e. $k_1 \neq k_2$, but $i_1 = i_2 = i \ (say)$ and $j_1 = j_2 = j \ (say)$, therefore, $c_{k_1} \neq c_{k_2}$ but $c_{i_1} = c_{i_2} = c_i\,(say)$ and $c_{j_1} = c_{j_2} = c_j\,(say)$.

   Sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ has polynomial share $\{f(c_i,\, c_j,\, c_{k_1},\, x_4)\}^{(\kappa+i)}$ and $\{f(c_i,\, c_j,\, c_{k_2},\, x_4)\}^{(\kappa+i)}$ respectively.

   Now sensor node $\langle i_1,\, j_1,\, k_1\rangle$ calculates the credential k_{i_2} and communication key K_1 as follows,

$$\begin{aligned} (c_{i_2},\, c_{j_2},\, c_{k_2}\,) &= \Phi(i_2,\, j_2,\, k_2) \\ &= (\mu + i_2,\, \mu + j_2,\, \mu + v + k_2) \\ \Longrightarrow c_{k_2} &= (\mu + v + k_2) \end{aligned}$$

$$K_1 = \{f(c_i,\, c_j,\, c_{k_1},\, c_{k_2})\}^{(\kappa+i)} \tag{4.7}$$

   Similarly, sensor node $\langle i_2,\, j_2,\, k_2\rangle$ calculates the credential c_{k_1} and communication key K_2 as follows,

$$\begin{aligned} (c_{i_1},\, c_{j_1},\, c_{k_1}\,) &= \Phi(i_1,\, j_1,\, k_1) \\ &= (\mu + i_1,\, \mu + j_1,\, \mu + v + k_1) \\ \Longrightarrow c_{k_1} &= (\mu + v + k_1) \end{aligned}$$

$$K_2 = \{f(c_i,\, c_j,\, c_{k_2},\, c_{k_1})\}^{(\kappa+i)} \tag{4.8}$$

Since $f(x_1,\, x_2,\, x_3,\, x_4)$ is a symmetric 4-variate polynomial, therefore from equation (4.7) & (4.8), we have

$$\{f(c_i,\, c_j,\, c_{k_1},\, c_{k_2})\}^{(\kappa+i)} \;=\; \{f(c_i,\, c_j,\, c_{k_2},\, c_{k_1})\}^{(\kappa+i)}$$

$$\Longrightarrow\; K_1 \;=\; K_2 \;=\; K\,, (say)$$

Using this key *K* sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ can communicate with each other securely.

2. **Case 2.2 -** Sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ are from different x_3 plain of the grid i.e., $k_1 \;\neq\; k_2$, but $i_1 \;=\; j_2$, and $j_1 \;=\; i_2$, so, $c_{k_1} \;\neq\; c_{k_2}$ but $c_{i_1} \;=\; c_{j_2} \;=\; c_1\,(say)$ and $c_{j_1} \;=\; c_{i_2} \;=\; c_2\,(say)$.

   Sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ has polynomial share $\{f(c_1,\, c_2,\, c_{k_1},\, x_4)\}^{(\kappa+i_1)}$ and $\{f(c_2,\, c_1,\, c_{k_2},\, x_4)\}^{(\kappa+i_2)}$ respectively.

   Now sensor node $\langle i_1,\, j_1,\, k_1\rangle$ calculates the credential c_{k_2} and communication key K_1 as follows,

$$\begin{aligned}
(c_{i_2},\, c_{j_2},\, c_{k_2}\,) &= \Phi(i_2,\, j_2,\, k_2) \\
&= (\mu + i_2,\, \mu + j_2,\, \mu + v + k_2) \\
\Longrightarrow c_{k_2} &= (\mu + v + k_2)
\end{aligned}$$

$$\begin{aligned}
K_1' &= \{f(c_1,\, c_2,\, c_{k_1},\, c_{k_2})\}^{(\kappa+i_1)} \\
K_1 &= \{K_1'\}^{(k+i_2)} \\
&= \{\{f(c_1,\, c_2,\, c_{k_1},\, c_{k_2})\}^{(\kappa+i_1)}\}^{(k+i_2)} \\
&= \{f(c_1,\, c_2,\, c_{k_1},\, c_{k_2})\}^{(k+i_1).(k+i_2)} \qquad (4.9)
\end{aligned}$$

   Similarly, sensor node $\langle i_2,\, j_2,\, k_2\rangle$ calculates the credential c_{k_1} and communication key K_2 as follows,

$$\begin{aligned}
(c_{i_1},\, c_{j_1},\, c_{k_1}\,) &= \Phi(i_1,\, j_1,\, k_1) \\
&= (\mu + i_1,\, \mu + j_1,\, \mu + v + k_1) \\
\Longrightarrow c_{k_1} &= (\mu + v + k_1)
\end{aligned}$$

$$\begin{aligned}
K_2' &= \{f(c_2,\, c_1,\, c_{k_2},\, c_{k_1})\}^{(\kappa+i_2)} \\
K_2 &= \{K_2'\}^{(k+i_1)} \\
&= \{\{f(c_2,\, c_1,\, c_{k_2},\, c_{k_1})\}^{(\kappa+i_2)}\}^{(k+i_1)} \\
&= \{f(c_2,\, c_1,\, c_{k_2},\, c_{k_1})\}^{(k+i_1).(k+i_2)} \qquad (4.10)
\end{aligned}$$

Since $f(x_1,\, x_2,\, x_3,\, x_4)$ is a symmetric 4-variate polynomial,, therefore, from equation (4.9) & (4.10), we have

$$\{f(c_1,\, c_2,\, c_{k_1},\, c_{k_2})\}^{(k+i_1).(k+i_2)} \;=\; \{f(c_2,\, c_1,\, c_{k_2},\, c_{k_1})\}^{(k+i_1).(k+i_2)}$$

$$\implies K_1 \;=\; K_2 \;=\; K\,, (say)$$

Using this key *K* sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ can communicate with each other securely.

Indirect Key Establishment :-

Let we consider that sensor node $\langle i_1,\, j_1,\, k_1\rangle$ wants to establish a communication key with the sensor node $\langle i_2,\, j_2,\, k_2\rangle$ where node ID's are such that sensor node $\langle i_1,\, j_1,\, k_1\rangle$ cannot establish a direct key with sensor node $\langle i_2,\, j_2,\, k_2\rangle$. So sensor node $\langle i_1,\, j_1,\, k_1\rangle$ search for some other sensor node $\langle i',\, j',\, k'\rangle$ in the sensor network such that sensor node $\langle i_1,\, j_1,\, k_1\rangle$ and sensor node $\langle i_2,\, j_2,\, k_2\rangle$ can directly communicate with the sensor node $\langle i',\, j',\, k'\rangle$. And hence, using sensor node $\langle i',\, j',\, k'\rangle$ as an intermediate node the two sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ can establish a communication key.

It is easy to show that if there is no compromised node in the network then a pair of sensor nodes requires at max two intermediate nodes to establish communication key. And if there is no compromised node in the network then there will always exist sensor nodes that can be used as intermediate nodes for indirect communication. And if there are compromised nodes in the network then by using more than two non-compromised intermediate nodes, it is always possible to establish a communication key.

Sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ requires only one intermediate sensor node if

1. Either both the sensor nodes are in same plane i.e. $i_1 = i_2$, or $j_1 = j_2$, or $k_1 = k_2$.
2. Or there is an intermediate node $\langle i',\, j',\, k'\rangle$ such that,
   (a) Either $\{i_1 = j',\, j_1 = i',\, k' = k_2\}$ and either $j' = i_2$ or $i' = j_2$,
   (b) Or $\{i_2 = j',\, j_2 = i',\, k' = k_1\}$ and either $i_1 = j'$ or $j_2 = i'$,

If sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ cannot establish a communication key using only one intermediate then they use two or more intermediate nodes for key establishment.

Now we enumerate number of distinct paths between sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ that use two intermediate nodes.

1. Let us consider sensor node $\langle i,\, j,\, k\rangle$, such that sensor node $\langle i_1,\, j_1,\, k_1\rangle$ can compute direct communication key and share two common credential values with this node. Also sensor node $\langle i,\, j,\, k\rangle$, and $\langle i_2,\, j_2,\, k_2\rangle$ are in same plane. There are four distinct ways to choose such sensor nodes, namely

$$\langle i_1,\, j_1,\, k_2\rangle,\ \langle j_1,\, i_1,\, k_2\rangle,\ \langle i_1,\, j_2,\, k_1\rangle,\ and\ \langle i_2,\, j_1,\, k_1\rangle$$

2. Now we determine number of possible intermediate nodes between node $\langle i_2,\, j_2,\, k_2\rangle$ and four possible choices of node $\langle i,\, j,\, k\rangle$,.

   (a) Sensor node $\langle i_2,\, j_2,\, k_2\rangle$ and first two choices of node $\langle i,\, j,\, k\rangle$, i.e. $\langle i_1,\, j_1,\, k_2\rangle,\ and\ \langle j_1,\, i_1,\, k_2\rangle$ are in $x_3 = k_2$ plane. So using two dimensional scheme, sensor node $\langle i_2,\, j_2,\, k_2\rangle$ can establish a communication key with $\langle i_1,\, j_1,\, k_2\rangle,\ and\ \langle j_1,\, i_1,\, k_2\rangle$ using only one intermediate sensor node. And there are eight such possible sensor nodes that can be used as an intermediate node.

   (b) Sensor node $\langle i_2,\, j_2,\, k_2\rangle$ and $\langle i_1,\, j_2,\, k_1\rangle$ can communicate with each other using one intermediate node. And there are three such possible intermediate nodes, namely

   $$\langle i_2,\, j_2,\, k_1\rangle,\ \langle i_1,\, j_2,\, k_2\rangle,\ \langle j_2,\, i_1,\, k_2\rangle,$$

   (c) Similar to previous case, sensor node $\langle i_2,\, j_2,\, k_2\rangle$ and $\langle i_2,\, j_1,\, k_1\rangle$ can communicate with each other using one intermediate node. And there are three such possible intermediate nodes.

So there are fourteen distinct possible paths by which sensor nodes $\langle i_1,\, j_1,\, k_1\rangle$ and $\langle i_2,\, j_2,\, k_2\rangle$ can establish communication key using two intermediate nodes. And there are four disjoint paths, which do not share any sensor node with each others. If an adversary compromised nodes from these four disjoint paths then sensor nodes can use more than two intermediate nodes to establish the communication key.

4.2 Security Analysis and Performance Evaluation

In this section we will analysis the security performance of our model and compute memory cost for each sensor node, node resilience, compromise attack, in the network and computation energy cost.

4.2.1 Node Compromise Attack

Because of hostile nature of deployment area, Sensor nodes may have to face wide variety of malicious attack. Also If an adversary gets physical access to sensor nodes then he/she may tamper the node and may get the share information of that node. Using this share information he/she may try to discover the secret communication key of other pair of sensor nodes. Here we will analysis the effect of node compromise attack at plane level and network level, as analysis of row and column level node compromise attack is similar to two dimensional case.

Node Compromise Attack in a Plane

Let we consider plane level node compromise attack in the network. A pair of sensor nodes in plane $x_1 = i$ computes communication key from a *(k + i).t*-degree tri-variate polynomial.

$$\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}$$

Where c_i is the common credential between the pair of nodes.
Now to compromise the pairwise key without compromising all the nodes from $x_1 = i,\ \{i = 1,\, 2,\, 3, \cdots,\, u\}$ plane adversary needs to compromise the shared polynomial $\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}$ of degree $(\kappa + i).t$. It has been shown in [2] that a *t*-degree bi-variate polynomial is *t*-secure i.e. coalition between less than *(t + 1)* node holding shares of *t*-degree bi-variate polynomial cannot expose the polynomial. so adversary needs to compromise at-least *((k + i).t + 1)* nodes in each row *j* of the plane $x_1 = i,$

And each row and column in plane $x_1 = i,$ of the grid holds *v* nodes. So to ensure that pairwise key between any pair of nodes from any row in the plane $x_1 = i,$ is unsolvable by other *(v - 2)* nodes from same row, degree of the polynomial must satisfy,

$$0 \le (v - 2) \le (k + 1).t \tag{4.11}$$

Now Suppose that in a plane level attack an adversary compromises all v^2 sensor nodes of i^{th} plane. then adversary can construct $2.v.\dfrac{v.(v+1)}{2}$ equations. where total number of nodes in the $x_1 = i,$ plane is v^2. In order to expose the tri-variate symmetric polynomial $\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}$ of degree $(\kappa + i).t,$ adversary needs to determine $\begin{pmatrix}(\kappa+i).t+3\\3\end{pmatrix}$ coefficients of the polynomial $\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}$. So we choose *k* and *t* such that

$$\begin{aligned} v^2.(v+1) &\le \begin{pmatrix}(\kappa+i).t+3\\3\end{pmatrix} \\ \forall i &= 1,\, 2,\, 3, \cdots,\, u, \end{aligned} \tag{4.12}$$

Even if adversary gets the polynomial $\{f(c_1,\, x_2,\, x_3,\, x_4)\}^{(\kappa+1)}$ for first plane $x_1 = 1,$ then also to get the base polynomial share $f(c_1,\, x_2,\, x_3,\, x_4)$, by the adversary . He/She must have to find $(\kappa + 1)^{th}$ root of the polynomial $\{f(c_1,\, x_2,\, x_3,\, x_4)\}^{(\kappa+1)}$, which is a hard problem in the finite field F_q, and $v^2.(v+1) < \begin{pmatrix}(\kappa+i).t+3\\3\end{pmatrix} \forall i > 1$, so adversary cannot get the polynomial $\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}\ \forall i > 1.$

Network Level Node Compromise Attack

Now we will consider network level node compromise attack. We consider that an adversary may compromise all the nodes of the sensor network. And total number of sensor nodes in the network is

$$N = u.v^2 \tag{4.13}$$

so, total number of equation T_e that an adversary can construct is,

$$\begin{aligned} T_e &= u.\frac{2.v^2.(v+1)}{2} + 2.v.\frac{u.v(3v-u+2)}{2} \\ &= u.v^2.(v+1+3.v-u+2) \\ &= u.v^2(4v-u+3) \qquad (4.14) \end{aligned}$$

Using these equation adversary needs to construct *u* - tri-variate symmetric polynomial share namely,

$$\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(k+i)}, \;\; \forall i \,=\, 1,\, 2 \cdots,\, u$$

The number of distinct coefficient of a *t*-degree tri-variate symmetric polynomial is $\binom{t+3}{3}$.

Therefore, to find these *u* -tri-variate symmetric polynomial share, adversary needs to find the coefficient of these polynomials. So on an average adversary has to find $u.\binom{(k+\lceil u/2\rceil).t+3}{3}$ distinct coefficients.So we choose the values of *k, t, u* and *v* such that,

$$\begin{aligned} u.v^2(4v-u+3) &\leq u.\binom{(k+\lceil u/2\rceil).t+3}{3} \qquad (4.15) \\ \forall i &= 1,\, 2,\, 3, \cdots,\, u \end{aligned}$$

4.2.2 Choice of parameters κ, *t*, *u*, and *v*

In this protocol we have considered the parameters κ, *t, u,* and *v*. where *u and v* are such that $u \cdot v^2 = N$. We will do analysis for possible values of these parameters.

Security requirement for the protocol and storage limitation imposes following restriction.

1. Sensor nodes associated to each row of the first plane $x_1 = 1$ of the grid are storing polynomial share from a bi-variate polynomial of degree $(\kappa+1).t$. So from equation (4.11) we have $0 \leq (v-2) \leq (\kappa+1).t$
2. Sensor nodes corresponding to each row of the last plane $x_1 = u$, of the grid are storing polynomial share from a bi-variate polynomial of degree $(\kappa+u).t$. So we need to choose *u*, κ and *t* such that $(\kappa+u).t$ should not be very large.

Using these restriction we will choose the values of *u, v,* κ and *t* such that they fulfil the security and storage requirements.

1. Secrecy of the bi-variate polynomial $\{f(c_1,\, c_j,\, x_3,\, x_4)\}^{(\kappa+1)}$ requires that, $0 \leq (v-2) \leq (\kappa+1).t$, so, we will choose κ and *t* such that

$$(\kappa+1).t \,\geq\, v \;\implies\; t \,\geq\, \lceil\frac{v}{(\kappa+1)}\rceil \,\approx\, \lceil\frac{v}{\kappa}\rceil \qquad (4.16)$$

2. if we choose $u.v^2 = \alpha.\sqrt{\frac{N}{\alpha}}.\sqrt{\frac{N}{\alpha}}$, where α $(4 \leq \sqrt{\alpha} \leq 8)$ is a small integer. Then $u = \alpha,\ v = \lceil\sqrt{\frac{N}{\alpha}}\rceil$, also consider $\kappa = \lceil\sqrt{\frac{N}{\alpha^3}}\rceil$, so that

$$\begin{aligned} t &= \lceil\frac{v}{\kappa}\rceil \\ \Rightarrow t &= \lceil\frac{\sqrt{\frac{N}{\alpha}}}{\sqrt{\frac{N}{\alpha^3}}}\rceil \\ &= \alpha \end{aligned} \tag{4.17}$$

By choosing the parameters as described above, we have

(a) Security of the bi-variate polynomial $\{f(c_1,\, c_j,\, x_3,\, x_4)\}^{(\kappa+1)}$ is ensured by,

$$(k+1).t \,\geq\, v$$

(b) Maximum storage requirement is bounded above as,

$$\begin{aligned} (\kappa+u).t &= (\kappa+\alpha).\alpha \\ &= (\lceil\sqrt{\frac{N}{\alpha^3}}\rceil+\alpha).\alpha \,\leq\, (\sqrt{\frac{N}{\alpha^3}}+1+\alpha).\alpha \\ &= (\sqrt{\frac{N}{\alpha}}+\alpha+\alpha^2) \qquad (4.18) \\ &= (v+\alpha+\alpha^2), \end{aligned}$$

where α is a small positive integer. This equation (4.18) shows that maximum storage requirement for a sensor node is of order $O(\sqrt{N})$.

4.2.3 Memory Cost

Each sensor node in the i^{th} plane of the grid is storing the partial information of the tri-variate symmetric polynomial $\{f(c_i,\, x_2,\, x_3,\, x_4)\}^{(\kappa+i)}$. In other words, each node $\langle i,\, j,\, k\rangle$ stores the uni-variate symmetric polynomial $\{f(c_i,\, c_j,\, c_k,\, x_4)\}^{(\kappa+i)}$. And degree of this polynomial is $(k+i).t$, so each node is storing $((k+i).t+1)$ coefficient of the polynomial.

Since the coefficients of the share polynomials are from the finite field F_q. Therefore, each coefficient needs $\log_2 q$ bits of storage. And each node is storing $((k+i).t+1)$ coefficients of the polynomial share, So storage requirement for polynomial share is $((k+i).t+1).\log_2 q$ per node.

Since sensor nodes are capable of determining the ID's of those sensor nodes to which they can establish a direct communication key. Therefore, we do not need to store the ID's of these sensor nodes. Also it is possible to determine the IDs of intermediate nodes for indirect key establishment.

4.2.4 Computation Overhead for Sensors

Our scheme is based on symmetric key cryptography. Here a *t*-degree tri-variate symmetric polynomial is used to distribute a polynomial share. Each sensor nodes associated with i^{th} plane of the grid can calculate the communication key using ((k+i).t)-degree uni-variate polynomial, which is derived from the share of the global polynomial. To calculate the communication key each sensor node needs to compute (2.(k+i).t - 1) modular multiplication over F_q. Where ((k+i).t - 1) for x^2, x_3, $\cdots$, $x_{((k+i).t)}$ and ((k+i).t) for $b_1.x$, $b_2.x^2$, $b_3.x^3$, $\cdots$, $b_{((k+i).t)}.x^{((k+i).t)}$ and then again on an average $(k+\alpha).\alpha$ modular exponentiation.

References

[1] R. Blom, An optimal class of symmetric key generation systems, EUROCRYPT 84, 1985.

[2] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectaly - secure key distribution for dynamic conferences. In Advances in Cryptology - CRYPTO '92,

[3] C. Boyd and A. Mathuria, Protocols for Authentication and Key Establishment. Springer-Verlag, 2003.

[4] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In IEEE Symposium on Research in Security and Privacy, 2003

[5] J. Deng, R. Han and S. Mishra, Enhancing Base Station Security in Wireless Sensor Networks, Technical Report CU-CS-951-03, Department of Computer Science, University of Colorado, 2003.

[6] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In Proc. of the 9th ACM Conf. on Computer and Communications Security, pages 4147, November 2002.

[7] Yun Zhou, Yuguang Fang, Scalable and Deterministic Key Agreement for Large Scale Networks, IEEE- 2007.

[8] Y. Zhou, Y. Fang, A Two -Layer Key Establishment Scheme for Wireless Sensor Network, IEEE-2007.

[9] M. Chen, W. Cui, V. Wen and A. Woo, Security and Deployment Issues in a Sensor Network, Ninja Project, A Scalable Internet Services Architecture, Berkeley, http://citeseer.nj.nec.com/chen00s 2000.

[10] D. Liu and P. Ning. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In Proc. of the 10th Annual Network and Distributed System Security Symposium, pages 263276, February 2003

[11] D. Liu, P. Ning, K. Sun, Efficient self-healing group key distribution with revocation capability, Proceedings of the 10th ACM conference on Computer and communication security, 2003.

[12] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, Proceedings of the 10th ACM conference on Computer and communication security, 2003.

[13] W. Diffie, P. van Oorschot, and M. Wiener, Authentication and authenticated key exchanges, Designs, Codes, and Cryptography, vol. 2, no. 2, pp. 107125, June 1992.

[14] W. Du, J. Deng, Y. S. Han, P. Varshney, A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks, In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), 2003.

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

[16] Sushmita Ruj, and Bimal Roy, Key Predistribution Using Partially Balanced Designs in Wireless Sensor Networks, LNCS 4742, pp. 431445, 2007

[17] H.C. Lin, and Y.M.Tesng, ”A Scalable ID-Based Key Establishment Protocol for Wireless Sensor Networks” 2007