UNIVERSITY OF CINCINNATI

Date: 26-Jul-2010

I, Amit Gaur, hereby submit this original work as part of the requirements for the degree of:
Master of Science
in Computer Science

It is entitled:
Secured Communication in Wireless Sensor Network (WSN) and Authentic Associations in Wireless Mesh Networks.

Student Signature: Amit Gaur

This work and its defense approved by:

Committee Chair: Dharma Agrawal, DSc
Dharma Agrawal, DSc

Raj Bhatnagar, PhD
Raj Bhatnagar, PhD

Carla Purdy, C, PhD
Carla Purdy, C, PhD
Secured Communication in Wireless Sensor Network (WSN) and Authentic Associations in Wireless Mesh Networks

A thesis submitted to the
Graduate School
Of the University of Cincinnati
In partial fulfillment of the
Requirements for the degree of
Master of Science
In the Department of Computer Science
In the College of Engineering and Applied Science

By
Amit Gaur

B.S. Bharati Vidyapeeth
2006

Committee Chair: Dr. Dharma Agrawal
Abstract

Wireless sensors are low power devices with small transmission range, restricted computation power, limited amount of memory and with portable power supply. Wireless Sensor Network (WSN) is a collection of such sensors where the number of sensors can vary from few hundreds to thousands. Performing secure pair-wise communication between sensors is a really difficult task due to inherent characteristics such as lack of any fixed infrastructure. As memory and power consumption are most stringent requirements for these devices, use of conventional techniques for secured communication are totally out of question. This thesis introduces scheme that enables a complete pair-wise secure connectivity between any two adjacent sensor nodes in spite of using small key ring (KR) for sensors. The Proposed Scheme (ELKPD) doesn't require any additional hardware while providing keys to the sensors irrespective of their location. Also, proposed scheme is easily scalable which allows enables addition of sensor nodes without any computational or hardware overheads.

Due to the varying degree of mobility of Mesh Clients has provided much more flexibility in Wireless Mesh Networks. And establishing an Authentic Association among entities is a non-trivial problem. In this thesis, we introduce a Polynomial Based scheme which not only provides high pair-wise connectivity, low communication and storage overhead and high scalability but also makes on the fly Authentic Association feasible. The proposed scheme is also observed to be resilient against both the traffic analysis and the node capture attacks.

Some of the problems for future work are also identified in this thesis.
Acknowledgements

I am highly obliged towards my academic advisor Dr. Dharma P Agrawal for his invaluable guidance and untiring efforts during my course of research. I am grateful that he always encouraged us to work in areas that personally interested us and found time to discuss and review our work. I would also like to thank Dr. Carla Purdy and Dr. Bhatnagar for being a part of my defense committee.

I am very much thankful to all my fellow researchers at the Center for Distributed and Mobile Computing (CDMC) for critiquing my work and providing honest views that helped me to enhance my research. I would like to specially mention Abhinav Prakash and Smita Toshnival for their invaluable help.

I would also like to thank my parents for supporting me in all my decisions.

On the end note, I would like to thank University of Cincinnati, College of Engineering for granting me University Graduate Scholarship and providing me a platform to fulfill my dreams of performing meaningful research.
# Table of Contents

List of Figures and Tables ........................................................................................................ 7

Chapter 1 .................................................................................................................................. 9

1 Introduction ................................................................................................................................. 9

1.1 Motivation for ELKPD .............................................................................................................. 9
1.2 Motivation for Polynomial based Scheme (PBS) .................................................................. 11
1.3 Organization of Thesis ............................................................................................................. 13

Chapter 2 .................................................................................................................................. 14

2 Related Work on Enhanced Location based Key Pre-Distribution Scheme (ELKPD) ............ 14

Chapter 3 .................................................................................................................................. 18

3 ELKPD Scheme .......................................................................................................................... 18

3.1 Key Generation and Pre-deployment Phase: ........................................................................ 19
3.2 Key Initialization Phase: ......................................................................................................... 22
3.3 Communication Phase: ............................................................................................................ 24

Chapter 4 .................................................................................................................................. 27

4 Performance Results and Security analysis ............................................................................. 27

Chapter 5 .................................................................................................................................. 33

5 Use of Polynomial based scheme (PBS) in Wireless mesh Network [17] ................................. 33

5.1 Proposed Scheme .................................................................................................................... 35
5.1.a Bivariate polynomial distribution ...................................................................................... 37
5.1.d Network Connectivity .......................................................................................................... 43
5.1.e Scalability of Network ......................................................................................................... 45
5.1.f Resilience against Node Capture Attack and Traffic analysis attack ................................. 46
List of Figures and Tables

Figures:

Figure 3.1(a). Illustration of key pool(KP) ................................................................. 20
Figure 3.1(b). Illustrating $i^{th}$ Key Matrix ................................................................. 20
Figure 3.2. Illustrating the common keys between Nodes A and Node B .................... 21
Figure 3.3. Illustrating Key Initialization ....................................................................... 23
Figure 4.1. Connectivity versus Number of Hop Counts to setup Pair-wise key. .......... 30
Figure 4.2. Impact of varying Power Levels on Compromise Ratio on ELDKP, with 500 sensors and 125 Special Nodes and LDK with 500 sensors and 200 Special Nodes. .......... 31
Figure 4.3. Impact of varying Maximum Transmission Radius on Compromise Ratio ... 32
Figure 5.1. A Generic Wireless Mesh Network (WMN) architecture ......................... 35
Figure 5.2. Illustrating Distribution of Bi-Variate Polynomial functions in a Three Dimensional Matrix. ................................................................. 38
Figure 5.3. Selection of polynomials from a single matrix. ........................................... 39
Figure 5.4. Illustrating Common Functions for two entities. ....................................... 40
Figure 5.5. Example AAA in a WMN ........................................................................... 43
Figure 5.6. Connectivity versus Number of Hop counts to set up pairwise key .......... 44
Tables:

**Table 4.1:** Illustrating optimum number of Special nodes required.------------------------ 28

**Table 4.2:** Illustrating the number of Beacon signals received and number of neighbors of each sensor node as number of special node and transmission range of sensor nodes is varied.-----29

**Table 5.1:** Network Size Vs Number of functions stored on single entity ----------------- 46
Chapter 1

1 Introduction

Wireless Sensor Networks (WSN) is a large collection of low power devices (i.e. sensor nodes), which are small in size and operate on small sized battery with very limited data processing power. These devices are usually deployed for remote hostile environments where their maintenance is quite difficult. Therefore they can be easily compromised by the adversaries [1] [2].

1.1 Motivation for ELKPD

To overcome such a situation, one possible solution is to encrypt the communication of sensitive data between sensor nodes. Encryption is a mechanism of converting a plain-text message into meaningless cipher-text so that even after interception of communicating data, it cannot be
interpreted. The conventional techniques for encryption are asymmetric cryptography (public-key) and symmetric cryptography (secret-key). Asymmetric cryptographic technique involves a pair of separate inter-related keys, one for encrypting the plain text into cipher text and another one for decrypting the cipher text back to plain text. An asymmetric cryptographic technique involves multiplication of very large numbers for generating encrypting keys which is not feasible on sensor nodes with very limited processing capability. So, despite the fact that they provide really good security for wired and cellular networks, asymmetric cryptographic techniques cannot be utilized to sensor networks.

Symmetric cryptographic technique involves using a single secret-key both for encryption and decryption purposes. This key is kept secret between a pair of users and is not revealed to anyone else. This technique is more suitable to wireless networks as it is computationally less extensive and less stringent on the memory requirements. Symmetric key cryptographic technique is applied to wireless and sensor networks in the form of Key Pre-Distribution approach where a small subset of keys (called as Key Ring) is chosen from a large pool of keys (called as Key Pool) and is distributed to sensor nodes before deployment. When a pair of sensor nodes find a common key between their pre-loaded keys into their memory, they can have secured symmetric communication. But, due to lack of any prior information regarding the topology of wireless network, distributing the keys securely and efficiently to sensor nodes is a difficult task. To address this problem of Key Pre-Distribution approach, several schemes have been proposed in the literature [5] [6] [7] [8]. Most of these schemes are based on either probability or random graph theories or bivariate polynomial calculations and are discussed in the next section.
To overcome limitations of already existing Key Pre-Distribution schemes, we propose a new scheme for secure pair-wise communication for a network of wireless sensor nodes in this thesis which is resilient against node capture attacks, provides security against attacks where communication between two nodes is being eavesdropped, provides complete network connectivity and is quite scalable without requiring any additional hardware or computational overhead.

### 1.2 Motivation for Polynomial based Scheme (PBS)

Wireless mesh networks (WMNs) have become increasingly important due to their deployment flexibility while covering a large distance of the order of tens of kilometers. Their usefulness has been further enhanced by introducing mobile relay stations (RS’s) that increases the coverage of the Base stations (BSs). In such a flexible scenario, the network traffic has to be propagated in a multi-hop fashion from the Mobile subscriber stations (MSS’s) to the AAA Server (*authentication, authorization and accounting*) through BS/RS for authentication. This is similar to the packets traveling in an ad-hoc network and we need to have a robust scheme that could protect the integrity of propagating message. According to specifications proposed by IEEE 802.16j [15] and the workgroup of IEEE 802.16m [16], the RSs can be static or mobile which exposes a WMN to threats like false handoffs and fake registrations. The mobility of the RS also makes the network prone to other types of attacks and the need for having a robust scheme more desirable.

A MSS can be connected directly to an AAA server or through a BS and/or through a RS. In such scenario, we need to employ a scheme that is both secure and consistent. Another
challenge is to keep the communication alive and secured, even in the presence of a handoff. Unlike a sensor network where we can hard code the keys on the sensors before actual deployment, RSs and MSs in a WMN topology have different levels of mobility and we need to have dynamic keys which can be generated on the fly, ensuring the communication to be secured. We propose to use a polynomial based distributed authentication scheme so as to achieve secure communication by generating keys in real-time.

We propose to use bivariate polynomials that enable us to compute a common seed by independent calculation on each individual entity. This scheme is observed to be highly scalable by storing very small number of functions on each entity, thereby supporting a very large network. Every entity in the network is distributed with its own distinct secrets that are used to generate pair-wise keys in real-time and create an authenticated association between two entities. The scheme is resilient to traffic analysis attack, as the information exchanged between any two nodes does not reveal any information about the functions they utilize and communication during the handshake takes place in an encrypted form. The polynomial based scheme (PBS) is immune to node capture attacks as information revealed on the capture of the node does not affect the communication being carried out between any other pair of entities. Moreover, PBS has very low computational overhead. This authenticated association takes place in a chained manner, that is, when a new MSs arrives and successfully associates itself with an entity already authenticated with AAA server, results in the establishment of an association between the new entity and AAA server in a chained fashion.
1.3 Organization of Thesis

The thesis is organized in six chapters. Chapter 2 provides an overview over various key pre-distribution schemes proposed in literature. Our proposed scheme (ELKPD) is described in Chapter 3. In Chapter 4, we have performance results and security analysis of the ELKPD scheme. The following Chapter 5 has overview of work related to Polynomial Based scheme (PBS) for IEEE 802.16m networks along with the working of PBS. In this chapter we also perform Security analysis. The final chapter summarizes our work along with some ideas for future work that can be pursued for further improvement to our work.
2 Related Work on Enhanced Location based Key Pre-Distribution Scheme (ELKPD)

Various key pre-distribution schemes have been proposed in literature [5] [6] [7] [8] that allow establishing security between wireless sensor nodes. Two naïve schemes are as follows.

First, in a Master Key Approach, we have a single master key loaded in all the sensor nodes before deployment and then this key is used for encrypting and decrypting all the communication in the network. As this scheme seems very easy for implementation, with small storage overhead and absolutely no communication overhead, it is straightforward to break as well. Compromise of any single node will give away the master key and will render whole communication vulnerable to such an attack.
Other naïve approach for carrying out secure communication between sensor nodes is by using Basic or Simple key distribution scheme in which, every node of n sensors is loaded with n-1 distinct keys for communication with rest of the network. The advantage of this scheme is again its ease of implementation, providing absolute security as compromise of any node doesn’t affect any communication between rest of the uncompromised sensor nodes. The disadvantage with this scheme is that it needs a large number of keys to be stored in the memory of each sensor node and this number increases even further as the number of nodes in the network increases which is a big constraint. Also, addition of any new node to the network requires addition of keys to all nodes that have been already deployed, which is impractical to utilize.

A scheme has been proposed by Blom [6] in which a trusted party is responsible for providing a secret key to each of n participants and a public identifier. Symmetric matrix calculation enables any two nodes to independently create a shared key for securely communicating with each other. But a limitation associated with Blom’s scheme is that, if an attacker can compromise the keys of at least k nodes, then every shared key can be reconstructed and hence the whole network would be rendered vulnerable.

Another scheme that is generalization of Blom’s scheme was proposed by Blundo et al. [7] which is based on bivariate polynomial calculation and allows any set of t nodes to compute a common key between them, with some of them being secure against any compromised node. The limitation with this scheme is that once t or more number of nodes have been compromised, (where t is also degree of polynomial) the network is no longer secure.
A random graph theory based key pre-distribution scheme was proposed by Eschenauer et al. [8] where the sensor nodes are considered as nodes in the graph and a probability is associated with the link between the nodes. The advantage of this scheme is that it is more secure than naïve scheme such as Master Key approach and fewer keys are required than the basic distribution approach. Limitation of this scheme is that there is no guaranty of connectivity for a desired node and a node would get isolated if it doesn’t share a common key with its neighbors. The communication overhead in shared key discovery phase and path-key establishment phase is too high which doesn’t adhere to limited processing capabilities and restricted power resources of sensors.

A random key distribution scheme has been proposed by Chan et al. in [5] where two nodes need to share more than q keys where (q>1) to form a secure link with each other. It improves the resilience against the node capture as q is increased. But this scheme suffers from a large storage overhead and the network may not be completely connected.

All the schemes that have been discussed above suffer from another limitation of not being easily scalable, requiring high storage capacities, not providing unique symmetric key between every pair of nodes in the whole network and none of them takes advantage of the deployment location information of the sensor nodes. If this information is used judiciously, one can have a distribution where nodes that are closer to each other, can have more keys in common and nodes that are far away don’t need to have any shared keys. This minimizes the number of keys that need to be stored in the memory of sensor nodes and they could have more common keys to communicate with their immediate neighbors.
Du et al. in [12] have utilized information regarding the deployment of sensor nodes to distribute the keys in a way such that nodes physically close to each other, share a larger number of common keys than far away nodes. But the assumption made in this scheme is that we have pre-existing knowledge regarding the deployment location of the sensor nodes which is not possible in a random deployment.

Although another scheme proposed in [10] doesn’t require information regarding the deployment of each individual sensor node, it makes use of the knowledge of the group of nodes that are going to be deployed together. So, these nodes have more keys in common than any other sensor nodes which are not part of the group, and assuming that these group of nodes will be physically close to each other after deployment.

The scheme that has been proposed in this thesis, without having any prior knowledge regarding location of sensor nodes or group of sensor nodes provides keys to the nodes that are location dependent. It makes sure that the system has unique symmetric key between every pair of nodes, at all times with very limited storage overhead. It is also possible to add more sensor nodes to the system without any need for additional hardware or computational overhead. This makes it quite scalable and secured against attack based on eavesdropping of transferred information.
3 ELKPD Scheme

In this section we discuss our proposed Enhanced Location Dependent Key Pre-Distribution Scheme (ELDKP), which is a combination of Enhanced Approach for Random Key Pre-Distribution proposed by Cheng et al. [3] and Location Dependent Key Management (LDK) Using Random Key Pre-Distribution in Sensor Networks proposed by Anjum [4].

Assumptions made in this scheme are the same as that of LDK. Along with regular sensor nodes, some identical but special sensor nodes called Special Nodes are deployed which have the capability to transmit at different power levels which results in more than one transmission range. These Special Nodes are tamper-proof and their expensive nature makes them to be deployed optimally. The use of such Special Nodes has already been discussed in the literature for location determination of sensors [13] [14], while their deployment can be done in the same way.
In the ELDKP Scheme, \( S_n \) is the number of sensor nodes and \( S_p \) is the number of special nodes and the following three phases are executed in their lifetime.

- Key Generation and Pre-deployment Phase
- Key-Initialization Phase, and
- Communication Phase.

### 3.1 Key Generation and Pre-deployment Phase:

During the Pre-deployment Phase, one trusted Central Key Distribution agency, before deployment, preloads the Sensor and Special nodes with information that would be required in the later phases for secure communication. So, the information given to sensor nodes is:

a) Set of keys from a pre-existing pool \( KP \) that forms original key ring \( KR \) of the sensor node. This selection of keys is done in the same way as described in the Enhanced Approach for Random Key Pre-Distribution proposed by Cheng et al. [3], where each of \( L \) matrices have \( m^2 \) keys, making the total number of keys as:

\[
KP = L^2(m^2),
\]
\[
= L^2m^2 \quad \text{(3.1)}
\]
Figure 3.1(a) and 3.1(b) respectively show the distribution of keys in entire key pool and an individual key matrix where each entry in matrix is an unique randomly generated cryptographic key and $K_{i,x,y}$ refers to the key in the $i^{th}$ matrix and $x^{th}$ row and $y^{th}$ column of that matrix.

Now, out of this pool $KP$, $S$ numbers of matrices are randomly selected [3] such that:

$$S \geq \lceil (L + 1)/2 \rceil. \quad (3.2)$$

Where Ceiling $\lceil \cdot \rceil$ is a Ceiling function that returns the smallest integer above a non-integer number (e.g. $\lceil 2 \rceil = 2$, $\lceil 5.6 \rceil = 6$).
This makes sure that any two pair of sensors have at least one matrix in common. And from these selected matrices, one row and one column is randomly selected for each matrix and preloaded in the memory of sensors to form their original key ring KR.

![Common keys matrix for Nodes A and B]

**Figure 3.2. Illustrating the common keys between Nodes A and Node B**

For example, if the matrix illustrated in Figure 3.2 is the (4*4) common matrix (i.e., m=4 in this case) for the sensor nodes A and B, the row 1 and column 2 defines the keys for node A and row 3 column 4 defines keys for node B. We can clearly see that keys $K_{i,3,2}$ and $K_{i,1,4}$ are in common and no matter how we select the rows and column for the common matrix, we would always have at least 2 or more keys in common. This ensures a complete connectivity in the network as we will see later on in this chapter.

b) A unique Sensor Node ID (SId).

c) Identities of the keys preloaded in its memory.

d) A common key $K$, which would be used for having secure communication between Sensor- Sensor pair and Sensor- Special Node pair.
e) A Bivariate Polynomial Function BPf() that leads to a common seed for hashing function.

f) A Hashing function Hf() produces new keys from pre-existing pool after receiving beacon signals from any Special nodes.

g) A Hashing function ESS() to produce secure symmetric key for encrypted communication between Sensor nodes.

The Special nodes are also preloaded with some information such as:

a) A unique Special Node ID (SpId)

b) The same common key K, which has been loaded in sensors.

After pre-loading the sensors with this information, $S_n$ sensor nodes and $S_p$ special nodes are deployed in a uniform random fashion in the monitored area.

**3.2 Key Initialization Phase:**

Once the deployment of nodes has taken place, initialization phase begins. In this phase, special nodes start transmitting different beacon signals at different transmission ranges, which is actually a different random number for different ranges, encrypted with common key $K$. Once the nodes start receiving the beacon signals, they first decrypt the number by using the common key $K$ and then by using hashing function $Hf()$ to generates a new set of keys.
As we can see in Figure 3.3, we have a special node $S_{p1}$ at the center which is transmitting 3 beacon signals at 3 different transmission ranges. The sensor node $S_{n1}$ receives 2 encrypted beacon signals $R_2$ and $R_3$ which it decrypts with key $K$ and then applies Hashing function $Hf()$ with seed $R_3$ and $R_2$ to its initial key ring $IK_1[ ]$ to lead to a derived key ring $DK_1[ ]$ and in similar fashion sensor node $S_{n2}$ by using hash function on its initial key ring $IK_2[ ]$ leads to $DK_2[ ]$.

It may be noted that the array $DK_i[ ]$ is the final array, that has the keys derived by using $Hf()$ on $IK_i[ ]$ by using different seeds.

Therefore, the final number of keys in a sensor node is the multiplication of the number of initial keys it had with number of beacon signals it receives [3]. There can be some cases where a single sensor node receives beacon signals from more than one special nodes, which is actually good for the system as it increases its location dependent diversity of the keys.

![Figure 3.3. Illustrating Key Initialization](image-url)
At the end of key initialization phase, we are left with the set of keys that have the diversity based on their location and the initial random distribution. The final number of keys in a sensor node can be given by:

\[ \text{Number of keys} = N_{ki} + (NB_s \times N_{ki}), \]

\[ = N_{ki} \times (NB_s + 1). \quad (3.3) \]

where \( N_{ki} \) is Number of keys in initial key ring and \( NB_s \) is Number of beacon signals received.

During the deployment of number of special nodes, it is essential to make sure that no node remains uncovered. Otherwise, that sensor node will just have the initial key ring and no location dependent derived keys. Since these Special nodes are tamper proof, they are expensive in nature and having too much of them will add to the system cost. So, to optimize their number and assuming coverage of almost whole of the network in a uniform random deployment, we use the following equation:

\[ \lambda = \frac{-ln(1-f_a)\times\text{Area}}{\pi r^2}, \]

\( f_a \) being the fractional area required to be covered as a fraction of the total area. \( \lambda \) is the node density, i.e., the number of Special Nodes/ Area.

\[ \text{The number of Special Nodes} = \frac{(-\ln(1-f_a)\times\text{Area})}{(\pi r^2)}. \quad (3.4) \]

### 3.3 Communication Phase:
In this phase, adjacent sensor nodes exchange information to establish trust (i.e. secure communication key) between them. The sensor node broadcasts the handshake message encrypted by a common key $K$ to all its neighbors which includes its ID along with the identities of the keys preloaded in its memory which is decrypted by key $K$. For example, referring back to the sensor nodes A and B of Figure 3.2, Node A will give out in encrypted form, the message $[IDA, ... (i,1,2) ..]$ where $IDA$ is its sensor id and it has keys of $i^{th}$ matrix which are in 1$^{st}$ row and 2$^{nd}$ column. Similarly Node B will send in encrypted form, the message $[IDB, ... (i,3,4) ..]$ where $IDB$ is its sensor id and it has keys of $i^{th}$ matrix which are in 3$^{rd}$ row and 4$^{th}$ column. After exchange of this information, they can find out their common keys to be $K_{1,1,2}$ and $K_{1,1,4}$. 

Also, while storing the derived key ring, we need to ensure that the indices of the keys from initial key ring remain unchanged. The same broadcast message helps in finding common keys in derived key ring also. Once the nodes have found their common keys, they use their Ids to generate a common seed for Hashing function $E_{SS} ()$ by using Bivariate polynomial function $BP_f()$. Since Bivariate polynomial function $F(x,y) = F(y,x)$, if $x,y$ are replaced with Ids of the nodes, the common seed can be determined. For example, again referring back to Nodes A and B, in the Bivariate polynomial function $BP_f()$ taking Node id for node A $IDA$, the seed $Cs$ would be $BP_f(IDA,IDB)$ and for node B with id $IDB$ common seed would be $BP_f(IDB,IDA)$ but as $BP_f(IDA,IDB) = BP_f(IDB,IDA)$, we end up with a seed $Cs$ that is common to both nodes A and B. Along with the Ids and identity of keys, node A transmits a random time stamped number $A_{Rand}$ and in similar fashion B sends $B_{Rand}$ in the handshake message. After exchange of all this
information, one-way hashing function $E_{SS}()$ is used to generate the secure pair-wise communication key $SCK$.

$$SCK=E_{SS}(Cs, CK_1, CK_2, CK_3, .. CK_n, A_{Rand}, B_{Rand})$$

(3.5)

where $CK_1, CK_2, CK_3, .. CK_n$ are common keys between pair of nodes A and B.

Now, this $SCK$ is used for encrypting communication between the sensor nodes.

For key revocation, a revocation message is broadcasted by all special nodes to all sensor nodes. Thereafter, all derived keys are deleted and the random time stamped number i.e. $A_{Rand}$, $B_{Rand}$ are generated again. After which, the whole process of key initialization takes place again which leads to next-generation of derived keys and a new pair-wise key. It has to be noted that the original key ring still remains with sensor node and is not deleted, it is kept encrypted with common key $K$ already loaded in the memory of sensor node and is done to have unique symmetric key between every pair of nodes in the network.

With all this arrangement, addition of new sensor nodes is a trivial problem which can be addressed by carrying out the same process of key pre-distribution at first where it is loaded with same information as loaded in previous sensor nodes already deployed. Once deployed, the new sensor node starts receiving beacon signal from special nodes and generates its derived key ring and can start communicating with its neighboring sensor nodes and due to presence of initial key ring with all deployed sensor nodes it will always find at least a pair of keys in common for communication. We consider the performance issues in the next chapter.
4 Performance Results and Security analysis

In this section we investigate the performance of ELDKP and analyze its security performance as well. We also compare ELDKP against LDK by varying various parameters. The metric that is used for comparison is the Compromise Ratio which is defined as “The ratio of the number of secure links formed by the non-compromised nodes that have become vulnerable to the total number of secure links formed by non-compromised nodes in the network” [4]. For a perfectly secure scheme the compromise ratio would be 0, i.e., compromise of any node would not make the rest of uncompromised nodes vulnerable.

We have considered a uniform random distribution of sensor and special nodes in an area of 100*100 units, with default transmission range of special nodes as 15 units and of sensor nodes as 5 units. The value is obtained after 10 trials and averaging the results hence obtained. Each special node is assumed to have 5 times power levels of a sensor node.
As discussed earlier, the number of special nodes that have to be deployed in the wireless sensor network so as to have all sensor node remain covered while minimizing the number to prevent excessive cost. We use equation (3.4) to find out the optimum number of special nodes that need to be deployed in the network.

From Table 4.1, we can have 99.5% coverage of area with 75 special nodes at maximum transmission range R of 15 whereas 169 nodes with 10 units at the maximum transmission range are required to have the same area coverage. In simulations, we assume the number of special nodes as 125.

<table>
<thead>
<tr>
<th>f&lt;sub&gt;a&lt;/sub&gt;</th>
<th>R</th>
<th>S&lt;sub&gt;p&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>95%</td>
<td>15 units</td>
<td>43</td>
</tr>
<tr>
<td>96%</td>
<td>15 units</td>
<td>46</td>
</tr>
<tr>
<td>97%</td>
<td>15 units</td>
<td>50</td>
</tr>
<tr>
<td>98%</td>
<td>15 units</td>
<td>56</td>
</tr>
<tr>
<td>99%</td>
<td>15 units</td>
<td>66</td>
</tr>
<tr>
<td>99.50%</td>
<td>15 units</td>
<td>75</td>
</tr>
<tr>
<td>95%</td>
<td>10 units</td>
<td>96</td>
</tr>
<tr>
<td>96%</td>
<td>10 units</td>
<td>103</td>
</tr>
<tr>
<td>97%</td>
<td>10 units</td>
<td>112</td>
</tr>
<tr>
<td>98%</td>
<td>10 units</td>
<td>125</td>
</tr>
<tr>
<td>99%</td>
<td>10 units</td>
<td>147</td>
</tr>
<tr>
<td>99.50%</td>
<td>10 units</td>
<td>169</td>
</tr>
</tbody>
</table>

Table 4.1: Illustrating optimum number of Special nodes required.

Where, f<sub>a</sub>: Fractional area covered;

R: Maximum transmission range of special node;
S_p: Calculated value of special node.

In Table 4.2, we can see the number of Beacon signals received (NB_s) and the number of neighbors of each sensor node (Avg.) are function of the number of special node and transmission range of sensor nodes is varied. The number of beacon signals received is an important parameter to be considered as from equation (3), it decides the final number of derived keys that would be residing in the memory of sensor nodes and this number should not be too large which we can see from results that in our scheme varies from 16 to 20.

<table>
<thead>
<tr>
<th>S_n</th>
<th>SR_s</th>
<th>Avg. Neighbor count:</th>
<th>No. Unconnected S_n</th>
<th>S_p</th>
<th>NB_s</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>100</td>
<td>16</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>17</td>
<td>0</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>500</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>125</td>
<td>20</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>17</td>
<td>0</td>
<td>125</td>
<td>20</td>
</tr>
<tr>
<td>1000</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>125</td>
<td>20</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td>35</td>
<td>0</td>
<td>125</td>
<td>20</td>
</tr>
<tr>
<td>1000</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td>34</td>
<td>0</td>
<td>100</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 4.2: Illustrating the number of Beacon signals received and number of neighbors of each sensor node as number of special node and transmission range of sensor nodes is varied.

Where, S_n: Number of sensor nodes deployed;

SR_s: Maximum transmission range of sensor nodes;

Avg. Neighbor count: Average number of neighbors of each Sensor node;

No. Unconnected S_n: Number of unconnected sensor nodes;

S_p: Number of special nodes;
NBₙ: Number of beacon signals received by each sensor node.

Network connectivity is an important parameter in evaluating the effectiveness of any key distribution mechanism, with a good scheme would have higher connectivity and low communication overheads. As we discussed in chapter 2 about a scheme based on graph theory [8], we observe that it does not provides us with unique communication key between every pair of sensor nodes and information needs to be exchanged up to three or more hops to setup a secured key which introduces additional communication overhead as information is exchanged between neighboring sensors. In our scheme as we can see in (Figure 4.1), complete connectivity is realized by just single exchange of information over a single hop. Schemes [3] [9] also match with our scheme in this respect.

![Connectivity versus Number of Hop Counts to setup Pair-wise key.](image)

**Figure 4.1. Connectivity versus Number of Hop Counts to setup Pair-wise key.**

We also compare our scheme with LDK by utilizing the compromise ratio as the performance metric, and we can see in Figure 4.2, that by increasing the number of power levels,
performance of LDK improves considerably. But, it requires a large key pool to generate pre-distributed key ring of sensor nodes while the key ring of LDK is considerably larger than our scheme. Compromise of any node doesn’t reveal any information regarding uncompromised nodes in our scheme. For the same reason in (Figure 4.3), increasing the transmission radius of special nodes initially decreases the compromise ratio of LDK and then increases as diversity of keys tends to decrease. But this does not have any adverse effect on our scheme and it still provides the best possible compromise ratio of 0.

Figure 4.2. Impact of varying Power Levels on Compromise Ratio on ELDKP, with 500 sensors and 125 Special Nodes and LDK with 500 sensors and 200 Special Nodes.
*Results of LDK have been reproduced from [4].
Figure 4.3. Impact of varying Maximum Transmission Radius on Compromise Ratio

*Results of LDK have been reproduced from [4].

Our scheme is also resilient against replay attack and traffic analysis attack as these attacks are based upon the inability of network to update its keys and always the same keys for communication. In our proposed scheme, we can update the keys in the network either periodically or on request. There is also a provision for key revocation once the previous set of keys has been compromised.
Chapter 5

5 Use of Polynomial based scheme (PBS) in Wireless mesh Network [17]

Wireless mesh networks (WMNs) are networks consisting of two types of nodes, Mesh Routers (MRs) and Mesh Clients (MCs) where MRs maintain arbitrary ad-hoc connectivity automatically between them [8] and provide wireless services to MCs. A typical WMN has been illustrated in the Fig. 1. WMNs have become increasingly important due to their deployment flexibility while keeping the cost at a fairly low level. Their usefulness has been further enhanced by introducing MRs that increase coverage of the special MRs also known as IGW (Internet Gateway) in WMN terminology. In such a flexible scenario, the network traffic has to be propagated for authentication from the MCs to the AAA Server (authentication, authorization and accounting) through MRs in a multi-hop fashion. This is similar to the packets traveling in an ad-hoc network and we need to have a robust scheme that could protect the integrity of propagating message in the WMN. Such networks suffer from threats like false handoffs and fake registrations, and the
network is also prone to attacks like Traffic analysis and Node Capture attack. This makes the requirement of having a robust WMN scheme more desirable.

Any MC can be connected directly to an AAA server or through an IGW for AAA services. Instead, we would like MRs to have this capability so as to support mobility of MCs in a distributed manner throughout the WMN. In such scenarios, we need to employ a scheme that is both secure and consistent. Another challenge is to keep the communication alive and secured, even in the presence of a handoff. Unlike a sensor network where we can code the keys on the sensors before actual deployment, MCs in a WMN topology have different levels of mobility and we need to have dynamic keys which can be generated on the fly, ensuring the communication is secured. We propose to use a polynomial based distributed AAA scheme between any two entities within each other's communication range so as to achieve secure communication by generating keys in a real-time manner.

In our proposed scheme, we use bivariate polynomials that enable us to compute a common seed by independent calculation on each individual entity. This scheme is observed to be highly scalable by storing a very small number of functions on each entity, thereby supporting a very large network. Every entity in the network has been distributed with its own distinct secret polynomials that are used to generate pair-wise keys in real-time and create an authenticated association between two entities. The scheme is resilient to traffic analysis attack, as the information exchanged between any two entities does not reveal any information about the functions they utilize, and communication during the handshake takes place in an encrypted form. The polynomial based scheme (PBS) is immune to node capture attacks as information revealed on the capture of the node does not affect the communication being carried out between any other pair of entities. Moreover, PBS has a very low computational overhead. This
authenticated association (AA) takes place in a chained manner, that is, when a new MC arrives and successfully associates itself with an MR already authenticated with AAA server, results in an association between the new entity and AAA server is established in a chained fashion.

Figure 5.1. A Generic Wireless Mesh Network (WMN) architecture.

5.1 Proposed Scheme
In this section, we introduce a distributed scheme for establishing a secure connection between pair of entities on the fly, the pair could be of (MC↔MC or MC↔MR or MR↔MR or MR↔IGW or MC↔IGW or IGW↔IGW). The basis of this scheme is that a central authorized server distributes a set of secrets to the components of the network before deployment. At a later stage, when the network is in operation, the pair of entities wanting to establish an AA connection, share a part of their information and perform individual computations to reach the same secured key which can be used as symmetric key for communication between them.

For establishing secure communication, the following secrets are distributed by the central server to all the entities:

1. Common key (K) which would be used for encrypting all the communication between entities prior to establishment of a secured symmetric key.

2. A set of Bivariate Polynomials Functions $F_{i,j,k}(x, y)$ (where $0 \leq i \leq l; \ 0 \leq j \leq m; \ 0 \leq k \leq m$) selected randomly from a three-dimensional matrix of polynomials along with indices of polynomial functions.

3. Hashing function $H_d()$ to hash together the seeds generated by computing the bivariate polynomials and obtain a secure symmetric key for encrypted communication.

The scheme has three stages:

1. **Bivariate Polynomial Distribution**: Where polynomials are randomly selected from a matrix of polynomials and given to the entity by using Yi-Cheng’s scheme [4,5] of distribution.
2. **Authenticated Association**: In this stage exchange of secrets and evaluation of common secure key by independent calculations take place.

3. **Secure Path Generation**: Chained secure path establishment from entity to AAA server.

---

### 5.1.a Bivariate polynomial distribution

A bivariate polynomial $F_{i,j,k}(x,y)$ of degree $p$ which is defined as:

$$F_{i,j,k}(x,y) = \sum_{r+s=p} a_{rs} x^r y^s$$  \hspace{1cm} (5.1)

where the coefficients $a_{rs}$ are randomly chosen over a Finite Field $Gf(Q)$ where $Q$ is a sufficiently large prime number and $i,j,k$ are the indices for the position of polynomial in a three-dimensional space.

Also, the polynomial satisfies the following property:

$$F_{i,j,k}(x,y) = F_{i,j,k}(y,x)$$  \hspace{1cm} (5.2)
Figure 5.2. Illustrating distribution of bivariate polynomial functions in a three dimensional matrix.

Once a large number of polynomials have been generated, a limited number of polynomials are randomly selected and placed in a three-dimensional matrix. This matrix can be viewed as $i$ number of two dimensional square matrices each with degree $t$, where $i$ vary between $0$ and $l$ and $t$ varies between $0$ and $m$. Hence, the total number of selected polynomials ($TNp$) from finite field to the matrix would be:

$$TNp=l \times m \times m \quad (5.3)$$
We can see from Figure 5.2, a bivariate polynomial $F_{i,j,k}(x,y)$ is at the position $i,j,k$ in the polynomial matrix.

Once the polynomials have been defined to form a three dimensional matrix, they need to be distributed as components of each entity within the WMN. This is done in two stages. In first stage out of total $l$ matrices, $S$ are randomly selected such that:

$$
S \geq \left\lceil \frac{(l+1)}{2} \right\rceil
$$

(5.4)

Where $\lceil \cdot \rceil$ is Ceiling Function which returns the smallest integer above a non-integer number (For example, $\lceil 5.2 \rceil = 6$, $\lceil 4 \rceil = 4$).

This condition makes sure that two sets share at least one common matrix between them. Once matrices have been selected, a row and column are randomly selected from each matrix, and all the functions present in that particular row and column are distributed to the entity. For example in Figure 5.3, we can see that, for $i$th selected matrix with $(m=5)$, component will be distributed all the highlighted functions in 2nd row and 4th column. By having $s$ number of matrices for all $m$ functions from a randomly selected row and $m-1$ functions from a randomly selected column, leading to a total of $(m-1)+m$ functions are selected from each matrix and the total number of functions (TFn) distributed to each entity is given by:
Once the functions have been chosen for the distribution, \( F_{i,j,k}(x,y) \) is evaluated for \( x = \text{ID of the entity} \) and then distributed, which means that for an entity A with ID IDa, polynomial received would be \( F_{i,j,k}(\text{IDA},y) \). After distribution of functions, we can look at some special properties of this distribution. In Figure 5.4, there is an \( i^{th} \) matrix that is common for two entities. For the first entity 2\(^{nd} \) row and 4\(^{th} \) column has been selected and for second entity 4\(^{th} \) row and 3\(^{rd} \) column has been selected. We can clearly observe that, between two matrices, functions \( F_{i,1,2}(\cdot) \) and \( F_{i,3,3}(\cdot) \) are in common or, in other words, any two entities can have at least two functions that will be shared with each other. This ensures that any two entities can establish AA between the two, once they contain functions from the central server.

\[
TFn = s \times ((m-1)+m),
\]
\[
=s \times (2 \times m -1)
\] (5.5)

5.1.b Authenticated Association

Once the secrets have been distributed, the entities can start their network operation. To communicate with other entities, they need to establish AA by sharing of some information and
carrying out independent calculations. So, in a handshake message, the following information is exchanged by encrypting a common key $K$ and later decrypted at receiver by the same key $K$ as all nodes share this master key, indices of their functions and their IDs. For example, we have two entities A and B with IDs $ID_a$ and $ID_b$ respectively and let us assume $F_{1,2,3}(x,y)$, $F_{2,3,3}(x,y)$, are the two functions they share with each other which are obtained after exchanging their indices. To have AA, two individual computations take place at entities A and B.

**At A:**

$$SD_{a,1} = F_{1,2,3}(ID_a, ID_b), \quad (5.6)$$

where $SD_{a,1}$ is Seed 1 for entity A.

$$SD_{a,2} = F_{2,3,3}(ID_a, ID_b), \quad (5.7)$$

where $SD_{a,2}$ is Seed 2 for entity A.

**At B:**

$$SD_{b,1} = F_{1,2,3}(ID_b, ID_a), \quad (5.8)$$

where $SD_{b,1}$ is Seed 1 for entity B.

$$SD_{b,2} = F_{2,3,3}(ID_b, ID_a), \quad (5.9)$$

where $SD_{b,2}$ is Seed 2 for entity B.

But, in a bivariate polynomial, $F(x,y) = F(y,x)$. Therefore $SD_{a,1} = SD_{b,1}$ and $SD_{a,2} = SD_{b,2}$.

Finally, one-way hashing function $H_f()$ is applied to all seeds to generate a unique secure symmetric key ($USSK$) for communication between two entities.

$$USSK = H_f(SD_{ID,1}, SD_{ID,2}, \ldots). \quad (5.10)$$

If two entities share more than two common functions, we can have more than two seeds in equation $5.10$ and the generated key will be even more secured. Generation of $USSK$ by individual computations establishes an AA between the two entities.
5.1.c Secure Path Generation

We now consider the establishment of a secured authenticated path from individual entity to AAA Server in a WMN. In Figure 5.5, $MR_i$ is for $i^{th}$ Mesh Router, $MC_i$ is for $i^{th}$ Mesh Client, $IGW_i$ is for $i^{th}$ Internet Gateway, $AA_i$ corresponds to $i^{th}$ Authentic Association between two entities and $(AA_i \leftarrow AA_j)$ corresponds to pair-wise association between the two entities. Consider a case when network is in operation and $AA_1$ and $AA_6$ (Authentic Association between $IGW_i$ and AAA and Authentic Association between $MR_i$ and $IGW_i$ respectively) are already established, as previous stage ensures authenticated association between any two entities that are part of the network. When a new Mesh Client $MC_1$ joins the network and forms $AA_8$ with $MR_1$. Establishment of $AA_8$ enables $MC_2$ to have $AA_{11}$ and $AA_{12}$ in a chained hop-wise fashion. As when we already have $AA_6$ and $AA_1$ established, $MC_2$ gets automatically authenticated with $MR_1$ which is denoted by $AA_{11}$ (Where $AA_{11} = AA_8 \leftrightarrow AA_6$) and since we have $AA_1$ already established, hop-wise chained association extends even further till AAA server and the association is termed as $AA_{12}$ (Where $AA_{12} = AA_{12} \leftrightarrow AA_1$).

Consider another case when $MC_5$ while roaming, comes in communication range of $IGW_2$. In this case it does not need to communicate with any MRs to establish association with IGW; therefore it establishes $AA_3$ with the $IGW_2$, which had already established $AA_2$ with AAA server. In other words $MC_5$ establishes $AA_{13}$ (where $AA_{13} = AA_3 \leftrightarrow AA_2$) with AAA Server. We can clearly see through this example that a secure path generation is done entirely on the fly and without any additional overhead for communication, once security mechanism has been done with the entities of the WMN.
Network connectivity is an important parameter to evaluate any scheme that is expected to provide secure connections between participating entities. Our proposed scheme guarantees a shared key between any two entities as there exists at least one common bivariate polynomial. The entities can reach to a common seed by individual computation in each entity and provide a secured symmetric key which can be used for encrypting and decrypting the data. The lack of any shared polynomial means the absence of the connectivity. The distribution of polynomials in our scheme makes sure that the conditions for complete connectivity are always met by
exchanging information over a single hop, as described in the second stage of our scheme. Mobility is an important parameter to consider for MCs in a WMN. While there is no adverse effect on the connectivity in our scheme due to mobility of MCs, we compare our scheme with conceptually similar schemes of key pre-distribution described for WSNs. The only difference is the distribution of functions to entities instead of keys.

For the simulation results in Figure 5.6, uniform random distribution of entities is considered in an area of 100*100 units with default transmission range of MCs as 5 units and of MRs as 15 units. Each value is obtained after 50 trials and averaging the values hence obtained. We observe from Figure 5.6, that our scheme along with [3],[9] establishes a pair-wise key just after exchange of information over single hop, whereas in [8], it takes exchange of up to 4 hops information to establish the same level of connectivity. This shows that our scheme has a much lower communication overhead. In Figure 5.6, k represents key ring size.

![Figure 5.6. Connectivity versus Number of Hop counts to set up pairwise key](image)

Figure 5.6. Connectivity versus Number of Hop counts to set up pairwise key
5.1.e Scalability of Network

By mathematical analysis, our PBS scheme is observed to be highly scalable, despite of the fact that a minimum number of functions are stored in each individual entity. Let’s consider a distribution of functions where we have \( l \) function matrices, each having \( m \) rows and \( m \) columns. Given \( s \) is the number of randomly selected matrices used for distribution, the number of entities \( N_e \) that can be supported [3] is then given by:

\[
N_e = \binom{l}{s} \times (m \times m)^s = \frac{l!}{s! \times (l-s)!} \times m^{2s}, \tag{5.11}
\]

By using equations (5.5) and (5.11), we can construct distribution of functions for entries as given in Table 5.1 where \( l = 4 \) and \( s = 2 \). We can clearly see that just by storing 30 functions on a single entity we can support a WMN of a size 24,576 entities which makes this scheme easily scalable for a very large WMN.

Also from the work in [9], we know that by having key ring size of 50, the work in [8] can only support 317 entities while [5] can provide support for 367 entities when \( q=2 \) or 408 entities when \( q=3 \). By storing the same number of functions (50), our scheme achieves the same results as in [3], with maximum supported network size of 85,638 entities, which is significantly more than [5],[8].

From equation (5.5), we have \( m \) in terms of \( Tfn \) and \( s \):

\[
m = \frac{(Tfn+s)}{s} \tag{5.12}
\]

Substituting value of \( m \) from equation (5.12) in equation (5.11) we get:

\[
N_e = \frac{l!}{s! \times (l-s)!} \times ((Tfn + s)/s)^{2s} \tag{5.13}
\]
Equation 5.13 [17] illustrates the relationship between $N_e$ and $T_f$ for the network which shows that for a given $m$, the number of entities that can be supported in the network by storing $T_f$ number of functions on a given entity. This equation also emphasizes the fact that there is a tradeoff between the storage that will be required to store the functions on the entity and the number of entities that will be supported in the network.

<table>
<thead>
<tr>
<th>$l$</th>
<th>$s$</th>
<th>$M$</th>
<th>$T_f$</th>
<th>$N_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>96</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>486</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>1536</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>18</td>
<td>3750</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>22</td>
<td>7776</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7</td>
<td>26</td>
<td>14406</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8</td>
<td>30</td>
<td>24576</td>
</tr>
</tbody>
</table>

Table 5.1. Network Size Vs. Number of functions stored on single entity.

5.1.f Resilience against Node Capture Attack and Traffic analysis attack

Node Capture attack is a serious threat in these WMNs where captured node can reveal the secrets of non-captured nodes and increase their vulnerability by attacks. The problem associated with [5],[8] is that different pair of nodes might use same key for securing communication in the network which means compromised nodes can reveal information regarding non-captured nodes. Whereas in our scheme like [3], every pair of entities generates a unique key for communication by independent calculations, capture of one entity does not affect the security between non-captured nodes just like in [3].

Also, at no point, any information is exchanged regarding the functions distributed to the entities and the only information that is exchanged is Ids of the entities and indices of functions.
which are transmitted in encrypted form in a handshake message. These make our scheme quite secure against traffic analysis attack. Another point to be noted is that the use of one sided hash function to generate key by hashing all the seeds makes this scheme even more secure.
6 Conclusions and Future Work

Distribution of keys to sensor nodes before deployment is a nontrivial problem and has been widely discussed in literature, with each scheme having its own set of merits and demerits. Our proposed scheme uses small key ring for sensors which makes it useful for sensors having small sized memory and along with which it provides complete pair-wise secure connectivity between sensor nodes. Our proposed scheme provides sensor keys to be changed dynamically depending on upon its location without any additional hardware. Our scheme is quite scalable, which enables addition of sensors nodes to the network at any point of time without any additional computational or hardware overheads.

Future Work for ELKPD:

In our current scheme, we don’t delete the initial key ring so as to have complete connectivity at all times and just keep the keys in an encrypted form. We propose to improve this by using series of chained hash functions exhibiting the following property:

Let the series of functions be $F, F_1, F_2, F_3, \text{and } F_4$ and so on.

The functions should be such that they satisfy this property.
\[ F(a,b,c...) = (a_1,b_1,c_1,...) \]

\[ F_1(a_1,b_1,c_1,...) = (a_2,b_2,c_2,...) \text{ and also } F_1(a,b,c...) = (a_2,b_2,c_2,...) \]

\[ F_2(a_2,b_2,c_2,...) = (a_3,b_3,c_3,...) \text{ and also } F_2(a_1,b_1,c_1,...) = (a_3,b_3,c_3,...) \text{ and also } F_2(a,b,c...) = (a_3,b_3,c_3,...) \]

\[ F_3(a_3,b_3,c_3,...) = (a_4,b_4,c_4,...) \text{ and also } F_3(a_2,b_2,c_2,...) = (a_4,b_4,c_4,...) \text{ and also } F_3(a_1,b_1,c_1,...) = (a_4,b_4,c_4,...) \text{ and also } F_3(a,b,c...) = (a_4,b_4,c_4,...) \]

and so on.

Where \((a,b,c.. )\) are keys of initial key ring and subsequent numbers hence generated are derived keys.

These functions will make sure that we don’t have to keep the initial key ring in the sensor nodes and any node can join the network at any point of time and can reach to the same set of keys as of present generation.

All these features make our scheme quite useful for the purpose of secure communication in wireless sensor networks (WSNs).

For WMNs where mobility of MCs poses a considerable problem in providing secure associations, PBS provides an effective solution by carrying out independent calculations to agree on a common key along with having low communication and storage overheads. The proposed scheme is highly scalable by supporting a network of large number of entities, despite storing very few functions on each individual entity. PBS also provides security against attacks.
such as Traffic analysis attack and Node capture attack in an effective manner. Since we can have communication between any pair of entities irrespective of their position and mobility, our scheme is equally applicable for IEEE 802.16m based WiMax based systems where Relay Stations can also be mobile [15],[16].

**Future Work for PBS:**

For future work of PBS, it would bevaluably informative to analyze the storage overhead and security performance when applied to IEEE 802.16m based WiMax Networks due to the presence of mobile relay stations. And it will provide deeper insight for implementing the changes to make PBS even more robust for WiMax Networks.
References

Bibliography


