SECURITY ENHANCED KEY PREDISTRIBUTION SCHEME USING TRANSVERSAL DESIGNS AND REED MULLER CODES FOR 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

Resource constraints of the nodes make security protocols difficult to implement. Thus key management is an important area of research in Wireless Sensor Networks (WSN). Key predistribution (kpd) which involves preloading keys in sensor nodes, has been considered as the best solution for key management when sensor nodes are battery powered and have to work unattended. This paper proposes a method to fix some loophole in an existing key predistribution scheme thereby enhancing the security of messages exchanged within a WSN. Here we use a model based on Reed Muller Codes to establish connectivity keys between sensor nodes. The model is then utilized to securely establish communication keys and exchange messages in a WSN designed on basis of two schemes using transversal designs for key predistribution. The combination of the key predistribution scheme and the connectivity model gives rise to highly resilient communication model with same connectivity between nodes as the chosen key predistribution scheme.

KEYWORDS

Connectivity, Communication, Reed-Muller Codes, Transversal Designs, Security.

1. INTRODUCTION

Wireless Sensor Network (WSN) consists of many tiny sensor nodes having very limited amount of storage, insufficient battery power, low computational power. They are scattered randomly or deterministically over a large target area. These sensors communicate between each other via radio frquency waves. These nodes gather sensitive information and they have widespread application in several civil and military purposes. These purposes include military surveilance, ocean-water monitoring, wild fire detection, temperature monitoring etc. to name a few. Since these sensors deal with very sensitive information, they must communicate securely so that no adversary can get hold of the information sent by them. To achieve this, cryptographic primitives have to be used for communication between sensors. Inevitably, this gives rise to usage of cryptographic keys.

1.1. Related Works and our contributions

Cryptographic keys can be established between two parties in many ways. The conventional

DOI : 10.5121/ijnsa.2011.3609

way using protocols like Kerberos [13] is expensive for sensor networks, which are resource constrained. The other method using public keys is being explored [5, 9] but not preferred because of costly operations involved. Key predistribution (kpd) is a method to preload cryptographic keys in sensor nodes, even before their deployment in the area of operation. It is a symmetric key approach, where two communicating nodes share a common secret key. The sender encrypts the message using the secret key and the receiver decrypts using the same key. Several key predistribution schemes that can be found in [3,6-8,4,11]

In this paper we propose a connectivity model based on Reed Muller Codes that we use to enhance the security of two existing key predistribution scheme proposed by Lee Stinson scheme [6-8]. This combination of the two schemes give rise to a secure communication model for Wireless sensor networks.

2 COMMUNICATION MODEL

Here, we shall be using two communication models proposed by Lee & Stinson. In both papers the authors used transversal design for key predistribution in Wireless sensor networks. A transversal design $TD(k,\lambda;n)$ is a triple (X,G,A) with the following properties:

1. X is a set of kn number of elements called varieties.
2. $G = G_1 \cup G_2 \cup \ldots \cup G_n$ is a partition of X where $|G_i| = p, \forall i \in \{1,2,\ldots,n\}$.
3. $A = \{B_1, B_2, \ldots, B_b\}$ where $|B_j| = n, \forall j \in \{1,2,\ldots,b\}$ and $|B_j \cap G_i| = 1, \forall j \in \{1,2,\ldots,b\}, \forall i \in \{1,2,\ldots,n\}$.
4. For any $x \in G_i, y \in G_j, i \neq j$, $|\{r : x, y \in B_r, 1 \leq r \leq b\}| = 1$.

The authors mapped the same design to key predistribution scheme in Wireless sensor networks. Their key predistribution schemes are described below.

2.1 Design 1

The details of the first design proposed by Lee Stionson can be found in [7,8]. A brief outline is presented here.

Let p be a prime number and k be an integer such that $2 \leq k \leq p$.

There are p^2 number of nodes in the network. These nodes are given by:

$$\begin{matrix} N_{0,0} & N_{0,1} & N_{0,2} & \ldots & N_{0,p-1} \\ N_{1,0} & N_{1,0} & N_{1,2} & \ldots & N_{1,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{p-1,0} & N_{p-1,1} & N_{p-1,2} & \ldots & N_{p-1,p-1} \end{matrix}$$

Let, $X = \{(x,y) : 0 \leq x \leq k-1, 0 \leq y \leq p-1\}$ is the set of varities.

$G_i = \{(i,y) : 0 \leq y \leq p-1\} \forall i \in \{0,1,\ldots,k-1\}$ are the groups of the design.

$A_{a,b} = \{(x, ax+b) : 0 \leq x \leq k-1\}$ where $0 \leq a,b \leq p-1$ are the block where all operations are done under modulo p.

Now , if each variety is mapped to a unique key and each block made to correspond to a node, then this will give rise to a key predistribution scheme. This is the key predistribution scheme of Lee Stinson. Here the set of keys is given by: $\mathsf{K} = \{(x,y) : 0 \leq x \leq k-1, 0 \leq y \leq p-1\}$.

The keys belonging to node $N_{a,b}$ is $K_{a,b} = \{(x, ax+b \bmod p) : 0 \leq x \leq k-1\}$ Now two

nodes $N_{a,b}$ and $N_{a',b'}$ will have a common key if $K_{a,b} \cap K_{a',b'} \neq \phi$. Such a key will exist if $ax+b = a'x+b'$ has a solution under division modulo p or if $x = (b'-b)(a-a')^{-1}$ exists and lies in between 0 and $k-1$. We shall get a solution for x if $a \neq a'$

2.2 Design 2

The second chosen kpd scheme was proposed in details by Lee Stionson in [6,8]. Briefly recalling:

Again let p be a prime number and k be an integer such that $2 \leq k \leq p$.

There are p^3 number of nodes in the network. These nodes are given by

$$\begin{matrix} N_{0,0,0} & N_{0,0,1} & N_{0,0,2} & \cdots & N_{0,0,p-1} \\ N_{0,1,0} & N_{0,1,0} & N_{0,1,2} & \cdots & N_{0,1,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{0,p-1,0} & N_{0,p-1,1} & N_{0,p-1,2} & \cdots & N_{0,p-1,p-1} \end{matrix}$$

$$\begin{matrix} N_{1,0,0} & N_{1,0,1} & N_{1,0,2} & \cdots & N_{1,0,p-1} \\ N_{1,1,0} & N_{1,1,0} & N_{1,1,2} & \cdots & N_{1,1,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{1,p-1,0} & N_{1,p-1,1} & N_{1,p-1,2} & \cdots & N_{1,p-1,p-1} \end{matrix}$$

$$\begin{matrix} \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix}$$

$$\begin{matrix} N_{p-1,0,0} & N_{p-1,0,1} & N_{p-1,0,2} & \cdots & N_{p-1,0,p-1} \\ N_{p-1,1,0} & N_{p-1,1,0} & N_{p-1,1,2} & \cdots & N_{p-1,1,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{p-1,p-1,0} & N_{p-1,p-1,1} & N_{p-1,p-1,2} & \cdots & N_{p-1,p-1,p-1} \end{matrix}$$

Let, $X = \{(x,y) : 0 \leq x \leq k-1, 0 \leq y \leq p-1\}$ be the set of varities.

$G_i = \{(i,y) : 0 \leq y \leq p-1\} \forall i \in \{0,1,\ldots,k-1\}$ are the groups of the design.

$A_{a,b,c} = \{(x, ax^2+bx+c) : 0 \leq x \leq k-1\}$ where $0 \leq a,b,c \leq p-1$ are the block where all operations are done under modulo p.

Now, if each variety is mapped to a unique key and each block made to correspond to a node, then this will give rise to a key predistribution scheme. This is the $1st$ key predistribution

scheme of Lee and Stinson. Here the set of keys is given by: $\mathsf{K} = \{(x, y) : 0 \le x \le k-1, 0 \le y \le p-1\}$ The keys belonging to node: $N_{a,b,c}$ is $K_{a,b,c} = \{(x, (ax^2 + bx + c) \bmod p) : 0 \le x \le k-1, k \le p\}$. Two nodes $N_{a,b,c}$ and $N_{a',b',c'}$ will have a common key if $K_{a,b,c} \cap K_{a',b',c'} \neq \phi$. Such a key will exist if $ax^2 + bx + c = a'x^2 + b'x + c'$ has a solution under division modulo p or if $x = \{-(b-b') \pm \sqrt{(b-b')^2 - 4(a-a')(c-c')}\}/(2(a-a'))^{-1}$ exists and lies in between 0 and $k-1$. We shall get a solution for x if $(b-b')^2 - 4(a-a')(c-c')$ is a quadratic residue modulo p.

Here we want to remark that both Design 1 and Design 2 generalizes to any finite field F_q by replacing p by any prime power q in the above argument.

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 $N_{i,j}$ and node $N_{i',j'}$ want to communicate securely. If they do share a key then it will have the id (x, y) where $x = (j'-j)(i-i')^{-1}, y = ix + j = i'x + j'$. For finding this key both the nodes must exchange their node-ids. 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 key((x, y)) between the nodes in a manner similar to the nodes. Then she can find the node id of a node containing the key with id (x, y) in the following manner:

Let the id of the node be (r, s). If this node contains the key (x, y) then $y = rx + s$ or, $s = y - rx$. By fixing an r she can compute s thus finding the node id of a third node containing the shared key between node (i, j) and (i', j'). Thus enabling selective node attack. She can capture node (r, s) and get to know the actual key with id (x, y).

Similar attack can be done on design 2. Here, the common key between two nodes $N_{i,j,k}$ and $N_{i',j',k'}$ is given by (x, y), where

$$x = \{-(j-j') \pm \sqrt{(j-j')^2 - 4(i-i')(k-k')}\}/(2(i-i'))^{-1}$$

and $y = ix^2 + jx + k = i'^2 + j'x + k'$. Now, the adversary can find the id of a node (say $N_{a,b,c}$) containing the key (x, y) like the following;

$ax^2 + bx + c = y$

or, $c = y - ax^2 - bx$

by fixing a and b, the adversary will be able to compute c.

To counter this problem, we first differentiate the two aspects communication and connectivity of a WSN. Then like in [12], apply Reed Muller Codes to suitably model the connectivity aspect. The construction of the model is presented the in 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 same connectivity amongst nodes as the initial models with virtually same communication overhead.

4 PROPOSED CONNECTIVITY MODEL

As stated above, Reed Muller codes will be utilized to structure the connectivity aspect of the WSN. These codes have been elaborately described in [2] and necessary notational changes have been highlighted by Sarkar et al. in [12, section IV]. We follow similar procedure as described in [12, section IV] baring some modification to be illustrated now.

Both the models will always have three tiers with the ``Base Station'' or ``KDS'' in the $1st$ or topmost tier. The second tier will consist of p & p^2 newly introduced cluster heads (CHs) for the first and second designs respectively. Each of these CHs will be assigned p many nodes in the $3rd$ tier in both the designs. Thus for `Design 1' we introduce $p$ many new CHs in the $2nd$ tier each having $p$ `ordinary nodes' under it. Whereas for `Design 2' we allocate $p^2$ many CHs in the $2nd$ tier each having $p$ `ordinary nodes' under it. This ensures key storage for each CH is same (= O(p)) for both designs.

It is evident that current connectivity model is heterogeneous in nature having different number of nodes in various clusters. Other than this exactly here 3 tiers are required for connectivity model. These facts distinguishes present designs from the original design of Sarkar et al. [12, section IV].

Clusters between various tiers of the connectivity model are designed using first order Reed Muller codes. Connectivity of $1st$ & $2nd$ levels of `Design 1' is given by a $p$ complete graph. Whereas connectivity pattern of $1st$ & $2nd$ levels of `Design 2' is a p^2 complete graph

Consider $\mathsf{Z}_2[x_1, x_2, \ldots, x_m]$ where $m = p$ or p^2 for `Design 1' and `Design 2' respectively. Like in [12], 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 $1 \le i \le m$ where m is mentioned above. Sample connectivity pattern for a cluster containing KDS & 3 CHs (meant for `Design 1') and another pattern with KDS & $4 = 2^2$ CHs (meant for `Design 2') are presented in the following matrix below:

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

$$\begin{bmatrix} \mathbf{KDS} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \mathbf{CH_1} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{CH_2} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{CH_3} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{CH_4} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

Matrices like the above one are used for construction of Reed Muller codes. In particular the first matrix (meant for `Design 1') has been referred to as $R(1;3)$ in [2]. 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 first 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 or CHs or ordinary 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 $m = p$ variables (for both designs) in the matrix. Each node is assigned a row. Thus we look at $Z_2[x_1, x_2, \ldots, x_p]$ as was similarly done in [12, section IV, subsection B] Connectivity matrix for a cluster having 1 CH & 3 nodes and 1 CH & 4 nodes for the respective designs are as follows:

$$\begin{bmatrix} \mathbf{CH} & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ \mathbf{N_1} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{N_2} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{N_3} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \mathbf{1} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ \mathbf{x_1} & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{x_2} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{x_3} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{x_4} & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

The construction of second matrices of respective designs from first can be found in [12, Section IV, Subsection B]. There is a broadcast channel and a provision for special link meant only for communication of the CH with KDS. CH need not be present in the inter-nodal links. Here also 1 means presence of connectivity link & 0 -its absence.

Figure 1 give an lively example of 9 nodes under $p = 3$ CHs constructed using `Design 1'. While in Figure 2, a small network example with 8 nodes under $p^2 = 2^2 = 4$ CHs have been constructed using `Design 2'. As was earlier stated both the models have 3 tier with KDS in topmost, CHs in $2nd$ & nodes in $3rd$. The line joining CH-1 and node 2, CH-2 and node 5, CH-3 and node 8 are bent to symbolize they do not interfere with other links.

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 *pseudo-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.

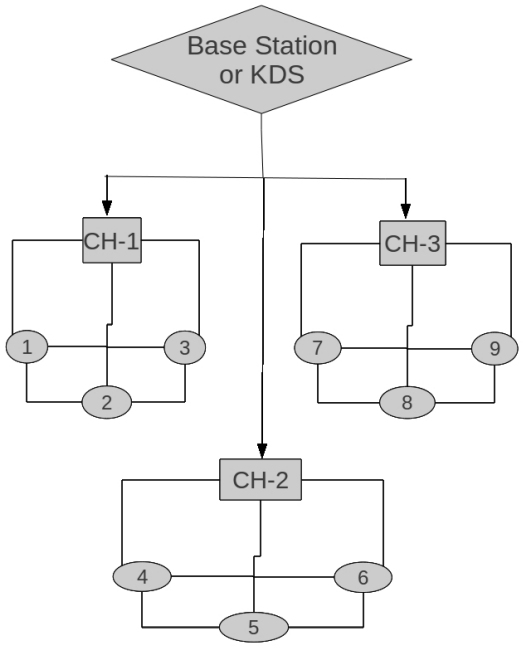

Figure 1: Network structure for $p = 3$ has $p = 3$ CHs in $2nd$ & $N = p^2 = 9$ nodes in $3rd$ tier using Design 1.

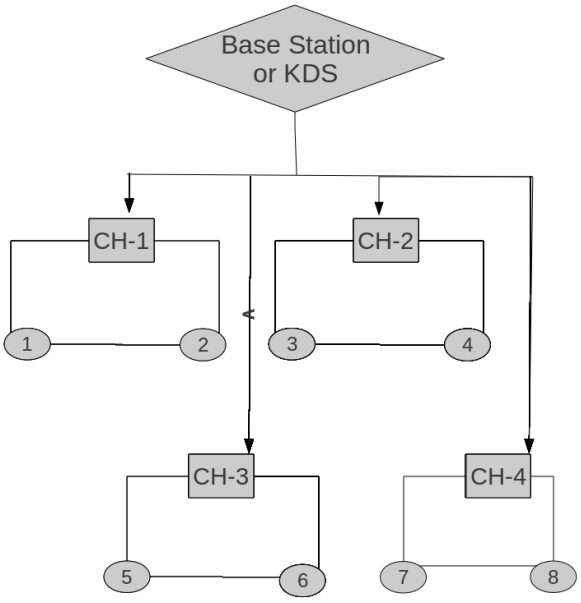

Figure 2: Network structure for $p = 2$ has $p^2 = 4$ CHs in $2nd$ & $N = p^3 = 8$ nodes in $3rd$ tier using Design 2.

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 [12] 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

We now describe how one can utilize the secure connectivity model for communication key establishment. As mentioned earlier node ids can be used for this purpose.

Any node encrypts its node id $N_{i,j}$ 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.
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 equating the resulting linear equations. Then the nodes are 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 more than one nodes. In this work based on Lee & Stinson's [7,6,8] key predistribution schemes. In Design 1 2.1 based on the scheme in [7] each key is shared between p nodes. On the other hand in Design 2 2.2 based on the scheme in [7] each key is shared between p^2 nodes. The method suggested here will work fine for Design 1 and hence we describe it in detail for Design 1 here. However for Design 2 storage may become a factor as shall be shortly explained.

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 $\binom{p^2}{2}$) should be distinct (Design 1). 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 [10] 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 are k keys in a node for $2 \le k \le p$. On top of that every key is shared among maximum of p distinct nodes (Design 1). Hence, in order to have an ideal security scenario, a node may have to store $kp = O(p^2)$ (max) 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.

Clearly, it is evident for Design 2, each node may end up dealing with $O(p^3)$ many bit pattern. This case is even more harder to handle.

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 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 p CHs. We begin by labeling all of these CHs. Call them $CH_1, CH_2, \ldots, CH_p$

Next we look into the last tier where p^2 nodes are placed, p under each CH, as described in section 4. Employing an obvious method of labeling, mark the nodes under the ith CH or CH_i as $id + j$ where $1 \le i, j \le p$. Thus nodes 1 to d or $N_1, N_2, \ldots, N_p$ are all the nodes under CH_1. Similarly, CH_2 comprises of nodes $p+1$ to $2p$ or $N_{p+1}, N_{p+2}, \ldots, N_{2p}$ 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 $kp\binom{p}{2}$ possible new keys corresponding to kp old keys in the network, one utilizes $\binom{p}{2}$ many bit patterns corresponding to a single key. This is mainly because ideally one should not assign more than $O(p)$ 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 p nodes, *bit patterns* corresponding to this key are to be considered. So we use the bit patterns generated by combining any 2 among these p nodes. These distinct $\binom{p}{2}$ 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 p 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.

***Remark 1**:*

- Clearly for Design 2, the storage of bit pattern per node by the above described strategy will be $O(p^2)$. In fact it is pretty evident that if we want all distinct bit patterns per key, p^2 such are required. Thus the above described strategy will not work for Design 2.
- We can replace p by any prime power q in the above argument. This takes care of generalization to any finite field F_q.

8 MESSAGE SENDING PROTOCOL

Suppose a message has to be sent from node $N_{i;j}$ to node $N_{i';j'}$ for some fixed $0 \le i, j, i', j' \le p-1$. Then the following protocol is to be executed.

Among existing common communication keys shared by nodes $N_{i,j}$ & $N_{i',j'}$ one key μ is selected.
The appropriate bit pattern is padded with μ and then hashed to get new communication key α.
N_i encrypts the message with the key α and not μ.
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.
if node N_j lies in the same cluster **then**
After decrypting with $N_{i,j}$'s con. key and encrypting with $N_{i',j'}$'s con. key, the common CH directly send it to node $N_{i',j'}$.

$N_{i',j'}$ decrypts outer encryption done using the con. key that it shares with the (common) CH giving message encrypted with α.

else

The doubly encrypted message from $N_{i,j}$ is decrypted using $N_{i,j}$'s con. key at the CH of $N_{i,j}$.

Re-encrypted the message encrypted with only α at CH of N_i using the con. key shared by CH of $N_{i,j}$.and CH of $N_{i',j'}$

Send this double encrypted message to CH of $N_{i',j'}$.

CH of N_j then decrypts it with the con. key shared with CH of $N_{i,j}$.yielding message encrypted with α.

This message encrypted with α is re-encrypted by CH of $N_{i',j'}$ using its shared con. key with $N_{i',j'}$ & send to $N_{i',j'}$.

N_j will first decrypt the outer encryption done using the con. key shared with its own CH.

end if

end if

Finally $N_{i',j'}$ uses the new communication key α. shared with $N_{i,j}$ to decrypt & read the message.

Remark 2 briefs important aspects of the combined scheme needed for analysis of network parameters.

Remark 2:

- Alternatively when $N_{i,j}$ & $N_{i',j'}$ 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 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 [12, 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), other nodes in that cluster remains connected in much the same way as they would without the compromised node(s).

***Remark 3**:*

- It should be noted that at any stage the communication keys are not known to the CH. Thus for affecting the resiliency of the network, definitely 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 ensure some CHs are also captured. In practice capturing a CH is quite infeasible.

10 EXPERIMENTAL RESULTS

Experimental results have been tabulated in Table 1. N and k are as defined in section 2. In the table, ``Exp." stands for experimental, ``Thry." means theoretical results for current scheme. ``LS Exp" is used as an abbreviation for Lee Stinson's experimental results as presented in [7] corresponding to `Design 1'. The tabulated values compares our results with [7].

Table 1: Simulation & comparative results for $E(s,t)$ for Design 1 with $p = 47$, hence $N = p^2$ where s nodes & t CHs are captured.

k	N	s	t	Our Exp. $E(s,t)$	LS. Exp. $E(s,t)$
30	2209	2	1	0.00851	0.4000
30	2209	4	2	0.01901	0.4469
30	2209	6	3	0.02852	0.4469
30	2209	8	3	0.02992	0.4689
30	2209	10	4	0.04171	0.4901

11 CONCLUSION

A secure connectivity model has been utilized to make key establishment secure and then enhance message exchange of two pre-existing key predistribution schemes. Both the scheme were designed by Lee and Stinson in their works [7] and [8]. Both these scheme are based on Transversal designs meant to support p and p^2 nodes respectively for a prime p. Significance of choosing a prime p is that the authors of [7] and [8] focused on the finite field Z_p. We have pointed out in section 2 how one can extend their idea to a general finite field F_q .

While designing our connectivity model, we have used a novel technique introduced by Sarkar, Saha and Chowdhury [12]. Like them, we have also utilized $1st$ order Reed Muller Codes for

generating the connectivity patterns in each cluster. However ours is an heterogeneous model as compared to homogeneous model of theirs as has been explained in 4 . Another point of distinction specially for `Design 2' is that we have exactly 3 tiers with unequal nodes/CHs in various clusters.

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. To the best of our knowledge proposing a secure connectivity model, then using it for secure establish and later for enhancing the security during message exchange was first proposed by [12].

Experimental results presented in section 10 exhibit the amount of improvement in resilience as compared the original key predistribution scheme proposed by Lee and Stinson. Though Sarkar et al. provided theoretical bounds of resiliency, experimental results were not mentioned. 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.

12 FUTURE WORK

Several future research directions stems out of our current work. The chosen key predistribution scheme does not guarantee direct node-to-node communication. Thus even though the connectivity is path connected graph, the resultant system does not have full connectivity.

The number of keys vary from 2 to $p^r - 1$. For better connectivity, we need the number of keys to be closer to p^r. This number is rather high and prove dangerous when a node is captured. Thus we must seek a scheme having lesser keys per node with O(1) keys shared between any pair of nodes. Then one can 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. One such key predistribution scheme has suggested by Sarkar and Chowdhury in their recently published work [15]. 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

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 CNSA 2011 for giving us this opportunity of extending our conference paper entitled 'Highly Resilient Key Predistribution Scheme Using Transversal Designs And Reed Muller Codes For Wireless Sensor Network' into a journal paper. This extension has been thoroughly revised with an entire new idea being presented in **section 7 and all its subsections**.

REFERENCES

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

[2] B. Cooke, Reed Muller Error Correcting Codes, *MIT Undergraduate Jour. of Mathematics*, 1999.

[3] S. A. C¸ amtepe. B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks. *ESORICS*, ser. Lecture Notes in Computer Science, vol. 3193. Springer, 2004, pp. 293–308.

[4] D. Chakrabarti. S. Maitra. B.Roy, A key pre-distribution scheme for wireless sensor networks: Merging blocks in combinatorial design. *ISC*, ser. Lecture Notes in Computer Science, vol. 3650. Springer, 2005, pp. 89–103.

[5] 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.

[6] J. Lee. D. R. Stinson, Deterministic Key Predistribution Schemes for Distributed Sensor Networks *SAC 2004 Proceedings.* Lecture Notes in Computer Science, vol. 3357, pp. 294307, Springer, 2005.

[7] J. Lee. D. R. Stinson, A combinatorial approach to key predistribution for distributed sensor networks *IEEE Wireless Communications and Networking Conference, WCNC 2005, New Orleans, LA, USA*, Vol. 2. IEEE Communications Society, pp.1200–1205 2005.

[8] J. Lee. D. R. Stinson, On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, 2008.

[9] D. J. Malan. M. Welsh. M. D. Smith, Implementing public-key infrastructure for sensor networks. *TOSN*, vol. 4, 2008.

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

[11] S. Ruj. B.Roy. Key predistribution using partially balanced designs in wireless sensor networks. *ISPA 2007.* LNCS, vol. 4742, pp. 431–445. Springer, Heidelberg, 2007.

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

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

[14] D. Xu. J. Huang., J. Dwoskin. M. Chiang. R. Lee, Re-examining probabilistic versus deterministic key management, *Proceedings of the 2007 IEEE International Symposium on Information Theory(ISIT)*, pp. 2586–2590, 2007.

[15] 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.