Secure Connectivity model in Wireless Sensor Networks (WSN) using 1st order Reed-Muller codes

Pinaki Sarkar
Department of Mathematics
Jadavpur University,
Kolkata, India.
Email: pinakisark@gmail.com

Amrita Saha
Department of Information Technology
Jadavpur University
Kolkata, India
Email: amrita.saha87@gmail.com

Morshed Udan Chowdhury
School of Information Technology
Deakin University-Melbourne Campus
Melbourne, Australia
Email: morshed.chowdhury@deakin.edu.au

***Abstract*—In this paper, we suggest the idea of separately treating the connectivity and communication model of a Wireless Sensor Network(WSN). We then propose a novel connectivity model for a WSN using first order Reed-Muller Codes. While the model has a hierarchical structure, we have shown it works equally well for Distributed WSN. Though one can use any communication model, we prefer to use communication model suggested by Ruj and Roy [1] for all computations and results in our work. One might use two suitable secure (symmetric) cryptosystems on the two different models viz. connectivity and communication. By doing so we have shown how resiliency and scalability are appreciably improved as compared to Ruj and Roy [1].**

***Keywords*-Connectivity, Communication, Reed-Muller Codes, Reed-Solomon Codes, Security.**

I. Introduction

Two of the most popular ad hoc networks are Mobile Ad Hoc Network (MANET) and Sensor networks. Here we will deal with Wireless Sensor Networks (WSN), where the sensors communicate among themselves using radio frequencies.

WSN has several military applications like collection of information about enemy movements, explosions, detecting and characterizing chemical, biological, radiological, nuclear and explosive (CBRNE) materials. On the other hand, detecting and monitoring of environmental changes in plains, forest etc, to monitor vehicle traffic on highways or in congested parts of a city etc are some of the important civilian uses of WSN.

A WSN consists of a large number of sensor nodes with limited power, computational, storage and communicational capabilities which are generally deployed in a very dense fashion. A sensor node typically consists of a power unit, a storage unit and a wireless transceiver. Other than the sensors being resource constrained, the WSN is faced with some more disadvantages. For example, the sensor nodes can fail or be easily captured and the network topology may keep on changing.

These facts make secure communication difficult and hence scope for more study to increase the security of communication in a WSN. Constrained resources of the sensors implies that we will have to use private key cryptosystem instead of public key cryptosystem. Inability to use a secure channel for key distribution during communication implies that keys need to be pre-distributed (i.e. keys are uploaded in the nodes prior to deployment) & key establishment has to be done prior to sending any message. One such key establishment method has been suggested by Ruj and Roy [1] using Reed-Solomon codes. In this paper we have used first order Reed-Muller Codes to design a hierarchical connectivity model of a WSN.

A. Related Works

A random key pre-distribution scheme was proposed by Eschemauer and Gligor [2], in which keys are drawn randomly from a key pool and placed in sensors prior to deployment. [3] provides an extended survey of different pre key-distribution schemes for Distributed Sensor Networks.

Two deployed nodes, which are willing to communicate between themselves, must find out a common key and this phase is termed as shared key discovery. In case of absence of any common key, a path has to be followed to establish a common key between those nodes. This phase is referred to as path key establishment. Lee and Stinson [4] and Ruj and Roy [5] showed that deterministic designs have the advantage of efficient shared key discovery and path key establishment.

A code based key management system was proposed by Al-Shurman and Yoo [6], where matrices have been used along with a random vector to generate a codeword which is the secret key chain. With certain probabilities the design satisfies Cover-free-family (CFF). However, neither the problem of connectivity in a network nor the scenario of node compromise is addressed.

In [1] Roy and Ruj the node identifiers are transmitted through wireless channels. Corresponding to each node identifier there is a unique polynomial. When two nodes try to find their common key identifiers and hence their common keys, they equate their respective polynomials. Here we note that an attacker may try to compute a (captured) node's polynomial and equate it to some other node's polynomial to get the common key identifiers, in case there is any. Also by directly solving the polynomials, key identifiers are revealed enabling the attacker to decide precisely which of the nodes should be attack to get a particular key needed for decrypting a particular message. Thus selective attack is actually possible from key establishment stage onwards.

B. Our Contributions

We try to address the above problem, by differentiating between connectivity and communication of a network. Then deployment of the sensor nodes can be achieved according to our connectivity model in a controlled WSN and one can use this knowledge of deployment to establish the connectivity model in uncontrolled WSN.

We then use a security protocol in our connectivity model which encrypts the node identifier that is broadcast, ensuring that only the intended recipients are able to decrypt it. Here the advantage is that though in case of node capture, all the stored keys and key identifiers corresponding to them are exposed, but in order to find out which of the uncompromised nodes share these keys, one has to successfully decrypt the node identifier.

We shall show how we can modify our model to fit into a Distributed Wireless Sensor Network (DWSN). Only requirement is that we have to plug in $O(\frac{N}{t})$ many nodes (N and t are defined in the next section) which will act as Cluster Heads (CH) and these nodes need to be made more secure. With this assumption we shall show that our system provides much improved security as compared to [1].

The connectivity model presented here uses a path connected graph. For communication (direct or indirect) between two nodes, a path between them is required but not vice versa.

II. Basic Concepts, Definitions and Notations

- Radius of communication: This is the maximum distance, r, of transmitting and receiving radio frequencies for each node. Different nodes might have different radii of communication. For instance, in an hierarchical WSN, the sensor nodes may have lesser value of r as compared to CHs.
- Communication and Connectivity keys: Our model has two aspects, viz. communication and connectivity, for which we use two different cryptosystems and hence two different sets of keys. Communication keys are pre-distributed and our target is to make key establishment secure. To achieve this, we make use of connectivity keys. Here since we are dealing with much less nodes in each cluster of a hierarchical system, one can employ group key distribution or group key agreement schemes. Such schemes have been described in page 3 of [7]
- Connectivity: Two nodes or CHs or a node and a CH are said to be connected if they are in the communication radius of each other and share at least one secure connectivity key. Please note that we are differentiating between security keys of connection and communication.
- Connectivity model of a network: Connectivity model of a network is a graphical representation depicting how the nodes are connected to each other in terms of transmitting / receiving radio frequencies, infrared & optical frequencies etc. A matrix design will be discussed in detail in section II, subsections $C-F$.
- Node Communication: Two nodes can communicate if there exist a path between them in the graphical representation of the connectivity model and they share at least one communication key.
- Notations Used: N - total number of nodes.
  N' - total number of sensor nodes + Cluster Heads.
  n - number of keys in the key ring of each node.
  t - number of nodes per cluster in the connectivity model.
  $\lceil a \rceil$ - denotes least integer $\geq$ the given real number a.
  Reed-Muller(r, q_{RM}): r - degree of the Reed-Muller code & q_{RM} - a prime power.
  $F_{q_{RM}}$ - field over which Reed-Muller vectors are defined.
  Reed-Solomon(k, q_{RS}): k - degree of the Reed-Muller code & q_{RS} - a prime power.
  Other than this we will use some notion related to resiliency of our model. They will be defined as and when required.

III. Models of the system

Our hierarchical system has two aspects viz. connectivity and communication.

A. Connectivity model:

In this paper we concentrate on the connectivity aspect for which we use 1st order Reed-Muller codes. The details will be explained in the sections IV.

B. Communication model:

While calculating the resiliency we need to use some communication model. For this purpose we have chosen the model suggested by Ruj & Roy [1] based on Reed-Solomon codes for key pre-distribution. Here we note that one can choose any existing model of communication. Indeed we have been able to obtain theoretically better results in terms of security and resiliency. Our model is also scalable.

Also in Ruj and Roy [1] two communicating nodes had to be in the radius of communication of each other and share at least one secret key. However in our case all that is required is for them to have at least one communication key in common. In case they are not in radius of communication of each other multi-hop communication with the help of cluster heads may be accomplished to obtain a secure communication. Since the CHs are much powerful units and only $O(\frac{N}{t})$ (see Theorem 4, Section VI) of (extra) CHs are required, the communication overhead remains unaffected.

IV. Connectivity Model

We would like to refer to [9] for an elaborate description of 1st order Reed-Muller codes. Baring a few minor notational changes we shall use them as described in [9] to develop our connectivity model. The changes required for us are as follows:

The variable x_i in our model is same as the variable x_{m-i} [9], where m is the number of variables. Here the vector associated with our monomial x_i has 2^{i-1} ones, followed by 2^{i-1} zeros and so on, where $1 \leq i \leq m$. For example, in a space of size 2^2,i.e. with $m = 2$ the vector associated with x_2 is (1100). Again in a space with $m = 3$, the vector associated with the monomial $x_3x_2x_1$ can be found by multiplying $(11110000)*(11001100)*(10101010)$ which gives (10000000). Addition of x_3 & x_2 yields (00111100).

In our connectivity model, we use matrices of the form:

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

as our connectivity matrix. The above has been denoted as $R(1;3)$ in [9]. Here 1 means degree of monomial is 1 and 3 means number of variable is 3.

Each node in the cluster including the cluster head is assigned a vector corresponding to a variable. In each vector of the given vector space defined over F_2, a 1 implies connectivity link is present and 0 implies no connection is present. Thus if two nodes have a 1 in the ith position, then they are connected by a connectivity link, i.e., they share a common frequency channel that can be made secure by use of secure connectivity keys.

1 is assigned to the Cluster Head and x_1, x_2 and x_3 to the nodes under that cluster head.

The 1st column has all 1s for all the nodes, this provides a broadcast channel for that cluster. This can be used for Traitor-tracing or for key distribution or pre-distribution when a node or cluster head is captured, as will be explained later.

A. Hierarchy based Model

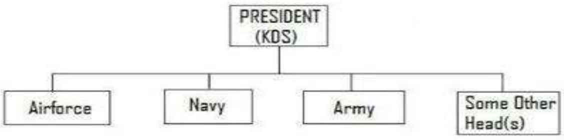


Figure 1. A typical Hierarchical system with one KDS and four nodes

In this section we present the most generalized form of our model which is meant for hierarchy based wireless sensor network. In the following diagram President(P) acts as the group head or cluster head (CH) which we often call as KDS (Key Distribution Server). Army(A), Navy(N) and Airforce(F) are three nodes in this group or cluster. We also make a provision for some other Head(s) of general public body(G) to be brought later on into this cluster (see fig.1). For such a model we use following connectivity matrix. Here $a - p$ are various connectivity channels which may or may not use same radio frequency but surely uses different connectivity keys.

$$\left[\begin{array}{ccccccccccccccccc} & a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p \\ \mathbf{P} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \mathbf{A} & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{N} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{F} & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{G} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

All the three nodes are connected to each other in various possible ways. KDS or President is present in all connection which is desirable as he is the supreme authority. It should be noted that any node can communicate with the President via an exclusive channel shared between the two and no other. Clearly there is a channel shared between any two of the nodes and the president, a channel for three out of the four nodes and the President and also one channel for all four to communicate with the president. Since all these nodes can have high computing power and need to deal with sensitive data, we can make use of either symmetric or public key encryption at this level.

Naturally, the hierarchy consists of heterogeneous nodes, with the nodes lower down having greater resource constraints. Depending on the capability of the nodes and sensitivity of message we provide suitable cryptosystem.

All these CHs are thus treated as trusted since they can be provided with more security than ordinary nodes. This is generally achievable as the number of cluster heads is much less as compared to ordinary nodes. We shall prove in section VII (Theorem 4) that the number of CHs $= O(\frac{N}{t})$.

B. Particular Case of Distributed Sensor Network (DWSN)

Since in sensor network the communication and the connection model is normally a pair-wise locally-complete, i.e., where any subset of two local (i.e., neighbouring) nodes can be allowed to communicate, hence we modify the Reed-Muller matrix, accordingly, in the case of more than three nodes under a Cluster Head (CH).

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

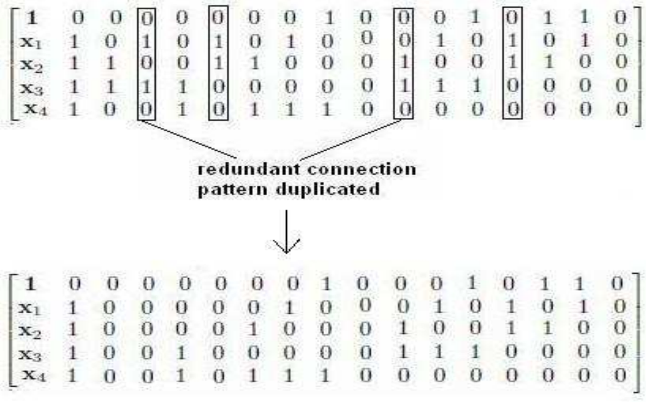
Here, the columns $2, 3, 5$ and 9 represent connection links connecting three nodes. Hence in a pair-wise locally connected network we can safely replace these columns with 0's resulting in the following connectivity matrix. Also the top-most cluster head associated with variable 1 has authority over all connections. This condition can be relaxed as follows: The top most cluster head may share only one connectivity link with each of the t nodes under it. Hence the connectivity matrix becomes:

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

Remark: The versatility of Reed-Muller is in its use for locally r-complete systems, where $r \leq t$ & $t \geq 2$ (see [7]).

C. Node addition

When a node needs to be added in a cluster of i pre-existing nodes, then the following changes need to be made to the connectivity matrix (see Figure 2). The dimension of the vector corresponding to each node is doubled, and for each of the pre-existing nodes the bit-pattern is duplicated and padded to the right. The new node to be added is assigned a new variable x_{i+1}, which has the Least significant half as all 0's and the most significant half assigned with 0's and 1's to represent connectivity with the pre-existing nodes. Furthermore, in those places in the Most Significant half of the variable x_{i+1} , where there is a 0, it is equivalent to the connection pattern among the pre-existing nodes being duplicated by alternate routes. Hence these columns can be safely made all 0s, without affecting the connectivity of the network.



$$\begin{bmatrix} 1 & 0 & 0 & \boxed{0} & 0 & \boxed{0} & 0 & 0 & 1 & 0 & \boxed{0} & 0 & 1 & \boxed{0} & 1 & 1 & 0 \\ x_1 & 1 & 0 & \boxed{1} & 0 & \boxed{1} & 0 & 1 & 0 & 0 & \boxed{0} & 1 & 0 & \boxed{1} & 0 & 1 & 0 \\ x_2 & 1 & 1 & \boxed{0} & 0 & \boxed{1} & 1 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & \boxed{1} & 1 & 0 & 0 \\ x_3 & 1 & 1 & \boxed{1} & 1 & \boxed{0} & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 1 & \boxed{0} & 0 & 0 & 0 \\ x_4 & 1 & 0 & \boxed{0} & 1 & \boxed{0} & 1 & 1 & 1 & 0 & \boxed{0} & 0 & 0 & \boxed{0} & 0 & 0 & 0 \end{bmatrix}$$

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

Figure 2. Generalized matrix for DWSN with four nodes

V. Various Network parameters

Here we discuss the various network parameters and highlight some of the improvements achieved.

A. Communication probability of the network

The communication probability of the network is defined to be the probability that two nodes can communicate with each other, i.e., the probability that there exists a communication key between them. Mathematically, we have

$$\rho_c = \frac{\text{Number of communication link present in networks}}{\text{Total Number of possible links}}$$

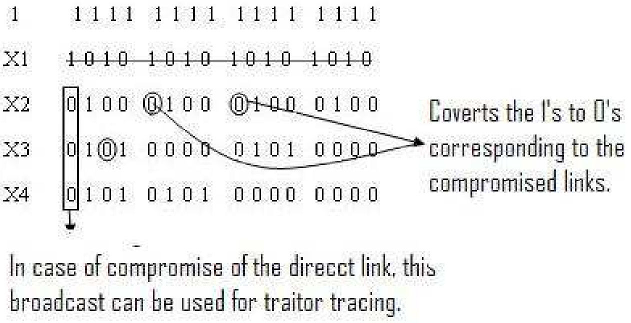
Theorem 1: *ρ_c is independent of connectivity.*

Proof: It is evident from our proposed model that each CH or node is connected to its sibling, its children (if any) and its parents (if any). So, when it wants to send a message to any of them, it does so directly. Otherwise, if it has to send to any node outside its cluster, it uses its CH or one of its children. Hence the communication is not dependent on the connectivity of the network as it is connected by some path to all the nodes in the network. The theorem follows.

B. Resilience:

Here we assume that we have a hypothetical intrusion detection (i.e., attack detection) mechanism to inform the KDS and subsequent nodes of the compromised node. When a node say, X_1 is captured, the keys that are compromised are the broadcast key, the keys between X_1 and the remaining nodes in that cluster and the key shared by its cluster head and X_1 exclusively, but it does not contribute to resilience.

One important characteristic of the use of Reed-Muller code is that it ensures, even after deleting all the above keys, the remaining nodes still remain connected with each other and hence can safely carry out communication along alternate direct or multi-hop links. As soon as the capture of X_1 is detected by other nodes through the broadcast channel, they delete all the keys that they shared with the compromised node X_1, which amounts to making the corresponding columns all 0s. Thus, the table now becomes:



1 1111 1111 1111 1111

X1 ~~1010 1010 1010 1010~~

X2 0100 0100 0100 0100

X3 0101 0000 0101 0000

X4 0101 0101 0000 0000

Coverts the I's to 0's
corresponding to the
compromised links.
In case of compromise of the direcct link, this
broadcast can be used for traitor tracing.

Figure 3. Changes in Matrix when one node (X_1) is compromised

Now consider the resilience due to capture of a newly added node X_{m+1}, in a pre-existing system consisting of $x_1, x_2, \ldots, x_m$. Since $X_m + 1$ does not contain the older set of $2m$ keys, (remember it has all 0s in the Least Significant Half). Hence when X_{m+1} is captured, it does not affect any of the pre-existing nodes which can still communicate using the older keys. Now we calculate some of the resilience coefficients.

Calculation of $V(s)$: $V_s = \frac{d_{n,ch}(s,t)}{N'}$, where $d_{n,ch(s,t)}$ = Number of uncompromised sensor nodes disconnected due to capture of s node and t cluster heads.

We break the $V(s)$ calculation into 3 parts.

$V_{node}(s) = \frac{d_n(s)}{N'}$, where $d_n(s)$ or $d_{n,ch(s,0)}$ = Uncompromised sensor nodes disconnected due to s node capture.

$V_{ch}(t) = \frac{d_{ch}}{N'}$, where $d_ch(t)$ $d_{n,ch(0,t)}$= Uncompromised sensor nodes disconnected due to t cluster heads capture & finally we calculate $V_{node,ch}(s,t) = \frac{d_{n,ch}(s,t)}{N'}$, where $d_{n,ch}(s_1, s_2)$ = Number of uncompromised sensor nodes disconnected due to s_1 node capture and s_2 cluster heads.

Theorem 2: *During key establishment when node id's are broadcast,* (i) $V_{node}(s) = 0$ *when s* $<$ *n,* (ii) $V_{node}(s) \leq \frac{N'-nq}{N'}$*. when s = n and* (iii) $V_{node}(s) \leq \frac{N}{N'}$ *when s* $>$ *n. After broadcast* $V_{node}(s) = 0$.

Proof: For capture of simple sensor node ($s = 1$), all keys in its key ring, all its key identifiers and its node ids get compromised. However nodes which are not communicating with are not affected. Thus none of the uncompromised nodes gets disconnected. Also no new key ring can be formed for $s < n$. Thus (i) follows

For $s \geq n$, we would like to refer our readers to the section 6.3: Analysis of V(s) of Roy and Ruj [1].

As has been noted earlier that unlike Roy and Ruj [1], an attacker cannot gain much information by listening to our connectivity channel. The encrypted node identifiers are passed only during broadcast in our system. So an attacker does not get the node ids directly. Thus unless physical capture of any node take place, the attacker can't get any extra information.

Also, after key establishment, the node ids obtained from the other nodes should be deleted from the node's and Cluster Head's memory. So that even if the attacker captures adequate number of nodes, he cannot predict the communications of any other node due to lack of node ids of those nodes. Hence the theorem follows.

Theorem 3: $V_{ch}(s) = 0$.

Proof: When a single CH is captured, i.e., when $s = 1$, the attacker can observe the communications being done through this Cluster Head. During key establishment, when all the node ids are broadcast in encrypted fashion, this cluster head can decrypt it and find the polynomials which can be solved to obtain the key identifiers of the keys that will be used for the communication for each node. However unless the communicating sensor node is captured the attacker cannot find out the key to which this key identifier maps. Thus he can only have some partial idea of communications of various nodes which is not sufficient. Hence $Vch(1) = 0$ in this case. Generalizing we have $V_{ch}(s) = 0$ during key establishment.

Post key establishment, compromise of a Cluster Head gives the attacker no extra information, as the node ids are no longer transmitted. The result follows.

Corollary 1: $V_{Ch+node}(s_1, s_2) = V_{node}(s_2)$. *That is* $V_{Ch+node}(s_1, s_2)$ *is independent of CH capture.*

Proof: From the proof of Theorem 3, it is clear that capture of only any number of Cluster heads yields the key identifiers of all the nodes. But this knowledge is of no good unless a node is captured. Hence the conclusion.

Remarks:

- The node identifiers are to be transmitted only once when key establishment takes place and this process is assumed to be very fast and secure. In the later stages, when the massage is to be sent, the sender encrypts it before sending and only the recipient can decrypt it using communication keys.
- The process of sending the message can be performed using the global IP address which the nodes can broadcast in the key establishment phase. Thus we can avoid repeatedly sending the node identifier and hence repeated decryption encryption at the Cluster Head.
- In case one want to avoid the use of global id, then node id has to be broadcast at every step. However still at least n nodes are to be captured to affect other nodes. Whereas in Ruj and Roy [1] the attacker gains much information by listening only to the communication channel which is not the case here.

Alternative Communication Approach:

Alternatively, if we remodel the system in such a way that during communication key establishment, when the broadcasted node id reaches the parent CH of the recipient sensor node, it does the polynomial evaluation of the broadcast node id and the recipient node id to find out the common key ids, if any. Next, it securely transmits the ids of the common keys to the recipient node. Thus, even if a sensor node is captured before key establishment, the attacker can only find out which node shares a key with the said compromised node. But he does not gain any information about the key ring of the other communicating nodes. Hence the capture of a node does not affect the communication among the other communicating parties. Also in the event of the capture of a Cluster Head security is not breached, since the actual keys are still secure. Thus, for this improved system, we get $V_{ch}(s) = 0$ and $V_{node}(s) = 0$.

Calculation of E_{con} : $E_{con}(s) = \frac{cl_{brk}}{cl_{bc}}$, where cl_{brk} = number of connectivity links broken due to capture of s nodes or cluster heads and cl_{bc} = total number of connectivity links present before capture. Since this concept is more related to communication, we give a brief outline of the following major issues needed for our model:-

1) Capture of a simple node at the lowest level: Consider there are t nodes in the cluster, one of which gets captured. When the node is captured, the connection link with its parent CH, the $t-1$ connection with the $t-1$ siblings in its cluster and the broadcast channel used by nodes in that cluster are broken. Thus a total of $t+1$ links which were connected to the captured node are broken. The remaining links remain unaffected.

2) Capture of a Cluster Head: After key-establishment, if a CH is captured, all links through it gets affected, however the resiliency of the system is unaffected. It is clear that now messages pass through it in encrypted fashion. During broadcast of keys, in the unlikely event of a CH compromise, the situation is more complicated. Here we have to make provision for some extra (buffer) CHs at each level. These Buffer CHs will be empowered to replace any CH and distribute fresh keys.

C. Scalability

Our model is scalable in the sense that any number of nodes can enter the network. In such a case there may be a case of increasing the number of tier with the q_{RM} fixed or choosing an higher value of it. Also the new nodes can be given connectivity key by rotation policy and use different frequency. Communication protocol will then dictate its communication keys.

VI. Security and Cluster head estimate, capacity

The connectivity model is determined in the pre-deployment phase. The wireless channels for connection can be made secure by using pre-deployed connectivity keys. Thus in our system the key ring is never sent in clear over the channel. We have also noted that the CHs play a very important role in the resiliency of the model. Thus it becomes necessary to have an estimate of the number of CHs and their storage capacity.

Theorem 4: *Number of CH $\sim O(\frac{N}{t})$, where $t \leq \frac{n}{2}$.*

Proof: If there be t-children at for each CH, then baring the KDS, each CH will have $2t$ connections (1 for its own CH, $t-1$ at its level and t children). Now if we restrict the connection to be n, i.e. $\leq$ to communication keys per node, we must have $t \leq n/2$. It can easily be seen that at the level just above the lowermost level, there are $\lceil N/t \rceil$ CHs. At the level above it there are $\lceil N/t^2 \rceil$ CHs and so on. Now as the height of the tree is $logN$, number of CH = $N \cdot [\sum_{i=1}^{r+1}(\frac{1}{t^i})] \sim O(\frac{N}{t})$. Hence the result.

VII. Conclusion and Future work

In this paper, firstly we have differentiated between connectivity and communication of a Wireless Sensor Network. Then to make the communication more secure, we have used cryptographic techniques in the connectivity models. We would also like to highlight that our connectivity model is based on 1st order Reed-Muller code.

Though one can use any communication model, we have based our calculations on the model proposed by Ruj and Roy in [1]. However as compared to them or Lee and Stinson's scheme of key pre-distribution [4], our resiliency is appreciably improved. As observed, the system is also scalable.

However, there is scope for further developments in this direction. For example, in the current model, repeated enciphering and deciphering is being done at each CH in between two communicating nodes of different clusters. It may be a nice work to develop a system avoiding this. To this end, it may be fascinating to see if one can apply any coding or other techniques. Moreover, in Ruj and Roy [1] each key is shared amongst q^{k-1} nodes, where $q^{k-1} \leq N \leq q^k$ & $q-$ a prime power. Codes may also be used in seeking a better system in which less number of nodes share the same key and still the system is scalable with improved resiliency.

Acknowledgment

We would like to thank Prof. Bimal Roy and Mr. Sumit Pandey of Indian Statistical Institute, Kolkata for discussing the paper and critically analyzing it. We would also like express our gratitude to Dr. Goutam Paul and Mr. Dibyendu Majumder of Jadavpur Univesity, Kolkata for their active participation in preparation of the paper.

References

[1] S. Ruj, B. Roy, *Key Predistribution Schemes Using Codes in Wireless Sensor Networks* Inscrypt 2008, LNCS 5487, pp. 275-288, 2009. Springer-Verlag Berlin Heidelberg, 2009.

[2] L. Eschenauer, V. D. Gligor, *A key-management scheme for distributed sensor networks* In: Atluri, V. (ed.) ACM Conference on Computer and Communications Security, pp. 41-47. ACM, New York, 2002.

[3] S. A. Camtepe, 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.

[4] J. Y. Lee, D. R. Stinson: *Deterministic key predistribution schemes for distributed sensor networks,* In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 294-307. Springer, Heidelberg, 2004.

[5] S. Ruj, B. Roy, *Key establishment algorithms for some deterministic key predistribution schemes* In: Rodrguez, A., Yage, M., Fernndez-Medina, E. (eds.) Workshop on Security In Information Systems, INSTICC, 2008.

[6] M. Al-Shurman and S. M. Yoo, *Key pre-distribution using mds codes in mobile ad hoc networks,* In: ITNG, pp. 566-567. IEEE Computer Society Press, Los Alamitos, 2006.

[7] Ruj. S, *Application of Combinatorial Structures in Key Predistribution to Sensor Networks using Combinatorial Designs,* Ph.D. thesis, Indian Statistical Institute, 2009.

[8] D. R. Stinson, *Combinatorial Designs: Construction and Analysis,* Springer, New York, 2004.

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