HIGHLY SECURE KEY PREDISTRIBUTION USING AFFINE PLANES AND REED MULLER CODES IN WIRELESS SENSOR NETWORKS

Pinaki Sarkar*[1], Amrita Saha[2] , Samiran Bag[3]

[1]Department of Mathematics, Jadavpur University, Kolkata-700032, INDIA
pinakisark@gmail.com
[2]CSE Department, IIT Bombay, Mumbai-400076, INDIA
amrita@cse.iitb.ac.in
[3]Applied Statistics Unit, Indian Statistical Institute, Kolkata-700108, INDIA
samiran_r@isical.ac.in

ABSTRACT

Wireless Sensor Networks (WSN) consist of low powered and resource constrained sensor nodes which are left unattended for long duration of time. Hence it is very challenging to design and implement cost effective security protocols for such networks. Thus symmetric key cryptographic techniques are preferred over public key techniques for communication in such scenarios. Prior to deployment, keys are usually predistributed into the nodes and this problem has been well studied. Highlighting that connectivity and communication are two separate aspects of a WSN, we propose a secure connectivity model using Reed Muller codes. The model is then utilized to securely establish communication keys and exchange messages in a WSN designed on the basis of a scheme that uses affine planes for key predistribution. By the introduction of connectivity model, the node identifiers (ids) are converted from public to private information to each node. These private node ids can be used to generate new communication keys from old ones by applying cryptographic hash functions. Novel combination of these ideas yields highly resilient communication model with full connectivity between nodes.

KEYWORDS

Security, Connectivity, Communication, Reed-Muller Codes, Affine Planes, Hash functions.

1 INTRODUCTION

Wireless sensor networks consist of tiny sensor nodes that have very limited battery power, less amount of storage, low computational power and they are scattered in large numbers over a vast region. The sensors communicate between each other and with the base station via radio frequencies. These networks are used in civilian purposes like smoke detection, wild fire detection, seismic activity monitoring, ocean temperature monitoring, salinity monitoring of sea water. Besides they have large application in military purposes, for instance monitoring enemy movements. Clearly, the nodes deal with very sensitive data and can communicate within a special range called Radio Frequency range. Since sensors are deployed unattended over the target area this makes them physically insecure and prone to adversarial attacks. Thus arises the need of secure communication model in WSN to circumvent these attacks.

DOI : 10.5121/ijcnc.2011.3519

A secure communication model makes use of (low cost) cryptographic primitives. Existing schemes like Kerberos [12] & public key cryptography [6] are not suitable to this kind of resource constrained system due to inherent cost associated to them.

Key predistribution is a method to preload cryptographic keys in sensor nodes before they are deployed in the target area. It is a symmetric key approach, where two communicating nodes share a common secret key. The message encrypted decrypted using the same secret key. Thus both the sender and receiver nodes must be preloaded with the same key. So prior to deployment every node has to be preloaded with a set of keys called its *key ring* or *key chain*. A centralized authority called Base Station or Key Distribution Server (KDS) preloads the *key ring* of every node from a pool (aka *key pool*) of keys meant for the entire network. Immediately after deployment shared keys are to be established between nodes before actual communication. This phase is called shared key discovery. In absence of common (shared) keys between two sensors a path-key need to be established between them (aka path key establishment).

1.1 RELATED WORK

Key predistribution in sensor networks was first considered by Eschenaur and Gligor [5]. In their work ever key is associated with an unique *key identifier*. Keys are randomly drawn from the *key pool* to form the *Key rings* of the sensors. *Key establishment* is also random. Such method of key predistributuion is *probabilistic* in the sense that both key distribution and etablishment is done randomly. Many such *probabilistic key predistribution* schemes have been well studied and presented in a survey report published in 2005 by Çampete and Yenner [2].

Shared key establishment and *Path key discovery* can become very difficult task for above probabilistic approaches. Lee and Stinson proposed two schemes [7, 8] where they have adopted combinatorial techniques for predistribuion and later establishment of keys. Their works also suggests that both *shared key establishment* and *path key discovery* can be better achieved by the suggested *deterministic approach*.

Chakrabarti et al. [3] proposed a *hybrid* key predistribution scheme by merging the blocks randomly over combinatorial designs. They randomly selected blocks from transversal design proposed by Lee and Stinson [7, 8] and merged them to form the sensor nodes. Though this technique increase the key ring sizes per node, it improves the resilience and communication probability of the network. Ruj & Roy [10, 11] used several combinatorial and coding techniques like Partially balanced incomplete block designs (PBIBD), transversal design, Reed-Solomon codes etc. to predistribute keys.

1.2 OUR CONTRIBUTION

Very recently, Bag and Ruj [1] have utilized *finite affine geometry* to propose a deterministic key predistribution scheme. In this paper we discuss enhancement of resiliency of their scheme. Their scheme uses finite affine plane over Z_q, where q is a prime. For this we observe that *communication* and *connectivity* are *two separate aspects* of a WSN. Then apply *Reed Muller Codes* to model the connectivity aspect so as to make it secure by using suitable *cryptosystems*. To the best of our knowledge, this novel idea of separating connectivity from communication and then applying a secure model to the connectivity aspect of a WSN been proposed for the first time by Sarkar et al. in [13]. Combination of both the schemes results in a highly resilient key predistribution scheme for a WSN providing *full connectivity* amongst the nodes.

1.3 BASIC NOTIONS

Before explicitly explaining the various aspect of our design, we require some basic notions like communication, connectivity, the respective key and communication radius which have been stated in [13, section II]. Throughout the paper we shall use the term ``Uncompromised nodes" to

mean nodes that are not compromised. The words ``communication" and ``connectivity/connection" are sometimes abbreviated to com. and con. respectively. The terms ``communication model/scheme" and ``key predistribution model/scheme" will mean the same.

2 COMMUNICATION MODEL

Our design is based on a scheme by Bag & Ruj [1]. In their scheme the authors used finite affine plane over Z_q where q is a prime number. Affine plane over Z_q contains as many as q^2 points and are usually denoted by $AG(2,q)$. The entire key space is split into 4 parts, each part containing $\lfloor \frac{q^2}{4} \rfloor$ points and from each part the i^{th} point is assigned to the i^{th} node. Thus there are a total of $\lfloor \frac{q^2}{4} \rfloor$ nodes, each containing precisely 4 points. The lines through all 4 points of a node represent the set of keys in that particular node. As demonstrated in [1, section VI] there can be $4q-2$ to $4q+1$ keys belonging to any node . The lines through any two points of two distinct nodes serve as the identifier of a common keys between the nodes. The authors showed in [1, section VI] that there can be 1 to 16 common keys between a pair of nodes.

Suppose 2 nodes with id i and j want to establish their common keys. They do so by finding lines through any two points belonging to them as follows: The points are distributed among the nodes in such a fashion that the node's ids reveal the points they contain. Thus on receiving the id of node j, node i gets to know the points in node j. So it can find one line passing through any of its 4 points and any of the points of node j. Similarly if node j uses the same algorithm as node i it will end up finding the same line as node i. As these lines represents the ids of the shared keys between the nodes, the nodes can communicate with thus established common keys.

3 WEAKNESS: MOTIVATION OF OUR WORK

We observe a weakness in the aforesaid key predistribution scheme. Here the node ids reveal the points inside a particular node. Let us say node i and node j want to establish their keys securely. An adversary, say Alice can tap the radio frequency channel and come to know the unencrypted node ids passing through them. She can then find the key ids of the shared keys between the sensors in a manner similar to the computation done by the nodes. This clearly implies that selective node attack is quite feasible.

These points are again contained in a number of nodes of the sensor network. She can capture one of them and get to know the actual keys. Combined with the knowledge of node ids, she can use these keys to actually affect the com. amongst other nodes.

To counter this problem, we first differentiate the two aspects communication and connectivity of a WSN. Then like in [13], apply Reed Muller Codes to suitably model the connectivity aspect. The construction of the model is presented in the following section. The model can be made secure by using suitable cryptosystems.

As shall be later established the combination of the two ideas results in a highly resilient key predistribution scheme for WSN providing full connectivity amongst nodes with virtually same communication overhead.

4 PROPOSED CONNECTIVITY MODEL

Reed Muller codes will be utilized to structure the connectivity aspect of the WSN. These codes have been elaborately described in [4] and necessary notational changes have been highlighted by Sarkar et al. in [13, section IV]. We follow similar procedure as described in [13, section IV] baring some modification to be illustrated now.

First our model will always have three tiers with the ``Base Station'' or ``KDS'' in the $1st$ or topmost tier. The second tier will consist of $\lceil \frac{q}{4} \rceil$ newly introduced cluster heads (CHs). Amongst these $\lfloor \frac{q}{4} \rfloor$ will be assigned q many nodes in the $3rd$ and the last level. Whereas $l = \lfloor \frac{q^2}{4} \rfloor - q\lfloor \frac{q}{4} \rfloor$ nodes has to be under the remaining 1 CH in the last level. Thus our model needs an extra $\lceil \frac{q}{4} \rceil$ many CHs and can support $\lfloor \frac{q^2}{4} \rfloor$ ordinary nodes (at the last level).

It is evident that current connectivity model is heterogeneous in nature, i.e., has different number of nodes in its various clusters. This along with the fact that exactly three tiers are required for our connectivity model distinguishes our design from the original design of Sarkar et al. in [13, section IV].

To build up the cluster between the various tiers of the connectivity model, we shall make use of first order Reed Muller codes. For connectivity of $1st$ and $2nd$ levels, we employ a m complete graph where $m = \lceil \frac{q}{4} \rceil$. We consider $\mathbb{Z}_2[x_1, x_2, \ldots, x_{\lceil \frac{q}{4} \rceil}]$ in much the same manner as the authors of [13] had considered $\mathbb{Z}_2[x_1, x_2, \ldots, x_m]$. Like in [13], the monomials x_i will represent the bit pattern of length $2^{\lceil \frac{q}{4} \rceil}$ having 2^{i-1} 1's followed by 2^{i-1} 0's where $1 \le i \le \lceil \frac{q}{4} \rceil$. A sample connectivity pattern for a cluster containing KDS and 3 CHs can be represented by the following matrix

$$\begin{bmatrix} \mathbf{KDS} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \mathbf{CH_1} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{CH_2} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{CH_3} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \end{bmatrix}$$

Matrices like the above one are used for construction of Reed Muller codes. This particular matrix has been referred to as $R(1;3)$ in [4]. Here 1 means the degree of the monomials is `1' and 3 stands for the number of variables. The significance of the entries 1 and 0 in the above matrix, *R(1;3)*, is the presence and absence of a connectivity link at that row and column position respectively. Thus for connectivity of two any entities (KDS/CHs/nodes), both of them should have a 1 in the same column for at least one column. Each column is assigned a separate connectivity key immaterial of them using the same radio frequency channel.

The connectivity pattern between of each of the clusters of the $2nd$ and $3rd$ level is meant to be a 2 complete graph having $m = q$ variables (nodes) in the matrix. Thus we look at

$Z_2[x_1, x_2, \ldots, x_q]$ as was similarly done in [13, section IV, subsection B] Connectivity matrix for a cluster having 1 CH and 3 nodes is as follows:

$$\begin{bmatrix} \mathbf{CH} & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ \mathbf{N_1} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{N_2} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{N_3} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The construction of the second matrix from the first can be found in [13, Section IV, Subsection B]. Here KDS is not present in the inter-nodal links. There is a broadcast channel and a provision for external only for KDS. In the present case instead of 3, we look at q or l many nodes. Here again wherever there is 1, connectivity link is present.

Figure 1 give an lively example with $q = 11$. There are $\lceil \frac{11}{4} \rceil = 3$ CHs in $2nd$ tier. This model supports $\lfloor \frac{11^2}{4} \rfloor = 30$ sensors in the $3rd$ & last level. Out of these 42 sensors, $11 * 2 = 22$ will be under 2 CHs and only $30 - 22 = 8$ under the remaining CH of $2nd$ level.

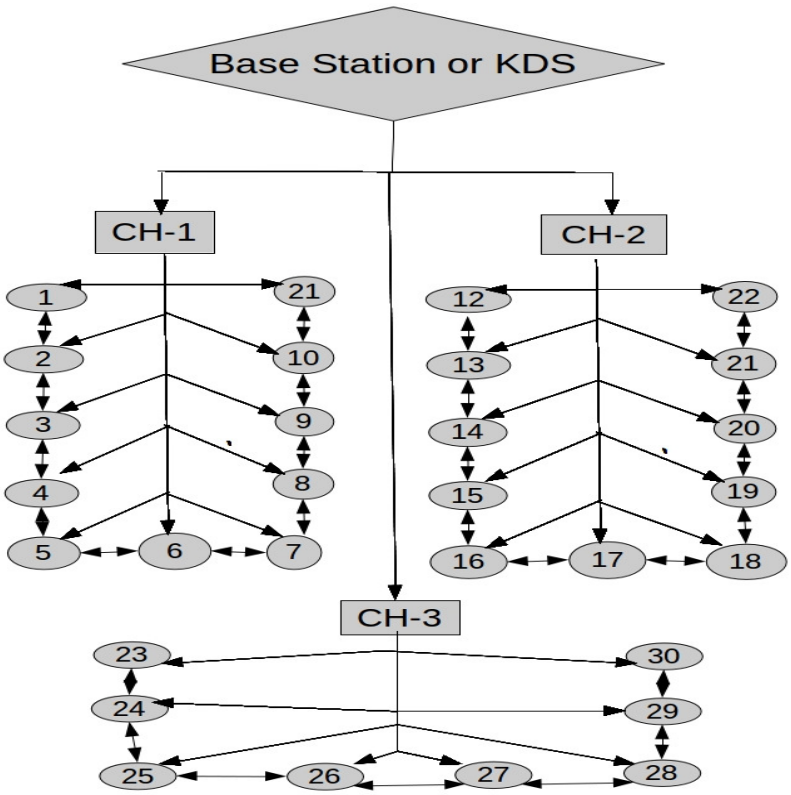

Figure 1: Network structure for $q = 11$ having 4 CHs in $2nd$ & $N = 30$ nodes in $3rd$ tier.

5 DEPLOYMENT

There can be various methods for node deployment. We discuss one of them here as an example. At the time of deployment, we shall drop the CHs along with the nodes of its cluster. Clearly instead of totally random deployment, we are deploying in small groups where exact position of nodes may still be unknown. Thus we adopt a kind of *group-wise-random* or *locally-random* deployment technique. This ensures that all the clusters are formed according to the model. However in an unlikely event of some nodes falling out of position, we adopt the following key re-scheduling technique.

Assume some node of one cluster A falls into another cluster B. In such a case, CH of cluster B broadcasts the node id or I.P. address of the misplaced node amongst all the CHs to find out the actual cluster where it should have been placed. On seeing the I.P. address or node id of this node, the CHs respond whether or not the misplaced node belongs to their cluster. Since this node was supposed to be in cluster A, its CH is the only who responds with 'YES'. Using the secure link between CH of cluster A and cluster B, the connectivity key corresponding to this sensor and CH of cluster A is transmitted to the CH of cluster B. This key is used to set up a secure connectivity link between the CH of cluster B and the misplaced. Depending on the requirements and practical hazards, CH of cluster B decides on the exact connectivity for this misplaced node in its cluster. Clearly a redistribution of connectivity keys may be required. In case this is not possible, still the node remains connected to the network but all communication will involve CH of B. It is clear that in this scenario, there is a process of node addition in cluster B and node deletion at cluster A. These processes have been described in [13] We would like to remark that instead of interconnectivity (clique connectivity) of sensor at the base level, one may desire to have just the connection with the CHs. This will enable better security, make (connectivity) key distribution easier and also reduce the importance of simple nodes at the bottommost level. In such a case the $2nd$ tier CHs may have to be powerful to ensure security.

6 COMMUNICATION KEY ESTABLISHMENT

The following protocol has to be followed during key establishment. Note that in our design the CHs are required to equate the node ids as opposed to the nodes in the original KPD scheme of Bag and Ruj [1].

- Any node i encrypts its node id N_i using the con. key that it shares with its CH and sends the encrypted node id to its CH.
- On receiving these encrypted ids, the CHs decrypts them and circulates them securely amongst themselves using the connectivity keys of one another (at CH level).
- For each incoming encrypted node ids, the CHs immediately decrypts them to get the unencrypted node ids.
- The node ids are then equated to find the common key ids of the corresponding node as described in section 2.
- Once the common key ids are obtained, they are immediately informed back to the node via the same secure channels between CHs and node.

Clearly when the nodes send their ids we utilize the connectivity model of last two tiers. Whereas when the node ids are being circulated at the CH level, we use the connectivity keys corresponding to $1st$ and $2nd$ level. Surely, if required one can make use of different cryptosystems for various clusters of $2nd$ & $3rd$ tiers and certainly for KDS-CH tier (i.e. $1st$

& $2nd$ tier) of our connectivity model.
Thus instead of the nodes, CHs get to know other nodes' id and find the lines through any two points belonging to these nodes. This helps in finding the key ids as has been described in section 2. The nodes are then securely informed about the common key by the CHs. Hence any attack on the resultant system during key establishment would require capture of some CH or somehow read the encrypted node ids. Considering both capturing CH or decrypting the encrypted node is high unlikely during key establishment, we are ensured of extremely secure key establishment of the resultant system.

7 RESILIENCY ENHANCEMENT: HASH FUNCTIONS

In this section an unique technique is presented which make the overall communication more secure. This method is particularly useful when one key of the WSN is shared by two or more nodes. In this work based on Bag & Ruj's [1] key predistribution scheme each key is shared between minimum of $\lfloor \frac{q}{4} \rfloor$ nodes to a maximum of q nodes. Since we want to maximize the distinction we look at the maximum number q.

Observe that by distinguishing communication from connectivity of a WSN, then applying a suitable cryptosystem to the connectivity model, one manages to convert the node identifier a secret or private information for each node. This information is known only to the concerned node at all times and to the CHs at the time of key establishment.

During *key establishment phase*, we use of the secret node ids of any given pair of nodes to generate a bit pattern unique to both the the nodes. When the CHs find a common shared key during key establishment, they are to generate bit patterns of length same as that of the key length of the cryptosystem being used for communication. The bit patterns must have the following properties:

- Given a bit pattern, one should not be able to compute the bit pattern of any of the node identifiers from whom it is generated.
- Any two bit patterns (amongst $\binom{N}{2}$ where $N = \frac{q^2}{4}$) should be distinct. That is no one should be able to guess one bit pattern by gaining information about another.

Next the CHs will securely send these bit pattern to concerned nodes during key establishment phase using the secure connectivity links. These bit patterns are meant to be padded or concatenated along with the corresponding key during message sending phase. Then a ``hash'' like function is to be applied to get a new set of communication keys having length same as the old cryptosystem key. One may use low cost hash function like Quark [9] for such purposes. These new keys must have the following properties:

- compute the new keys *easily* from combination of the existing communication keys and the bit pattern for any pair of nodes.
- infeasible to find any of the node ids that is used to generate a given bit pattern and hence form new keys.
- infeasible to find two different pairs of node ids generating same bit pattern and hence the new keys.

Emphasizing again, the node ids are unique to every node and the bit pattern is generated using the node ids of the two communicating nodes only. The ultimate *new key is hence unique to both the communicating parties*. The *randomness* of these *new keys* as compared to initial communication keys is half the length of the initial communication keys, which is quite desirable.

However one major storage problem arises. There is a maximum of $4q-2$ keys in a node & every key is shared among maximum of q distinct nodes. Hence, in order to have an ideal security scenario, a node may have to store $4q^2-2q=O(q^2)$ such bit pattern, which is not desirable. This prompts us to provide an alternative strategy of distributing these bit patterns so as to counter the storage issue. In the bargain we are forced to compromise on an ideal security scenario as above.

7.1 STORAGE PROBLEM: KEY ENUMERATION

To ensure minimum storage of such bit patterns while maximizing the security of the system, it is very important that all the q^2+q keys of the network has some ordering. This enumeration plays a huge role in ensuring maximum distinction among the new keys when they get generated. Since the network is partitioned into small clusters we can label the CHs & nodes and deploy accordingly.

We are primarily interested in the penultimate tier having $c=\lceil \frac{q}{4} \rceil$ CHs. We begin by labeling all of these CHs. Call them $CH_1, CH_2, \ldots, CH_c$

Next we look into the last tier where $\lfloor \frac{q^2}{4} \rfloor$ nodes are placed as described in section 4. Recalling form section 4 there are q nodes under first $\lfloor \frac{q}{4} \rfloor$ CHs and $l=\lfloor \frac{q^2}{4} \rfloor - q\lfloor \frac{q}{4} \rfloor$ nodes under the last CH. Call this number `d'. That is $d=q$ for first $\lfloor \frac{q}{4} \rfloor$ and l nodes under the last CH. Employing an obvious method of labeling, mark the nodes under the ith CH or CH_i as $id+j$ where $1 \le i \le c$ & $1 \le j \le d$. Thus nodes 1 to d or $N_1, N_2, \ldots, N_d$ are all the nodes under CH_1. Similarly, CH_2 comprises of nodes $d+1$ to $2d$ or $N_{d+1}, N_{d+2}, \ldots, N_{2d}$ and so on. With this enumeration of nodes and CHs in mind, we distribute the bit patterns as explained in section 7.2.

7.2 Distribution of Bit Patterns

Out of the distinct $(q^2+q)\binom{q}{2}$ possible new keys corresponding to q^2+q old keys in the network, one utilizes $\binom{q}{2}$ many bit patterns corresponding to a single key. This is mainly because ideally one should not assign more than $O(q)$ bit patterns per node and also the inherent symmetry of key predistribution using Affine planes.

Without loss of generality select the first key of N_1, say k_1 as the key which is shared by q nodes (the maximum value), *bit patterns* corresponding to this key are to be considered. So we use the bit patterns generated by combining any 2 among these q nodes. These distinct $\binom{q}{2}$

many patterns will be utilized by all the keys as follows.

For any other key in the network, first make a list of all nodes sharing them. Now arrange the nodes in an ascending order according their index (explained in above subsection 7.1). Thus it is clear for every key, a maximum of q nodes are arranged in ascending order of their index. Now for the communication of ith and jth corresponding to a particular key, assign the bit pattern as that of ith and jth node of k_1 and not this key.

Till now we have described a strategy how to distribute bit patterns among nodes sharing a single key. However in the current model any given pair of node shares 1 to 16 keys in common. We now describe how to use the bit patterns for two or more common keys between a pair of nodes. Our strategy generalizes quite easily. Without loss of generality, assume nodes N_x and N_y have two common keys k_s and k_t amidst others. Also let N_x be the ith node in order for k_s and ath node in order for k_t. Similarly N_y be the jth node in order for k_s and bth node in order for k_t. Then for communications between N_x and N_y using k_s, we are to use the bit pattern corresponding to ith and jth node of the key with which these patterns are generated (k_1). On the contrary if k_t is to be used then the bit pattern will correspond to ath & bth node of the chosen key (k_1). The system decides upon the key to be used and hence automatically fixes up the bit patterns by above policy. These bit patterns can then be securely distributed among the sensors using the connectivity keys shared by each node with its CH.

8 MESSAGE SENDING PROTOCOL

Suppose a message has to be sent from node N_i to node N_j for some fixed $1 \le i \ne j \le \lfloor \frac{q^2}{4} \rfloor$. Then the following protocol is to be executed.

- Among 1 to 16 existing common com. keys shared by nodes N_i & N_j one key μ_{ij} is selected.
- The appropriate bit pattern is padded with μ_{ij} and then hashed to get new communication key α_{ij}.
- N_i encrypts the message with the key α_{ij} & not μ_{ij}.
- **if** N_i and N_j share a connectivity key **then**
    - The message encrypted with com. key is again encrypted with the shared con. key
    - and send directly to node N_j.
    - N_j decrypts the outer encryption done using the con. key common to both the nodes.
- **else**
    - node N_i uses the con. key that it shares with its Cluster Head (CH) and send

the doubly encrypted message to its CH. the doubly encrypted message to its CH.

- **if** node N_j lies in the same cluster **then**
  - After decrypting with N_i's con. key and encrypting with N_j's con. key, the common CH directly send it to node N_j.
  - N_j decrypts outer encryption done using the con. key that it shares with the (common) CH giving message encrypted with $\alpha_{i,j}$.
- **else**
- The doubly encrypted message from N_i is decrypted using N_i's con. key at the
- CH of N_i.
- Re-encrypted the message encrypted with only $\alpha_{i,j}$ at CH of N_i using the con.
- key shared by CH of N_i and CH of N_j.
- Send this double encrypted message to CH of N_j.
- CH of N_j then decrypts it with the con. key shared with CH of N_i yielding message
- encrypted with $\alpha_{i,j}$.
- This message encrypted with $\alpha_{i,j}$ is re-encrypted by CH of N_j using it shared con. key with N_j & send to N_j.
- N_j will first decrypt the outer encryption done using the con. key shared with its
- own CH.
- **end if**

- **end if**
- Finally N_j uses the new communication key $\alpha_{i,j}$ shared with N_i to decrypt & read the message.

Remark 1 briefs important aspects of the combined scheme needed for analysis of the network.

Remark 1:

- Alternatively when N_i & N_j have common connectivity key, they can use only this key for message exchange instead of double encryption. So in case the communicating pair of nodes share a common connectivity, either of them has to be captured to affect their communication. Thus we are assured of total security from cryptographic view point in this case.
- The node identifiers are to be transmitted only once when key establishment takes place. This phase is very fast and secure. In later stages, when massages are exchanged, the sender encrypts it before sending and only the recipient can decrypt it completely.
- At any stage the communication keys are not known to the CH. For affecting resiliency of the network, definitely nodes have to be captured.

- Introduction of a secure connectivity model enables doubly encryption of the message while transmitting. The second encryption involves connectivity of the nodes & CHs.
- Nodes contain only the connectivity keys concerned to itself. Connectivity keys of all nodes in a cluster can only be found in CH of that particular cluster (not even in other CHs or KDS). This automatically implies to affect the communication of any node in the network, its CH must be captured.
- Though in practice capturing a CH is quite infeasible, while calculating the effect of the system on node capture, we make provision of capture of some CHs.

9 COMMUNICATION PROBABILITY AND OVERHEAD

The probability of direct communication of any given pair of nodes is defined as the communication probability of the network. Since the connectivity model is a path connected graph & communication model assures direct communication between every pair of nodes, we conclude that the **communication probability of the proposed scheme is 1**. However there has to be some trade offs in regards to communication overhead. n many extra connectivity keys have to be stored per node to ensure clique connectivity in every cluster. In the event of nodes getting overloaded, we can alternatively assign only one extra key meant for connection with its CH. It automatically implies every communication between nodes of the last leyer passes through the CHs of $2nd$ tier. So these CHs must be much powerful units to enable efficient communication. Analyzing resiliency in way similar to [13] assures significant improvements.

10 RESILIENCE

A hypothetical intrusion (i.e. attack) detection mechanism informs the KDS, CHs & subsequently the nodes about compromise of any node(s) as and when it occurs. For capture of a node X_1, connectivity keys sacrificed are its broadcast key, keys between X_1 & remaining nodes in its cluster and the exclusive key shared by X_1 & its CH.

Based on this information the concerned nodes and CH delete all the (above) connectivity keys ensuring that the captured node gets thoroughly delinked from the network. This deletion process has been elaborately described in [13, section V, subsection B]. In fact the beauty of this process is that after deletion of required connectivity links due to capture of some node(s), the other nodes in that cluster remains connected in much the same way as they would without the compromised node(s).

Remark 2:

- Noted that at any stage the communication keys are not known to the CH. Thus for affecting the resiliency of the network, some nodes have to be captured.
- Introduction of a secure connectivity model enables doubly encryption of message while transmitting. The second encryption involves connectivity of the nodes & CHs. Nodes contain only the con. keys concerned to itself. Connectivity keys of all nodes in a cluster can only be found in CH of that particular cluster (not even in other CHs or KDS). This automatically implies to affect the communication of any node in the network, its CH must be captured. Thus while calculating the effect of the system when some nodes are captured, we must make provision for capture of some CHs. In practice capturing a CH is quite infeasible.

10.1 ANALYSIS OF V(s,t) AND E(s,t)

Define $V(s,t)$ to be the proportion of nodes disconnected when s nodes of $3rd$ and t CHs

of $2nd$ tier are compromised. Now let us assume that b nodes gets disconnected when all the c CH of $2nd$ layer are captured. Thus clearly:

$$V(s,c) = \frac{b}{N-s}$$

Since t CH at $2nd$ tier are captured, only t out of c clusters should get affected. Assuming that the nodes gets disconnected evenly over the entire network, we conclude:

$$V(s,t) = \frac{bt}{(N-s)c}$$

$E(s,t)$ measures the ratio of links broken when s nodes of $3rd$ of t CHs at $2nd$ tier are compromised. Denote the initial number of links in the network by tot_links and the number of broken links case by l_{brk}. Then like in the above case for capture s nodes and all the c CHs of $2nd$ tier, we get:

$$E(s,c) = 1 - \frac{l_{brk}}{tot\_links}$$

As only t CH at $2nd$ tier are compromised & assuming the keys are uniformly distributed under the CHs, we conclude:

$$E(s,t) = \frac{t}{c}[1 - \frac{l_{brk}}{tot\_links}]$$

Note: The assumed distribution of keys under the CHs is uniform. This is not guaranteed fact. However our simulation results suggest that the assumption is reasonable.

11 SCALABILITY: ADDITION OF NODE

Connectivity model in [13] allows any number of nodes to be added in the network, whereas the communication model of Bag and Ruj [1] is not flexible in this regard. However we propose alternative tricks allowing extra nodes to come in and communicate with pre-existing nodes. In our $1st$ suggestion the $2nd$ tier CHs are required to act as trusted authorities (TAs) temporarily upon deployment of any extra node. These CHs then re-organize the clusters, distribute fresh connectivity keys to these nodes and pre-existing nodes. Thus the new node get connected to the network. These connectivity keys are to be used for communication purpose also. Though this method seems quite reasonable for practical applications, however one may look to avoid this method as online key redistribution is required here.

Alternatively if we know the number of additional nodes to be deployed, then we can pre-load the $2nd$ tier CH with that many extra con. keys. The extra nodes are to carry only one of these keys meant for connection as well as communication with the appropriate CH. Thus although clique connectivity is not achieved here but still is model is surely scalable. On top of this, if we want clique connectivity for the clusters where these extra nodes join, one has to ensure the number of extra nodes per cluster is less than q. In such a case we can also preload extra q keys per node. (Our aim is to restrict the key ring to O(q)). Under such circumstance, any incoming node should be loaded with the same (extra) keys of the the old nodes along with keys meant for the CH and other new nodes. In this section by key(s) we meant connectivity key(s) only.

12 SIMULATION RESULTS

Experimental results tabulated in Table 1 confimed our analysis of $V(s,t)$ and $E(s,t)$ discussed earlier in section 10.1. s & t denotes the assumed number of ordinary sensors and CHs captured respectively. ``BR Exp". is used as an abbreviation for Bag and Ruj's experimental

results as presented in [1]. Appreciable improvements in resiliency can be observed when our experimental (``Our Exp'') values are compared with those of Bag & Ruj [1] as is clearly visible in Table 1.

Table 1: Simulation and comparative results for $V(s,t)$ & $E(s,t)$

q	N	s	t	Our Exp. $V(s,t)$	BR Exp. $V(s,t)$	Our Exp. $E(s,t)$	BR Exp. $E(s,t)$
59	870	5	1	0.000380	0.0057	0.00458	0.068958
59	870	10	2	0.001531	0.01149	0.02094	0.157406
89	1980	11	2	0.000472	0.0055	0.00788	0.090639
89	1980	15	3	0.000979	0.00757	0.01812	0.139159
89	1980	20	4	0.001752	0.0101	0.03687	0.212303

13 CONCLUSION

First one observes that connectivity and communication can be treated as two separate aspects of a WSN. A key predistribution scheme based on affine planes and providing full node-to-node connectivity is then chosen. Now after necessary modifications to the novel secure connectivity model suggested in [13],we apply it to the chosen key predistribution scheme to obtain a highly resilient communication model providing full connectivity amongst nodes. Experimental results presented in section 12 not only confirm this fact but also exhibit the amount of improvement in resilience as compared the original key predistribution scheme proposed by Bag and Ruj in [1]

It is worth noticing that any two given pair of nodes of the resultant system can communicate between one another without their message been exposed to any other node. As has been elaborately explained in section 8, if these two nodes are in `radio frequency range' of each other (and share a connectivity key), doubly encrypted messages can be exchanged directly. In case they are not in each other's `radio frequency range' or don't have any common connectivity key, they are supposed to communicate through their CHs. However these CHs can not decrypt the encryption done with communication key shared by the nodes.

However the communication model chosen by [13] didn't provide full connectivity, hence the resultant system didn't have full connectivity. Choosing a well connected key predistribution scheme settles this issue. Other than this, they didn't indicate any particular deployment strategy. Thus how exactly the connectivity model was achieved in the target area was not clear. Section 5 has been devoted to address the deployment issue. From the discussion in section 5, it is clear that no physical movement of a node is required as long as there is some CH in its `radio frequency range' after deployment. Considering the hazards of deployment of nodes in a target area of WSN, this observation can be pretty useful to set up a network.

14 FUTURE WORK

Several future research directions stems out of our current work. Though the chosen key predistribution scheme provides direct node-to-node communication, each node has $4q-2$ to $4q+1$ where the size of the network is $\lfloor \frac{q^2}{4} \rfloor$ keys and shares 1 or 16 keys with any other node. These may prove dangerous when some nodes gets captured. Thus we must seek a scheme

having lesser keys per node having O(1) keys shared between any pair of nodes. Then one can perhaps apply the connectivity model in a suitable way to get promising results. Repeated enciphering and deciphering has been suggested at each CH in between two communicating nodes of different clusters. Certainly some communication cost will be reduced if one develops a system avoiding this. Such a key predistribution scheme has suggested by Sarkar and Chowdhury in their recently published work [14]. Even in their scheme doesn't have constant number of key shared between a pair of nodes. In this regard, it may be fascinating to see applications of other Mathematical tools.

We are also faced with the challenging problem of distributing the bit patterns in the sensors under the space constraint restriction. More precisely, our aim is to store maximum possible distinct bit patterns within a space of order $O(q)$. Combinatorial solution of this problem will be extremely fascinating.

ACKNOWLEDGEMENT

Firstly we want to express our gratitude to University Grants Commission of India for financially supporting the doctoral program of Mr. Pinaki Sarkar. This work is meant to be a part of the doctoral thesis of Mr. Pinaki Sarkar. A special thanks is due to Mr. Aritra Dhar for his sincere efforts in assisting us while converting the manuscript from latex to word. Finally we like to thank the organizers of WIMO 2011 for giving us this opportunity of extending our conference paper entitled 'Highly Resilient Communication Using Affine Planes For Key Predistribution And Reed Muller Codes For Connectivity In Wireless Sensor Network' into a journal paper. This extension has been thoroughly revised with an entire new idea being presented in **section 7.**

REFERENCES

[1] S. Bag., S. Ruj. Key Distribution in Wireless Sensor Networks using Finite Affine Plane. Accepted for publication in AINA–2011.

[2] S. A. C¸ amtepe, B. Yener, Key distribution mechanisms for wireless sensor networks: A survey 2005. Technical Report, TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[3] D. Chakrabarti, S. Maitra, and B. Roy, A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design, International Journal of Information Security, vol. 5, no. 2, pp.105–114, 2006.

[4] B. Cooke. Reed Muller Error Correcting Codes, MIT Undergraduate J. of Mathematics, 1999.

[5] L. Eschenauer and V. D. Gligor, A key-management scheme for distributed sensor networks, ACM Conference on Computer and Communications Security, pp. 41–47., 2002

[6] N. Gura., A. Patel., A. Wonder., H. Eberle., S. C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-BIT CPUs. CHES 2004. LNCS, vol 3156, pp. 119-132. Springer, Heidelberg, 2004.

[7] J. Y. Lee, D. R. Stinson, Deterministic key predistribution schemes for distributed sensor networks, Selected Areas in Cryptography, ser. Lecture Notes in Computer Science, pp. 294–307, Springer, 2004.

[8] J. Y. Lee, D. R. Stinson, A combinatorial approach to key predistribution for distributed sensornetworks. IEEE Wireless Communications and Networking Conference, WCNC 2005, NewOrleans, LA, USA, 2005.

[9] J. P Aumasson, L. Henzen, W. Meier, M. Naya-Plasencia, Quark: a lightweight hash, CHES,2010.

[10] S. Ruj and B. Roy, Key predistribution using partially balanced designs in wireless sensor networks, ISPA 2007, ser. Lecture Notes in Computer Science, Springer, Heidelberg, pp. 431–445, 2007.

[11] S. Ruj and B. Roy, Revisiting key predistribution using transversal designs for a grid-based deployment scheme, International Journal of Distributed Sensor Networks, IJDSN5(6), pp:660–674, 2009.

[12] J. G. Steiner, B. C. Neuman, and J. I. Schiller, Kerberos: An authentication service for open network systems, USENIX Winter, pp.0 191–202. 1988.

[13] P. Sarkar., A. Saha, M. U. Chowdhury. Secure Connectivity Model in Wireless Sensor Networks Using First Order Reed-Muller Codes. MASS 2010. pp. 507–512, 2010.

[14] P. Sarkar and M. U. Chowdhury. Key Predistribution Scheme Using Finite Fields And Reed Muller Codes, *SNPD 2011*, Springers Studies in Computational Science, Springer, 2011.

Authors

Pinaki Sarkar* is currently pursuing his Ph.D. from Jadavpur University in collaboration with cryptology group of ISI, Kolkata. Earlier he graduated with Mathematics honours from St. Xaviers' College, Kolkata and did his masters with Mathematics from CMI, Chennai. His subjects of interests are Algebra, Number Theory, Coding Theory, Combinatory, Cryptology and Wireless Sensor Network Security.

Amrita Saha is currently pursuing MTech in Computer Science from Indian Institute of Technology, Bombay. She had earlier done her Bachelors in Engineering on Information Technology from Jadavpur University, Kolkata. Her research interests are network security in Wireless Sensor Networks and Machine Learning.

Samiran Bag received a B.Tech degree in Computer Science and Engineering from West Bengal University of Technology, India. Then after completing his M.Tech in Computer Science from Indian Statistical Institute, Kolkata he is currently continuing to be at ISI, Kolkata in pursuit of Ph.D. degree in Computer Science.