Appendix 5

List of Publications


WSN: Key Issues in Key Management Schemes-A Review

D. Manivannan and P. Neelamegam
School of Computing, SASTRA University, Thanjavur-India

Abstract: The real challenges in the field of Wireless sensor networks (WSN) are to control the trade-off between providing security and conserving scarce resources. To provide security besides cryptographic primitives, secret key also play a vital role. The reason for key management is to set up and to provide secure channels among sensor nodes in WSN applications. Various key management schemes are proposed for WSN. Deploying sophisticated network based Key Management protocol in WSN is critical and important task. From the literature survey, it is understand that more energy is consumed to establish wireless sensor network in any real time applications. In this paper, various key issues in key management schemes are discussed based on the evaluation metrics such as energy responsiveness, key connectivity, resilience, scalability and efficiency.

Key words: Distributed wireless sensor networks, group keying, hierarchical wireless sensor networks, lu matrix, network keying, pair-wise

INTRODUCTION

A wireless sensor network comprises large number of distributed, self-directed devices called sensor nodes also called as motes. WSN naturally encompasses a large number of spatially dispersed, petite, battery-operated, embedded devices that are networked to supportively collect, process, and convey data to the users, and it has restricted computing and processing capabilities. Sensor node consists of three major units like sensing, processing and transceiver Unit. The sensor field of WSN comprises of different n number of heterogeneous and homogenous nodes (Du et al., 2007) which are deployed in open surroundings and the interconnection among the nodes is established through wireless transceivers and that to be grouped with corresponding cluster heads and connected to gateway, and to end with the base station. In homogenous Sensor Networks, all sensor nodes have the same property in terms of computation, communication, memory, energy level and reliability. In heterogeneous Sensor Networks, the mixture of different sensors has the different capabilities in terms of computation, communication, memory, energy level and reliability.

It’s very important to evaluate the list of factors which is used to determine the performance of a sensor network. The general issues of WSN are data redundancy, addressing scheme, message size and resources. WSN’s general security goals are; Confidentiality, Integrity, Authentication:, Availability, Survivability, Efficiency, Freshness and scalability. Wireless sensor networks are susceptible to many attacks because of its transmission nature, resource restriction on sensor nodes and deployment in uncontrolled environments. To ensure the security services in WSN many crypto mechanism are proposed. But the security strength of the entire cryptosystem of WSN mainly depends on the secret keys used, not in the algorithm. The reason of key management for WSN is to load, distribute and handle the secret keys in sensor nodes to establish a secure communications among sensor nodes. Security critical applications are depends on key management scheme because it has to provide high fault tolerance when a node get compromised. While designing the key management schemes, the important metrics to be evaluated: Local / Global Connectivity: each node communicates with every other node in the sensor field region. Resilience: whenever a sensor node is compromised, the key management scheme assured how secure the remaining communication link there will be i.e., resistances against node capture (Marcos et al., 2010) Scalability: capability to support when large numbers of nodes are added to the sensor network. Efficiency: In terms of storage, communication and computation.

In general key distribution scheme are widely classified into three categories. These are network, pair-wise and group keying. In network keying a single shared secret key is used for communication. This keying mechanism is excellent in terms of scalability and flexibility. But when one node gets compromised, entire network will gets compromised. In pair-wise scheme for n node, n-1 keys are stored, with the total n (n-1) / 2 distinguishable keys in the network. Though it achieves perfect resilience, scalability is an issue. Group keying mechanism combines network keying and pair-wise keying mechanisms (Yang et al., 2007). Many Key management scheme are proposed for WSN are based any
The key management schemes in wireless sensor networks are classified as follows; these are: Network-Wide Key distribution-BROSK, SKKE; Pairwise Scheme-full pairwise scheme, Probabilistic Based approach: EG scheme, Q composite scheme, Multipath Key Reinforcement Scheme, Random Pair-wise Key Scheme, key redistribution scheme and AP Scheme; Matrix Based: Blom’s scheme, multi space key distribution scheme, LU Matrix and hierarchical LU matrix; Polynomial KMS: Blundo, polynomial pool based key pre-distribution scheme; Deployment Knowledge: Extending Probabilistic scheme, extending matrix based scheme; Group Keying: Tree based approach, EBS: SHELL, LOCK and LEAP.

In this study Network Wide Key Distribution Schemes, Pairwise key management schemes, Matrix based schemes, Polynomial based key management schemes, Key management with deployment Knowledge, Group keying schemes are discussed, finally summary of various key management schemes are done.

**Network wide key distribution schemes:** The general key distribution schemes in WSN are broadly classified into three types. They are network keying, pair wise keying and group keying. These classifications mainly concentrate on the number of keys being used in the system. The network keying scheme is very simple, manageable and involves the usage of very less resources. This method satisfies all the functional requirements like accessibility and self organization. A single key is used for all the nodes in the network. All the secured data processing in sensor nodes depends on a single secret key. All the nodes in the entire network use only a shared secret key. For example, in broadcast session key negotiation protocol (BROSK), proposed by (Lai et al., 2002), the secret session key for pair of nodes A and B is generated with the help of master key K, random nonce RN_A and RN_B and round function F. The secret session key K_{AB} = F(K||RN_A||RN_B), Where F is pseudo random function (Marcos et al., 2010). Like that, in Symmetric-Key Key Establishment protocol (SKKE), key establishment involves in between initiator node and responder node. The common shared secret key K_{AB} is found with the help of master key K, random nonce RN_A and RN_B, Identifier, responder and round function F. The common shared secret session key K_{AB} = F(K||RN_A||RN_B||ID_A||B), where F is pseudo random function. It is very flexible, scalable and self organized key distribution scheme. The main drawback of this scheme is the lack of robustness. Compromisation of any one node in the network will affect the security of the entire WSN.

**Pairwise schemes:**

**Full pair wise scheme:** Full pairwise scheme is a very basic and easily acceptable system. Each pair of node in the sensor field will have different key for data communication. Communication between any two nodes will take place only after the establishment of a common key between them. In this scheme, N-1 keys are required in each node where N is the total number of sensor nodes in the network. The total number of distinguishable keys in the network is N(N-1)/2. Using this scheme, each node can be authenticated and it also provides good robustness (Marcos et al., 2010). The main drawback of this scheme is its non-flexibility, non scalability and complexity.

**Probabilistic approach:**

**EG scheme:** This scheme proposes a simple key distribution in distributed wireless sensor networks. This scheme (Eschenauer and Gligor, 2002) consists of three basic steps they are

- Key Pre-distribution phase
- Shared secret key discovery
- Path key establishment phase

Prior to distributed WSN (DWSN) deployment, the first phase is ensured. In key initialization phase the large pool of P Keys is generated with its identifier. Then k Keys are selected from large pool P and stored in each sensor node memory unit. These k Keys along with its identifier is called key ring. After establishment of these steps, it is important to ensure that only less number of keys is placed on each sensor node and also ensure that any two nodes in the field share a common secret key. During second phase, the nodes have to find shared common key between its neighbors. Using simple broadcasting techniques with key identifier, the shared keys are discovered. It leads to general attacks by an adversary. This methodology provides easy loopholes to any misfeasor who wants to analyze or affect the sensor network. Alternate scheme for shared key discovery phase is generate random value & encrypt with the keys in the key ring and then broadcast. The node which shares the common key will decrypt and get the value of α. So the node will communicate with that key.

In the path key establishment phase, at the end of shared key discovery phase some nodes do not have the shared key between the neighboring nodes, so communications between nodes are not possible. Such node will establish a path key. For new path key instead of generating a new key, the key ring will have some unused keys. That key can be used as path key. These
path key will be shared between nodes using the communication link which is discovered during shared key discovery phase.

**Revocation:** In DWSN, the entire sensor nodes in the sensor field must be controlled by controller node. There may be many controller node and group of sensor nodes comes under the control of controller node. Each node shares a secret key with the controller node. Controller node also maintains the information like the list of key identifiers each node is having. If any sensor node in the sensor field is compromised, revoking the entire key ring of that particular node is very important and also essential.

Controller node broadcasts a single revocation message, which contains a signed list of key ring of compromised node. Controller node generates the signature key $K_s$ for sign the list of key identifiers, then it encrypt with secret key which is shared with the nodes and unicast it to each and every node in the network. All the sensor nodes decrypt the signed message and it verifies the signature of the signed list and locates the corresponding keys and finally it will be removed. After revocation of the compromised key ring, reconfiguration of links is done by starting the shared key discovery phase followed by path key establishment phase. If the life time of the keys in the key ring expires, the self revocation of shared secret key by a node in the network is done.

Key connectivity is proportional to the sensor node memory. If the number of keys loaded into sensor node is high. Better key connectivity is obtained in this scheme. The advantage of this scheme is simple, efficient and scalable, than full pair-wise scheme. Communication cost also high because the node get compromised the revocation message is unicast to every node in the network. This scheme fails to provide node to node authentication so this scheme is not suitable for high end security applications.

**Q composite scheme:** This scheme (Chan et al., 2003) is similar to basic schemes; the main difference is that EG scheme (Eschenauer and Gligor, 2002) requires single key to communicate between neighboring nodes but Q composite scheme requires $q$ keys i.e. $q > 2$. The rest of the phases are similar to basic scheme i.e., generating the large key pool $P$, and from the key pool randomly select $m$ keys without replacement and load in to the sensor node. In Key sharing phase, each node found out the common keys with each of its neighbors. This can be done by each node by broadcasting its key identifier. Two nodes can communicate each other if both node have $q'$ keys in common where ($q' > q$). Once common keys are discovered, shared secret key $K$ between two nodes has to be established by taking hash of all common keys. i.e., for example $K = \text{Hash}(k_1||k_2||...||k_q')$. They can be hashed in any order preferably in the order of key pool.

The major work involved in Q composite schemes is computation of key pool size. If the key pool size is large, the probability of sharing common keys between two nodes will be less. If the key pool size is reduced, even when small number of nodes is compromised, the attacker can get large keys of $P$. Based on this; the key pool size has to be chosen in such a way, the probability of sharing common keys should be in expected level and at the same time security level should also be maintained. With respect to security analysis, this scheme achieves perfect resilience against node capture compared to EG scheme (Eschenauer and Gligor, 2002). For example when $q = 2$, $P = 0.33$, $m = 200$: for this criteria if 50 nodes get compromised, the additional communication link compromised will be 4.74 in Q composite, where as 9.52 in EG scheme. But the drawback is if the large number of nodes compromised, there is a possibility of compromising the entire communication link. The advantage of this scheme is achieving better resilience but scalability is an issue. This scheme also fails to provide node-to-node authentication.

**Multipath key reinforcement scheme:** Chan et al proposed a scheme (Chan et al., 2003) which offers good security compared to EG scheme. In Q composite scheme the links are stronger than basic scheme because each link shares $q$ keys ($q > 2$). General problem in basic scheme is, when a link gets compromised, there is a probability of breaking many communication links since keys are randomly selected from key pools, so probability of repeating same key will be more. When one node gets compromised, establishing new key using previously established link is insecure. So link must be secure before updating the communication key. For that find out the multiple independent path from source to destination node. For example if node $A$ wants to establish a new communication key with node $B$, first multiple disjoint path have to be find out say $h$ paths. Then node $A$ generates $h$ random values and sends out in all possible disjoint path. Once after receiving, node $B$ computes the new key. Simultaneously node $A$ also computes the key as $K' = K \odot g_1 \odot g_2 \ldots \odot g_h$, where $K$ is the original key from the key pool. Larger the value of $h$ greater will be the security but leads to communication overhead and also finding the independent path from source to destination is also difficult. A Two hop approach is recommendable i.e., one intermediate node between source and destination. The main advantages of this scheme is that it achieves better security than basic scheme and Q composite scheme but there will be a communication overhead.

**Random pair-wise scheme:** This scheme (Chan et al., 2003) combines the features of full pair-wise scheme and EG scheme. Full pair-wise scheme achieves node-to-node authentication, perfect resilience and perfect key connectivity. But storage is problem i.e. if $n$ nodes are in
the network then n-1 keys are loaded into sensor nodes and scalability is difficult. But in EG scheme storage can be reduced but fails to provide node to node authentication and perfect resilience. Chan et al. (2003) proposed a random pair-wise scheme in that for n nodes, m (m-n) unique identifiers are created. Each node has unique identity and is paired with K randomly selected identifiers. Pair-wise key is generated for each pair and is loaded into sensor node. Finally each sensor node is loaded with its identity, K identifier which is paired with its id and K corresponding pair-wise keys which is less than n-1 keys in fully pair-wise scheme. Once after loading shared key discovery phase starts, i.e. each node broadcasts its id. Nodes will find out the neighboring node which shares the same key. While identifying the neighbors, sometimes it may be beyond the communication range also, for that the node id will be rebroadcasted. But this rebroadcast should be limited to number of hops to avoid denial of service attack.

A random pair wise scheme adopts different mechanism for revoking the compromised node than basic scheme. In basic scheme, revocation is initiated by controller node. But here, any nodes which identify the compromised node will cast a vote against it and broadcast to all other node. If threshold t, vote has been casted, that node will get revoked. This is done to reduce the overhead of the base station. For voting each node is loaded with a key K_i and K-1 voting member of the hash values. So if K pair-wise keys are loaded in the node the storage requirement is O (K^2). Though random pair wise scheme achieves perfect resilience against node capture, scalability is an issue, because it can be extend till m-n nodes only. Storage also an issue because it has to store its identity and K identifiers, K pair wise keys, voting key K_i, and K-1 hash values and threshold t value.

**Key redistribution scheme:** Law et al proposed a scheme (Law et al., 2007) with small changes in basic scheme i.e. it replaces the original path key establishment instead it includes key redistribution scheme. For example node A and node C share a common key K_1 and Node B and Node C share a common key K_2. There is no common key between Node A and Node B, so Node A does not communicate with Node B. In general Node A sends a request message to node C to get K_1. Node C encrypt K_2 value using K_1 and send to Node A. Node A will decrypt the received value using K_1 and get the K_2, and establish the communication between Node B. If suppose Node C refuses Node A request, the alternate approach is Node A select an unused key K_3 from the key ring and send to Node C. Node C computes new key K_23, where K_{23} = Hash(K_2||K_3) then K_23 encrypted with K_1 and K_2 and send back to Node A and Node B respectively. Node A and Node B will decrypt the value using K_1 and K_2 and get the value of K_{23}. Now Node A and Node B will have the common key K_{23}, using that they establish the communication link between them. Compared with basic scheme, this Key Redistribution scheme provides higher key connectivity and also provides better resilience but the drawback is communication overhead.

**AP scheme:** In Basic scheme, to achieve higher key connectivity the number of keys loaded into sensor nodes should be high. For example if P is 10,000, each node should load more than 150 keys to achieve 0.9 key sharing probabilities. Since sensor nodes are resource constrained, loading large number of keys is difficult. To overcome this (Du et al., 2005) proposed a scheme for heterogeneous sensor networks called AP (asymmetric pre-distribution) scheme. Since heterogeneous it includes two different types of sensors called H sensors and L sensors. In terms of storage, computation and communication range H sensor will have strong capabilities than L sensors. H sensor and L sensors are scattered in the network. L sensor which comes within the communication range of H sensor will form a cluster. H sensor will act as a cluster head. As like basic scheme, AP scheme also undergo three phases. The phases are Key pre-distribution, shared key discovery and path key establishment. The difference is if any L sensor does not share any key with its neighbors H sensors obtains the key of both nodes compute the key and send to each node. The performance analysis of this scheme shows that it achieves better storage than EG scheme (Eschenauer and Gligor, 2002). For example if 1000 L sensor and 10 H sensors are there in AP, the keys loaded in L sensors is 20 and H sensor is 500 but for EG scheme, each L sensor is loaded with 150 keys to achieve the same key probability. This scheme works better in resilience because H sensors are tamper resistant and also compromising L sensor will not reveal much of the communication link because number of keys loaded in L sensor is less.

**Matrix based scheme:**

Blom scheme: In a perfect pair wire scheme, if the network consists of n users each user will have to store n-1 keys. If the size of the network increase number of keys stored in each nodes will be increased. Since sensor nodes are resource constrained implementing perfect pair wire scheme is difficult.

Blom proposed (Blom, 1985) a symmetric key generation system in which with the least information every node will communicate with every other node. Blom scheme is λ secure property i.e. the communication links will be secure until λ nodes get compromised. When λ nodes get capture the entire communication links in a network will get compromised.

To establish a pair wire key between nodes, the procedure is:

**Step 1:** Construct a (λ+1) x N matrix G over a finite field GF (q) where N represents the number of
nodes and $q > N$. $G$ is a public matrix, so even an adversary may know about this matrix. Matrix $G$ is a Vandermonde matrix which takes the forms as follows

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \lambda & \lambda^2 & \lambda^3 & \cdots & \lambda^N \\ \lambda^2 & (\lambda^2)^2 & (\lambda^3)^2 & \cdots & (\lambda^N)^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \lambda^i & (\lambda^2)^i & (\lambda^3)^i & \cdots & (\lambda^N)^i \end{bmatrix}$$

The use of Vandermonde matrix is, instead of storing an entire column the seed can be stored, and from that each node can compute the column values.

**Step 2:** Construct the another matrix $(\lambda + 1) \times (\lambda + 1)$ symmetric matrix $D$ over a finite field $GF(q)$. From the two matrixes $G$ and $D$, construct $A$ matrix, i.e. $A = (D.G)^T$. Matrix $D$ has to be secret and it should not be disclosed to adversary. All the matrix construction will be done in base station. Since $D$ matrix is symmetric.


If $K = A.G$ then $K_i = K_{ji}$. Based on this idea, the pairwise key can be computed as follows.

Consider Node $I$, which store the $i^{th}$ row of $A$ matrix and seed to generate the column. Similarly Node $J$ stores the $j^{th}$ row of $A$ Matrix and seed. If Node $I$ and $J$ wants to establish a pair wise key $K_i$ and $K_{ji}$ Node $I$ will send its seed or ID to Node $j$ and vice versa. Simultaneously both nodes will compute the pair wise key. As mentioned earlier, Blom scheme is a secure properly if $\lambda$ nodes get compromised the entire network will get revealed thus resilience is not optimal.

**Multi space key distribution scheme:** Based on Blom scheme Du et al. (2005) proposed a scheme (Marcos et al., 2010) with small changes. Blom Scheme is a perfect single key space but. Du et al. (2005) uses multiple key spaces. Various phases involved in Du et al. (2005) scheme to establish a pair wise scheme is In Key Pre-Distribution Phase: All nodes will have unique identity ranging from 1 to N. Generate $G$ matrix and Generate $\omega$ key spaces. Since Du et al. (2005) scheme lies multiple key spaces, generate $\omega$ symmetric matrices $D_1, D_2, \ldots D_\omega$ of size $(\lambda + 1) \times (\lambda + 1)$. From the two matrixes $G$ and $D$ matrices compute $A_i = (D_i . G)^T$ and select $\Gamma$ unique key spaces and stored in each node. Nodes which share the same key space will establish a common shared key as Blom scheme. In Key agreement phase, after deployment each node has to establish a shared key with its neighbor for that each node broadcast its node identifier and index of the key spaces. The nodes which do not share common key spaces have to undergo path key establishment phase as in EG Scheme (Eschenauer and Gligor, 2002). Since $\Gamma$ key spaces are loaded into sensor node and each key spaces is having $(\lambda + 1)$ elements, the total memory usage will be $M = (\lambda + 1) \Gamma$. This scheme achieves optimal resilience but incurs computation overhead as like Blom’s scheme. Scalability is another issue. To achieve key connectivity this scheme undergo path key establishment phase which incurs communication overhead.

**LU matrix:** In LU matrix (Dai et al., 2010), two matrixes called Lower triangular and Upper triangular matrix are involved. Both the matrix is used to generate the secret Key matrix for sensor nodes. From the randomly generated key pool, Lower triangular matrix is generated. Based on assumptions that product of $L$ matrix and $U$ matrix will produce symmetric matrix $K$, the Upper triangular matrix is generated. The relationship between $L$ matrix, $U$ matrix and $K$ matrix is clearly given as follows.

$$\begin{bmatrix} 11 & 0 & \cdots & 0 \\ 21 & 22 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ N1 & N2 & \cdots & NN \end{bmatrix} \begin{bmatrix} U11 & U12 & U13 \\ U22 & U22 & U23 \\ \vdots & \vdots & \cdots & U23 \\ N1 & N2 & \cdots & N1 \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ \vdots & \vdots & \cdots & V33 \end{bmatrix}$$

Since $K$ matrix is symmetric, $K_j$ will be equal to $K_{ji}$, the value of $U_{ij}$ is calculated using the following relationships, and these are

$$k_{12} = k_{51} = l_{11} . u_{12} = l_{21} . u_{11} \Rightarrow u_{12} = (l_{21} . u_{11}) / l_{11}$$

$$k_{13} = k_{51} = l_{11} . u_{13} = l_{13} . u_{11} \Rightarrow u_{13} = (l_{13} . u_{11}) / l_{11}$$

$$k_{23} = k_{32} = l_{21} . u_{13} + l_{22} . u_{23} = l_{31} . u_{12} + l_{32} . u_{22}$$

$$u_{23} = (l_{31} . u_{12} + l_{32} . u_{22} - l_{21} . u_{13}) / l_{22}$$

Using the random values of $u_{11}$, $u_{22}$ & $u_{33}$, the remaining unknown values of $u$ matrix are calculated. The important condition is the product of $L$ and $U$ matrix would be symmetric.

From the value of $L$ and $U$ matrix, the value of $K$ is calculated

$$I_{11} . u_{11} = k_{11}; \frac{l_{11} . u_{12} = k_{12}, l_{11} . u_{13} = k_{13}, l_{21} . u_{12} = k_{21}, l_{11} . u_{11} = k_{31}, l_{21} . u_{12} + l_{22} . u_{22} = k_{22}, l_{21} . u_{13} + l_{22} . u_{23} = k_{23}, l_{31} . u_{12} + l_{32} . u_{22}$$
\[ k_{32} = l_{31} \cdot u_{13} + l_{32} \cdot u_{23} + l_{33} \cdot u_{33} = k_{33} \]

In general when pair-wise scheme is consider, many have an assumption that it is suitable for DWSN. But the drawback in DWSN every node must communicate with every other node since sensor nodes are scattered and have short transmission range, establishing a pair wise key is difficult. Storage is also not efficient. When pair wise scheme is implemented in HWSN, group of sensor node will form the cluster so the key loaded into sensor node will be less. But the problem is most of the information will be loaded in cluster head. So when cluster Head gets compromised all information will get revealed.

**Hierarchical LU:** Based on all above mentioned problems Wen *et al.* (2007) proposed a key management scheme using LU matrix for HWSN with small information’s loaded in cluster head (Wen *et al.*, 2007). This scheme is three tier architecture: base station, cluster head and sensor nodes. Two different symmetric key matrixes are generated: SM\(_{CH}\) & SM\(_{CS}\). SM\(_{CH}\) is used to communicate between cluster heads. SM\(_{CS}\) is used to communicate between cluster head and sensor nodes.

Initially all nodes are loaded with one row and one column from its corresponding LU matrix. Rows and Column are selected randomly. If two nodes or two cluster heads or cluster head and sensor nodes wants to communicate procedure to follow is: (between Cluster Head)

- Cluster head \( CH_i \) send its column \( U_{CH_i,c} \) to \( CH_j \)
- Once after receiving Cluster head \( CH_j \) compute \( K_{CH_{ji}} \) and send its column and \( F(K_{CH_{ji}}) \) to \( CH_i \)
- Cluster Head \( CH_i \) compute \( K_{CH_{ji}} \) and verifies \( F(K_{CH_{ji}}) = F(K_{CH_{ij}}) \). If equals it sends \{ok, \( F(K_{CH_{ji}}) \)\} else it sends error message which will be authenticated by the receiver using any authentication methods such as TESLA (Perrig *et al.*, 2002)

The Key \( K_{CH_{ji}} \) will be used as pair wise key between two cluster heads.

In this scheme when cluster head get compromised nothing will get revealed, since all information is stored in base station. But some node has to be act as cluster head because sensor nodes are not in contact with base station. Any node cannot be elected as cluster head also because that node doesn’t have information about SM\(_{CH}\). The change to be done is replace with other cluster head. When compared with the blom scheme (Blom, 1985), transmission overhead is more because blom scheme broadcast the seed of column but in this scheme the entire column will be broadcasted, but computational complexity is less because no need to generate the column. In LU matrix most of the row and column elements will be zero, so memory requirement is less. But the problem with this scheme is scalability because when more number of sensor nodes is added to the network possibility of selecting the same rows and columns from the matrix will be high.

Based on this hierarchical based LU matrix scheme, (Manivannan *et al.*, 2011) proposed another scheme will some added advantage. In this scheme both node to node communication and group communication has been achieved using LU matrix. But group communication is used for commanding the node rather than sending data with the group key. Because in the above scheme if any node get compromised it has to be unicasted to each node by cluster head. So communication overhead is more. The methodology behind this scheme for group communication is the same row and same column is loaded into group of sensors. So all nodes will share a same key, for group communication. Regarding the scalability, when number of nodes added is more, instead of selecting row and column from the same LU matrix, base station will create another new LU matrix and load row and column from that matrix, so repetition of keys will be avoided.

**Polynomial based schemes:**

**Blundo’s scheme:** This scheme (Blundo *et al.*, 1993) is based on polynomial based key pre distribution scheme. Blundo scheme uses \( \lambda \)-degree bivariate polynomial \( F(x,y) \)

\[
F(x, y) = \sum_{j=0}^{\lambda} a_{ij}x^jy^{\lambda-j}, \text{ over } GF(q) \text{ which satisfies the property } F(x, y) = f(y, x), q \text{ is a prime and cryptographic keys should be within this } q. \text{ Once generating the polynomial, polynomial shares are loaded into sensor node. For example node } i \text{ get it polynomial share as } f(i, y). \text{ Similarly node } j \text{ gets its polynomial share as } f(j, y). \text{ To establish common key between two nodes } i, j, \text{ node } i \text{ evaluate } f(i, y) \text{ at node } j \text{ and node } j \text{ evaluate } f(j, y) \text{ at node } i. \text{ Memory requirements will be } (\lambda+1)\log_2 q. \text{ Like Blom’s scheme (Blom, 1985) it won’t cause communication overhead but Blundo scheme also } \lambda \text{ secure property. If } \lambda \text{ nodes get captured, the entire network information will get revealed.}

**Polynomial pool-based key pre distribution:** Based on Blundo’s scheme (Blundo *et al.*, 1993) and basic scheme Liu *et al* proposed a polynomial pool based scheme (Liu *et al.*, 2003) similar to Du *et al.* (2005) multiple space key pre distribution scheme. Generate \( \omega \) randomly bivariate polynomials of the form \( f(x, y) = \sum_{j=0}^{\lambda} a_{ij}x^jy^{\lambda-j} \) called polynomial pools. As basic scheme, randomly select the polynomial and load into sensor node Liu *et al.* proposed two instances establish a communication key. In the first approach i.e. random subset assignment each node is loaded with set of id’s with which the node will share the
common polynomial. Though these schemes simplify the shared key discover phase and achieves perfect resilience, scalability will be an issue. To overcome this basic scheme methodology can also be used i.e.; randomly select the polynomial and load in to sensor nodes. In shared key discover phase nodes will broadcast its polynomial ids. The node which shares the common polynomial id will establish a common key. The nodes which do not establish a common key will undergo path key establishment phase. Problems with this scheme is that if any one communication link is compromised, chance for compromising remaining links also possible since polynomials are randomly drawn from polynomial pool, So chance for repetition is possible.

Second approach is based on grid based pre-distribution in which m x m 2 dimensional grid is constructed from a set of 2m polynomial i.e. \( f_c(x,y) \), \( f_r(x,y) \) where \( 1 \leq \alpha, \beta \leq m \), \( m = \sqrt{n} \). Where \( n \) represent the size of the network. Each row \( \alpha \) is associated with the polynomial \( f_{\alpha\beta}(x,y) \) and each column \( \beta \) is associated with the polynomial \( f_{\alpha\beta}(x,y) \). In the grid, sensor node \( i \) is assigned to \((\alpha, \beta)\) and the polynomial share is \( f_{\alpha\beta}^\alpha(x,y) \) and \( f_{\alpha\beta}^\beta(x,y) \). During shared key discovery phase node \( i \) and \( j \) will establish a key if \( c_i = c_j \) or \( r_i = r_j \). If not equal path key establishment phase is performed. This grid based scheme is further extended to hyper cube, instead of 2 dimensions \( \Gamma \) dimension grid is constructed. If the \( \Gamma \) is high, key connectivity will be less during shared key discovery phase. The advantage of using grid based and hypercube based scheme is that the id’s are not broadcasted so communication overhead will be less and achieves optimal resilience.

Deployment Knowledge:
Extending probabilistic schemes: The schemes which proposed earlier does not depend upon deployment knowledge since sensor nodes are randomly scattered in an area, it is hard to find deployment region. But sometimes for example if sensor nodes are scattered to particular point, those sensor nodes can be grouped in particular pattern and based on this node location can be obtained. If deployment knowledge is incorporated in key management scheme it leads to better storage, connectivity and better resilience. Du et al. (2005) first proposed deployment knowledge based key management (Du et al., 2004) scheme which is the extension of basic scheme. This scheme has undergone two models.
• Node deployment knowledge in the network
• Develop a key pre-distribution scheme

Node deployment knowledge in the network: If \( N \) sensor nodes are deployed in a location, they are divided into equal size groups and grid base approach is used for deployment mode. Groups which deployed closely will share common keys. If the distance between the groups increases the overlapping of keys will get decreased. In basic scheme keys are loaded into sensor nodes from the same key pool but in this scheme keys are loaded from different pools for different group. Ultimately the key pool \(|S|\) is divided into sub key pool. Each pool will have \(|Sc|\) Keys.

Key pre-distribution scheme: This scheme also consists of three phase: Key pre-distribution, Shared Key discovery and path key establishment. First phase alone is different in this scheme, rest of the phases are same like EG scheme (Eschenauer and Gligor, 2002). In key pre-distribution phase, key pool is divided into \( t \times n \) keys. While dividing the key pool, the neighboring pools will have more common keys. Each node in the group is loaded with the key from corresponding key pool.

To create key pools: Since deployment model follows grid based approach
• Pools which are horizontal or vertical share \( \alpha|S|\) keys, where \( 0 < \alpha < 2.5 \)
• Key pools which are diagonal share \( \beta|S|\) keys, where \( 0 < \beta < 2.5 \)
• Two non neighboring pools don’t share keys.

The procedure for selecting the key for various key pools is
• Select \( |S| \) keys from \(|S|\) for the group placed in \( S_{1,1} \) and remove those key from \(|S|\).
• Select \( \alpha|S| \) for \( S_{1,2} \) from \( S_{1,2} \) and the remaining \( \omega = (1-\alpha)|S| \) keys from key pool \(|S|\) an remove the selected \( \omega \) keys from \(|S|\).
• Select \( \alpha|S| \) keys for \( S_{2,1} \) from \( S_{1,2}, S_{2,2}, S_{3,1} \) and select \( \beta |S| \) from \( S_{1,1} \) and \( S_{3,2} \). Finally select and remove \( \omega \) keys from key pool \(|S|\).

Figure 1 Shows that, how to select the keys for the key pool \( S_{2,1} \). To achieve perfect resilience same key should not be shared between the groups, for that also rules should be established i.e. if group G1 select keys from G2, no other
group should not select the same key. Similarly the overlapping factor $\alpha$ and $\beta$ should be assumed properly. For example $\alpha = 0.25$ and $\beta = 0$; only horizontal and vertical pools will share the key. Performance analysis of this scheme shows that good in storage, connectivity and resilience factors. But the disadvantage is its complexity.

Extending matrix based schemes: Yu et al proposed deployment knowledge scheme (Yu and Guan, 2008) with Blom’s scheme (Blom, 1985) where N sensors nodes are divided into groups. As Blom scheme each sensor in the group is loaded with seed to generate the column of G matrix and one row from Di matrix so nodes within the group will establish a key. To provide inter group communication each node also store t rows from Di matrix such that at least there will be one common D_i matrix between the neighbors to establish a key. The advantage of this scheme is that key connectivity is achieved. As Blom’s scheme, computation overhead is more i.e. to generate the column G using seed and multiplication to get the key. Similarly storage is $t \times (\lambda + 1)$ elements where t represents the number of rows of D_i matrix, $\lambda + 1$ represent the elements in each column of D_i matrix.

Group keying: In group keying, there will be clusters called group. Each group will share common group key for communication. For group keying four parameters to be consider n-number of sensor nodes, k-number of keys loaded into sensor nodes, m-rekeying message and s-session key. In group communication, all nodes will use single session key for communication. All data are encrypted using session which is common to all nodes. So this allow data aggregation and sometimes dissemination if duplication. Similarly the value of k and m should depends upon the situation i.e. if network size is large m should be increased because the number of keys stored will be increased and if the sensor eviction rate is high, the N should be low. k administrative keys are loaded into sensor nodes, which are used for rekeying when nodes get compromised or when new nodes get added. Sometimes individual keys are used for authentication and for initialization when new user joined the group. For group keying, the two approaches widely used are Tree based and EBS.

<table>
<thead>
<tr>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
<th>$k_4$</th>
<th>$k_5$</th>
<th>$k_6$</th>
<th>$k_7$</th>
<th>$k_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 3: Canonical matrix (5, 3)

Tree based approach: Tree based Approach: In tree based approach (Eltoweissy et al., 2004), for k-ary trees the number of keys stored in each node will be the height of the tree i.e. $\log_2n$. For binary tree it is $\log_2n$ keys. Compared to all trees, binary tree will have less rekeying message. Figure 2 Shows a binary tree structure.

User1 knows $\{K_{11}, K_{12} & K_{13}\}$. If suppose if user 1 evicts, keys $K_{12}, K_{12} & K_{13}$ have to be changed. So the number of rekeying messages required will be three. As a result, if the height of the tree get increased i.e. number of user increased, the number of keys stored will also be increased. Simultaneously rekeying message also increased. This leads to communication overhead and storage also an issue.

EBS-exclusion basis system: Eltoweissy et al. (2004) proposed a new key management technique called EBS (Exclusion Basis System) (Eltoweissy et al., 2004) which gives optimal result for the parameters n, k, m when compared to tree based approach.

Definition: Choose the value of n, k and m such a way that $k=1$ and $n=m$. An exclusion basis system denoted by EBS (n, k, m) is a collection of subsets of (1, n) = {1,2…n} such that for every integer $t \in (1, n)$ the following two properties hold.

Property (a): $t$ is in at most k subsets of $\Gamma$ and $U_m = 1 \ i A_i = (1,n)-\{t\}$

Property (b): There are exactly m subsets say $A_1, A_2, ..., A_m$ in $\Gamma$ such that $U_m = 1 \ i A_i = (1,n)-\{t\}$.

Consider an example EBS (10, 3, 2); The canonical matrix for C (5,3) is shown in Fig. 3.

Subsets are ; $A_1 = \{1,2,3,4,5,6\}; A_2 = \{1,2,3,7,8,9\}; A_3 = \{1,4,5,7,8,10\}$

$A_4 = \{2,4,6,7,9,10\}; A_5 = \{3,5,6,8,9,10\}$

According to first property since k = 3 each element t is in at most k subsets for example the element 1 is in $\{A_1, A_2, A_3\}$ and 10 is in $\{A_3, A_4, A_5\}$ etc.

According to second property each element is excluded by union two subsets
In the above example, since k is 3, each user will have 3 administrative keys i.e. user 1 will have \{k_1, k_2, k_3\}, for user 2 \{k_1, k_2, k_4\}, etc.,

Use of administrative keys: As explained earlier administrative keys are used for rekeying i.e., when user gets compromised / added.

For example when user 1 is evicted in EBS (10, 3, 2) the steps to be taken are: User 1 knows the key \{k_1, k_2, k_3\}, so these keys have to be changed.

The steps are Generate the new key \(K_1', K_2', K_3'\) and encrypted by former key \(K_1, K_2, K_3\).

Generate the new session key \(S'\), since the rekeying message \(m = 2\) by sending two message the previous key can be changed. i.e, \(K_6(S', K_1(K_1'), K_2(K_2'), K_3(K_3'))\) & \(K_5(S', K_1(K_1'), K_2(K_2'), K_3(K_3'))\)

Construction of EBS: To construct EBS \((n,k,m)\), canonical enumeration has to be found i.e. combination of forming subsets of \(K\) objects from a set of \(k+m\) objects. Positive solutions to EBS \((n,k,m)\) can be obtained if and only if \(C(k+m,k) \geq n\). Though the number of keys stored in binary tree is greater than the EBS (Eltoweissy et al., 2004), collision is not possible in binary tree. This is the major drawback in EBS. For example in EBS (10, 3, 2). User 1 and user 6 get compromised the entire key set will get revealed.

SHELL: Based on this, Younis et al. (2006) proposed a location aware combinatorial Key management scheme for clustered sensor nodes called SHELL (Younis et al., 2006), Which incorporates the features of EBS (Eltoweissy et al., 2004). Shell Exclusion base system has been designed for secure group communication for wired and adhoc networks. SHELL is a collision resistant scheme. The major components involved in SHELL are command node, Gateway, Key generating Gateways and sensor nodes. Command node or Base station communicates with gateways and is capable to detect compromised gateway. Gateway can communicate with other Gateway and also with the sensor node within its cluster. Gateway is capable to detect compromised sensor node. Gateway will form the EBS matrix, but actual keys will be generated by KGN. This is because if gateway gets compromised nothing will get revealed. In a cluster, there will be more than two KGN. The administrative keys will be loaded into sensor node from all KGN. The major problem in EBS Scheme is collision as explained above.

\[ A_1UA_2 = (1, 10)\{-10\} \]

\[ A_1UA_3 = (1, 10)\{-3\} \]

\[ A_1UA_5 = (1, 10)\{-1\} \]
In general if the hamming distance is less, the keys revealed will be less. Therefore before deploying the sensor node, Hamming Distance has to be calculated. But it leads to complexity.

**LOCK:** Based on EBS, Eltoweissy et al. (2004) proposed a dynamic key management scheme called Localized Combinatorial Keying (LOCK) (Eltoweissy et al., 2004). In this scheme administrative keys and session keys have to be changed periodically to achieve resilience, failures and topology changes. Though SHELL (Younis et al., 2006) is collision resistant rekeying is fully depend on centralized key server and also location dependent. But LOCK is post deployment i.e. information about the location of nodes are not required. Architecture of LOCK is similar to SHELL i.e. three tiers with common node, cluster head and sensor nodes. But the main difference is communication mechanism between command node and cluster head as shown in Fig. 5. Two layers of EBS (Eltoweissy et al., 2004) keys are constructed. First layer between command node and cluster head, and second layer between cluster head and sensor nodes. But in SHELL, pair-wise scheme is incorporated between command node and cluster head. During Initialization phase some backup keys are loaded into sensor nodes for authentication when cluster head get compromised. SHELL fails to provide authentication. When comparing LOCK with static schemes, the number of keys loaded into sensor nodes is less. The use of key polynomials in LOCK is to achieve resilience avoid location based information.

**LEAP:** LEAP (Localized Encryption and Authentication Protocol), protocol is designed for heterogeneous networks. LEAP (Zhu et al., 2003) support all types of communication pattern i.e. Network, Pair-wise and Group keying. In addition to that it supports the features like in network processing, passive participation and data fusion. In WSN since sensor nodes are closely deployed, possibility of sensing the same information will be more. If same value is send by an individual node, the communication cost and redundancy will be more. So network life time is reduced. To avoid these problems LEAP (Zhu et al., 2003) come out with the features of data fusion. i.e. it will gather the information then in networks processing. It means processing on gathered information before transmitting. It support basic requirements such as confidentiality, authentication and it avoids attacks like warm-hole and Sybil. For communication pattern, LEAP uses four different keys. These are Individual key, pair-wise shared key, cluster key and group key.

**Individual key:** Individual Key $K_{B_i}$ is used to communicate between base station and sensor nodes. This key is loaded into sensor node before deployment. The base station will not store all the individual keys of the node. When base station wants to communicate with sensor node, it generates the key using Pseudorandom function $f$ and its master key i.e. $K_{BS} = f(K_{hi})$.

**Pair-wise shared key:** This key is shared between node and its neighboring node. It is used for sending the cluster key and data to the nearby data fusion node. Sensor node will establish a pair-wise key with neighboring nodes after deployment. To establish a pair-wise key, the four stages are followed. These are key pre-distribution, neighbor discovery, pair-wise key establishment and key erasure.

Initially all sensor nodes are loaded with key $K_i$ and from that node will derive the master key $K_m$. If node $U$ wants to discover its neighbors, first it will initialize the timer because the sender node has to identify its neighbor before $t_{max}$; else the attacker will compromise the network. Then node $U$ broadcast HELLO message with its identity.

Node $U$ computes the pair-wise key $K_{UV}$ using node V’s identity. Node V will also follow the same procedure. In the fourth phase node $U$ erase $k_1$ and all the keys of neighboring node after $t$ minimum.

**Cluster key:** As LEAP (Zhu et al., 2003) support data aggregation, some node will have the responsibility to decide while forwarding the data to avoid duplication and thus communication overhead can be reduced. For that neighboring node will share the same key which is called cluster key, this will be distributed using pair-wise key.

**Group key:** Base station shares this key with all nodes in the network to send commands. This key either pre loaded or it can be distributed using cluster key. Proper rekeying mechanism should be handled while changing the group key.

The advantage of LEAP is support all four types of communication pattern, provide authentication and avoid attacks like Sybil, Sink hole etc. Because of it’s, inter-network processing, network lifetime can be increased. The disadvantage of this scheme is computation and communication overhead increases when the density of the network increases and storage also an issue.

**SUMMARY AND DISCUSSION**

While designing KMS for WSN the metric to be evaluated against KMS are key connectivity, resilience, efficiency and scalability. In general achieving all metrics in a single key management scheme is difficult. Based on keying mechanism, metrics can be evaluated i.e. for pair-wise scheme resilience and key connectivity and for group keying scalability can be measured. Efficiency plays a vital role for both schemes. Other than that other metrics like deployment knowledge, node authentication can also be considered. Table 2 shows the summary of all
Table 1: DWSN Vs HWSN

<table>
<thead>
<tr>
<th>Distributed WSN</th>
<th>Hierarchical WSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>EG Scheme, Q Composite, Multipath Key Reinforcement Scheme, Random Pair-wise Key Scheme, Key redistribution scheme.</td>
<td>AP Scheme, LU Matrix and hierarchical LU matrix, EBS: SHELL, LOCK and LEAP.</td>
</tr>
<tr>
<td>Matrix Based: Blom’s scheme, multi space key distribution scheme</td>
<td>Group Keying: Tree based approach,</td>
</tr>
<tr>
<td>Polynomial KMS: Blundo, Polynomial pool based key pre-distribution scheme</td>
<td></td>
</tr>
<tr>
<td>Deployment Knowledge: Extending Probabilistic scheme, Extending matrix based scheme</td>
<td></td>
</tr>
</tbody>
</table>

Each scheme will have both advantage and disadvantage while considering basic scheme though it is simple, efficient and achieve optimal resilience but fails to provide authentication. Basic scheme have been modified as Q-composite scheme to achieve better resilience. Random pair-wise scheme have been proposed to avoid memory usage but the drawback is, it cannot be extended not more than n-m nodes. To achieve better resilience in basic scheme multipath key reinforcement scheme is proposed. In that instead of using single path multi path is used for key redistribution. To avoid path key establishment in basic scheme, key redistribution scheme have been proposed. All the above methodology concentrates on key connectivity and resilience. Node revocation is also another issue in the above scheme; which will be handled by the controller node. Node authentication is also an issue. In matrix based scheme there will be 100% key connectivity. It provides node authentication also. But the problem with this scheme is scalability. If deployment knowledge is incorporated in matrix based scheme, it can be implemented for group communication to achieve inter and intra cluster communication. But matrix schemes are λ secure properties i.e. if λ nodes get compromised the entire network will get revealed. Polynomial pool based schemes achieves scalability but also λ secure property. All above mentioned schemes are perfect for node to node communication.

Table 2: Summary: Key management schemes

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Key connectivity (%)</th>
<th>Authentication</th>
<th>Scalability</th>
<th>Resilience-resistance against node capture</th>
<th>Deployment knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full pair-wise 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>E-G scheme 90</td>
<td></td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>K /</td>
<td>P</td>
</tr>
<tr>
<td>Probabilistic 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td></td>
<td>q</td>
</tr>
<tr>
<td>Multipath Key Pre-distribution 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Random pair-wise AP scheme 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Blom’s 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Du et al. 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Matrix based LU hierarchical 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>LU (19) 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Polynomial EBS 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>SHELL 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>LOCK 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>LEAP 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Group Extended basic 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Deployment scheme Extended basic 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
<tr>
<td>Extended matrix based Extended polynomial 100</td>
<td>Yes</td>
<td>Achieve 100 after path key establishment phase</td>
<td>Difficult</td>
<td>100%</td>
<td>No</td>
</tr>
</tbody>
</table>
communication. But Blundo scheme is used for secure group communication also.

For group communication, Exclusion Basis Scheme has been designed. But in EBS collision will occurs. To avoid collision, SHELL has been proposed but it must need deployment knowledge. SHELL fails in authentication. Based on SHELL, LOCK has been proposed but this scheme is post deployment and provides authentication. LEAP supports both node to node communication and group communication. But storage is an issue because it has to store large key for achieving all communication pattern. The comparison is made between different key management scheme in distributed and hierarchical. The Table 1 shows summarize the list of DWSN and HWSN.

CONCLUSION

Security plays a vital role in wireless sensor networks. Key management is an essential security issue in wireless sensor networks. Implementing key management schemes along with cryptographic algorithms are complex because of sensor node constraints. In this paper a review of key management schemes both in Distributed wireless sensor networks and Hierarchical wireless sensor networks have been done based on keying mechanisms, such as network, pairwise and group keying. In various key management schemes different evaluation metrics are discussed and it has been summarized in Table 2. Finally from key management scheme are concentrating on some evaluation metrics rather than all.

REFERENCES


An Efficient and Hybrid Key Management Scheme for Three Tier Wireless Sensor Networks Using LU Matrix

Manivannan Doraipandian, Ezhilarasie Rajapackiyam, P. Neelamegam, and Anuj Kumar Rai

School of Computing, SASTRA University, Thanjavur- India
dmv@cse.sastra.edu, crezhil@yahoo.com,
neelkeer@eie.sastra.edu, anujlucky007@gmail.com

Abstract. Security plays a vital role in Wireless Sensor Networks and attention on key part is essential to encrypt the information processing among the sensor nodes in the network. Due to sensor node constraint, Key Management plays an important role. Existing Key Management Schemes (KMS) for Cluster-based architecture either supports Group communication or Node-to-Node communication. The proposed hybrid KMS concentrates on both Group communication and Node-to-Node communication using LU matrix and to enhance the strength of the security between cluster head and base station the ElGamal Public key encryption techniques is used. The main feature of this proposed protocol is 100% Node-to-Node connectivity and perfect resilience is achieved when Sensor node/Cluster Head is compromised. The scheme and its detailed performance analysis are discussed in this paper.

Keywords: LU matrix, Wireless Sensor Network, Group communication, Node-to-Node connectivity.

1 Introduction

Sensor network is a collection of sensor nodes generally referred as motes and each node has different capability in terms of memory, processing, and transmission range. There are many applications based on wireless sensor network such as Health monitoring, Industrial Automation, Military application, Area Monitoring etc., where sensor network plays a very important role. Since sensor nodes are deployed in unfriendly area, security becomes a major issue in WSN. Before any exchange of data, encryption key must be known. Key Management plays a significant role in Security. Many Key Management schemes are available, but due to sensor node constraints they are not effective.

Various Keying mechanisms: Key Pre-distribution schemes, Self-enforced schemes and arbitrated schemes. But Key Pre-distribution keying mechanism is accepted widely because of motes constraints. For communication either Group keying for group communication or Pair-wise keying for Node-to-Node communication are used. Communication Patterns are selected based on the Architecture of Sensor Network and the types of applications.
In proposed three-tier architecture, in between base station and cluster head, public key cryptography is used and between cluster head and sensor node, Symmetric key cryptography is used. While designing Key Management protocol, the metrics to be evaluated are:

**Key Connectivity:** Each node communicates with every other node in the clustered region.

**Resilience:** When node gets compromised, how secure the remaining communication links, i.e., resistance against node capture

**Scalability:** Capability to support when large number of nodes or cluster is added to the network.

**Efficiency:** In terms of storage, communication and computation.

## 2 Related Works

KMS are mostly based on key pre-distribution schemes i.e., keys will be distributed to Sensor nodes before deployment. A Key Pre-distribution scheme is widely classified into two classes. 1. Random key Pre-distribution 2. Deterministic Approach.

Many Random Key Pre-distribution schemes are available. The E-G scheme [1] is the basic scheme, in which large key pool $(2^{20})$ is generated $P$. From the key pool $P$, randomly draw $m$ keys and assign to Sensor nodes. Then each sensor node broadcasts the Key ID’s of $m$ key to find a common key and make use of that key as shared secret key. If any pair of nodes does not share any key, it will communicate through secure links. Q- Composite scheme [2] is the improvement of E-G scheme in which instead of one shared key between nodes, $q (>1)$ keys are needed for Communication. Advantage of this scheme when compared to E-G scheme is, resilience will be more.

Matrix Based scheme is proposed by Blom [3], to establish a pairwise key between any pair of nodes. Blom’s scheme have two matrices one public $(\lambda +1)^n$ matrix $G$ and another $(\lambda +1)^n(\lambda +1)$ symmetric matrix $D$. These two matrices are used to calculate $n^\lambda(\lambda +1)$ matrix, $A = (D.G)^T$. Matrix A should be secret and in each sensor node one row from matrix A will be stored. Since G matrix is based on Vandermonde matrix, no need to broadcast the entire column, just by broadcasting the column id / seed, the sensor node is capable to compute the matrix. Though it provides perfect connectivity, scalability and resilience will be an issue. If $\lambda$ nodes get captured, the entire network will get captured. To improve the Blom’s scheme, Du et al. [4] proposed a pairwise key pre-distribution schemes. In that instead of one key space, multiple key spaces are used.

To increase the resilience, Liu and Ling [5] proposed a Polynomial pool based key pre-distribution scheme, which incorporated the idea of Blundo’s scheme [10] and Random Key Pre-distribution [1]. In all the above schemes, when numbers of compromising nodes reach to a certain threshold $\lambda$, the entire network will get capture.

Deterministic approaches use combinatorial designs i.e., grouping of certain elements based on certain properties. For Group communication EBS (Exclusion Basis System) scheme [11] has been used. SHELL [6] incorporated EBS. Though
number of Re-keying message is less, collision of nodes (k<m) will reveal all the administrative keys. Tree based approach is also used for group communication. If one node gets compromised, the cluster key gets revealed. So in the case of group communication, the cluster key should be changed periodically. For Key refreshing, i.e., updating a new key periodically, energy consumption will be more. So network lifetime will get reduced.

Achieving all metrics in a single key management protocol is difficult. For group communication since changes takes place periodically, number of Re-keying message and scalability can be evaluated. In Node-to-Node communication, metrics to be evaluated are Resilience and Key connectivity. To increase the network lifetime, node-to-node connectivity i.e., pair wise schemes can be used.

2.1 Our Contributions

In proposed scheme, LU matrix is introduced for Group communication and Node-to-Node communication whereas PKI is used for communication between Cluster head and Base Station. The important assumption in proposed scheme is that the base station will never get compromised.

In Proposed scheme:

1. Node-to-Node Communication within Cluster is achieved by using LU matrix. Any pair exchanges their row with each other to get secret key. ElGamal Public key encryption scheme is used for secure communication between Cluster head and Base Station.
2. Divide the cluster into sensor groups for group communication. Communication between group and cluster head can be established using LU matrix.
3. Group Key Manager (GKM) node is introduced to reduce the overhead of cluster head for refreshing the Group key.
4. Perfect Resilience and Full network Connectivity is achieved.
5. Performs well in terms of Storage and Scalability.

2.2 Preliminaries

2.2.1 LU Matrix

Lower Triangular matrix is formed from randomly generated key pool. Upper Triangular matrix can be constructed based on assumption that product of L and U will yield symmetric matrix. For example consider 3*3 matrixes

\[
\begin{bmatrix}
l_{11} & 0 & 0 \\
l_{21} & l_{22} & 0 \\
l_{31} & l_{32} & l_{33}
\end{bmatrix}
\times
\begin{bmatrix}
u_{11} & u_{12} & u_{13} \\
u_{21} & u_{22} & u_{23} \\
u_{31} & u_{32} & u_{33}
\end{bmatrix}
=\begin{bmatrix}k_{11} & k_{12} & k_{13} \\
k_{21} & k_{22} & k_{23} \\
k_{31} & k_{32} & k_{33}\end{bmatrix}
\]

From above the value of each K element can be calculated

\[
\begin{align*}
l_{11} & \times u_{11} = k_{11} \\
l_{11} & \times u_{12} = k_{12} \\
l_{11} & \times u_{13} = k_{13} \\
l_{21} & \times u_{11} = k_{21} \\
l_{31} & \times u_{11} = k_{31} \\
l_{21} & \times u_{12} + l_{22} \times u_{22} = k_{22} \\
l_{21} & \times u_{13} + l_{22} \times u_{23} = k_{23} \\
l_{31} & \times u_{12} + l_{32} \times u_{22} = k_{32} \\
l_{31} & \times u_{13} + l_{32} \times u_{23} + l_{33} \times u_{33} = k_{33}
\end{align*}
\]
Since $K$ is symmetric, $K_{ij}$ will be equal to $K_{ji}$

\[
    k_{12} = k_{21} = l_{11}.u_{12} = l_{21}.u_{11} \Rightarrow u_{12} = (l_{21}.u_{11}) / l_{11};
    k_{13} = k_{31} = l_{11}.u_{13} = l_{31}.u_{11} \Rightarrow u_{13} = (l_{31}.u_{11}) / l_{11};
    k_{23} = k_{32} = l_{21}.u_{13} + l_{22}.u_{23} = (l_{31}.u_{12} + l_{32}.u_{22}) / l_{12}
\]

By using these equations, randomly set the values for $u_{11}, u_{22}, u_{33}$ based on that calculate the values for remaining unknown variables of $U$ matrix so that their product would be symmetric.

### 2.2.2 ElGamal Scheme

Between Cluster head and base station, ElGamal PKI scheme is used because communication between cluster head and base station will have to be more secure than sensor node communication. Communication between base station and cluster heads are infrequent; but the level of secrecy is more and Communications between cluster heads and nodes and among nodes is frequent but the level of secrecy is normal than the former. In regular PKI scheme there is a possibility of Man-in-the-Middle sAttack due to its broadcasting nature. In this proposed scheme ElGamal Public key is not broadcasted, so there is no possibility of Man-in-the-Middle attack.

### 2.3 Notations Used

- $m$: number of Cluster heads.
- $n$: number of sensor node in each Cluster.
- nodeId: unique id for each node in one Cluster.
- groupId: unique id for each group in Cluster.
- $N_{i,j}$: $j^{th}$ node in $i^{th}$ Cluster.
- $n_{Li}$: lower triangular matrix for generating Symmetric matrix for $i^{th}$ Cluster.
- $g_{Li}$: lower triangular matrix for generating Symmetric Matrix.
- $n_{Ui}$: upper triangular matrix for generating Symmetric Matrix for $i^{th}$ Cluster.
- $g_{Ui}$: upper triangular matrix for generating Symmetric Matrix.
- $C_{i}$: Cluster head of $i^{th}$ Cluster.
- $l_{a_{i,j}}$: a row in $n_{Li}$ matrix in Cluster $i$ for group $j$.
- $u_{a_{i,j}}$: a Column in $g_{Ui}$ matrix in Cluster $i$ assign to sensor node $j$.
- $l_{a_{i,j}}$: a row from $n_{Li}$ matrix assigned to Cluster Head $C_{i}$.
- $u_{a_{i,j}}$: a column from $n_{Li}$ matrix assigned to Cluster Head $C_{i}$.
- $l_{g_{C_{i}}}$: a row from $g_{Li}$ matrix assigned to Cluster Head $C_{i}$.
- $u_{g_{C_{i}}}$: a column from $g_{Li}$ matrix assigned to Cluster Head $C_{i}$.
- $GKMi_{j}$: $j^{th}$ key group manager of $i^{th}$ Cluster.
- $K_{i,j,k}$: key between $N_{i,j}$ and $N_{i,k}$ in Cluster $i$ calculated by node $k$.
- $K_{GKM_{i,j}}$: Key between GKM and node $j$.
- $K_{GKM_{i,j}}$: Key Between group $j$ and cluster head $C_{i}$.
- $P_{GKM_{i,j}}$: Public key of cluster head $C_{i}$.
- $P_{ab}$: Public key of base station.
- $P_{ch}$: Private key of Cluster head.
- $P_{bs}$: Private key of base station.
- $P_{GKM_{i,j}}$: Public key of $j^{th}$ Group Key Manager of $i^{th}$ Cluster.
- $P_{GKM_{i,j}}$: Private key of $j^{th}$ Group Key Manager of $i^{th}$ Cluster.

### 3 WSN Architecture

The proposed architecture comprises of Sensor nodes, Base Station, Cluster Head, and Group Key Manager. Since the architecture is a cluster based, base station communicates with all Cluster heads. Inside the Cluster there will be sensor groups consisting of some number of nodes. Sensor groups are formed for group communication, in which sending a single message encrypted with group key rather than sending individual message to each and every node and mainly used for commanding the nodes. Transfer of data from sensor nodes to Cluster head is through shared secret key. For refreshing the group key, Group Key Manager is used which updates the key. There will be more than one GKM in Cluster, like GKM$_{1}$ take care of Group 1, 2, 3 and GKM$_{2}$ takes care of Group 4, 5, 6 likewise.
**Base Station**: Base station is assumed to be highly secure with high processing speed and memory. The entire $n_{Li}$, $n_{Ui}$, $g_{Li}$, and $g_{Ui}$ matrix (for $i=1$ to $m$) of all the clusters, one public and one private key for communication between the Cluster Head and Base Station and public key of all the Cluster Heads are saved in the Base Station.

**Cluster Head**: It is assumed to have more resources than sensor node in terms of transmission range, memory, and processing speed. Each cluster head saves one public, and one private key for communication with Base Station. The public key of base station and public keys of Group Key managers of its cluster are saved in the cluster head which also saves one row from $n_{Li}$ and one column from $n_{Ui}$ matrix for node to node connectivity, another one row of $g_{Li}$ and one column of $g_{Ui}$ matrix for Group communication between cluster Head and group members. It is assumed that Cluster head is able to reach all the nodes within a cluster.

**Group Key Manager**: Group Key Manager is used for maintaining the group and refreshing the key, so that security in-group communication is maintained. GKM stores the information of which node belongs to which group and which nodes come under its maintenance. There will be more than one GKM in a cluster. GKM of cluster $i$ stores one row of $n_{Li}$ and one column of $n_{Ui}$ matrix so that it can unicast message to the node for refreshing key. One public, one private key and public key of cluster head are saved for communication with cluster Head. In the proposed three-tier architecture, within a cluster there will be sensor group as shown in the figure Fig 1.

![Fig. 1. WSN Architecture](image-url)
Sensor Node: Each sensor node has limited memory, less processing speed, small transmission range that can be vulnerable also. Each sensor node in Cluster i save one row from nL_i and one column from nU_i matrix which is used for node to node communication, and one row of gL_i and one column of gU_i matrix which is for group communication. Each node in the group has same row of gL_i and column of gU_i matrix, so that they all can have the same key with the cluster Head.

4 Proposed Scheme

Phase I: Key Pre-distribution

Step 1: The base station first randomly generates a large pool of numbers (2^{20}) and randomly picks number to construct nL_i matrix and gL_i matrix for all m clusters.

Step 2: Calculate nU_i upper triangular matrix for each nL_i lower triangular matrix such that their product is symmetric matrix.

Step 3: A large number of public and private key pair is generated using Diffie-Hellman Protocol. Since this is pre-deployed scheme, all keys will be stored in the nodes before deployment, so possibility of Man-in-the-Middle attack is avoided

Step 4: For assigning sensor node in i^{th} cluster randomly select one row and column from nL_i and nU_i matrix and store it in sensor nodes.

Step 5: For assigning sensor node to group j in cluster i store U_{gU_{i,j}} and L_{gL_{i,j}} from gU_i and gL_i respectively which is common to all the nodes in the group j.

Step 6: Cluster head C_i, of i^{th} cluster is assigned a row and column from nL_i and nU_i respectively, for communicating with all the nodes within the cluster. For communicating with the Base station, a pair of private key P_{rci}, public key P_{uCi} is selected randomly and saved in the cluster head. Public key P_{uB} of Base Station and public Key of all the GKM are stored in the cluster head

Step 7: In Group Key Manager GKM_{i,j} one row and one column of nL_i, nU_i is randomly selected, a pair of public key P_{uGKM_{i,j}} and private P_{rGKM_{i,j}} from the pool is randomly selected and both are saved. Public key of cluster head P_{uci} is also loaded. Group key manager is also loaded with the table, which contains the nodeId and its groupId to which it belongs.

Phase II: Pair wise Key Establishment

A. In-between sensor nodes

After key deployment each sensor node needs to establish pairwise key with its neighbors and cluster head. To establish a pair wise key between N_{i,j} and N_{i,k} of cluster i, following steps are required:
Step 1: Node $N_{i,j}$ sends its row $L_{nLi,j}$ to node $N_{i,k}$

Step 2: Receiving $L_{nLi,j}$ from $N_{i,j}$, $N_{i,k}$ multiplies its column $U_{nUi,k}$ and gets $K_{i,jk}$ and replies with $(L_{nLi,k}, F(K_{i,jk}))$ and sends it to node $N_{i,j}$, where $F(K_{i,jk})$ is the hash value of key.

Step 3: Node $N_{i,j}$ receives $L_{nLi,k}$ from $N_{i,k}$, multiply it with its column $U_{nUi,j}$ and calculate $K_{i,kj}$ and computes the hash value of it and checks whether $F(K_{i,jk}) = F(K_{i,kj})$.

Step 4: If hash value of both are same then node $N_{i,j}$ sends $(ok, F(K_{i,kj}))$ to $N_{i,k}$ otherwise an error message $(err, Ni_k)$ is broadcasted. The error message will authenticate by the receiver using TESLA [12].

If the number of error message cross a threshold for $N_{i,k}$ then messages from $N_{i,k}$ is discarded by other member of cluster $i$.

**B In-Between Cluster Head and Base Station**

Encrypt and decrypt the messages shared between Cluster head and Base Station using pre-deployed public and private key.

**Phase III: Group Key in the Cluster**

Step 1: Cluster head $C_i$ sends an encrypted message $P_{uB}(\text{Group, groupId})$ to the Base station

Step 2: Base station decrypt the message by its private key $P_{rB}$. Base station sends the row $(P_{uCi}(L_{gLLi,j}))$ of group $j$ to $C_i$ encrypted by public key $P_{uCi}$ of cluster head.

Step 3: Cluster head decrypt the message by its private key $P_{rCi}$ and gets the row $L_{gLLi,j}$ and it multiplies it with $U_{gLUci}$ gets group Key $K_{gi,j}$ and then broadcast group message in format (Group||groupId||LgLci||Kgi_j(Message))

Step 4: Upon receiving the message every node checks whether the group id in the message is same as its groupId then it multiplies received Cluster’s group row $L_{gLLi}$ with $U_{gUi,j}$ and gets the key $K_{gi,j}$ and use this key to decrypt the group message.

Step 5: If the groupId is not same as nodes groupId message is discarded.

**Phase IV: Key Refreshing by Group Key Manager**

Step 1: Base Station sends a encrypted message $P_{uCi}(\text{KeyRef}||\text{groupId}||L_{gLLi}||U_{gLU})$ to the Cluster head $C_i$ for refreshing $L_{gLLi}$ & $U_{gLU}$ of Group members having GroupId.

Step 2: Cluster Head decrypts the message using $P_{rCi}$. 

Step 3: Cluster head encrypts the message with public key of Group key Manager \( Pu_{GKM,j}(KeyRef||groupId||gL||Ug) \) and sends it to the Group Key Manager.

Step 4: Group Key Manager Decrypts the message with \( Pu_{GKM,j} \) and then it sends the new group row an column received to all nodes having groupId using Unicast message to all the node in the group.

It is the responsibility of the Group Key Manager not to send new row and column to the compromised sensor node, so that new Group key is not revealed to the compromised node. The Group Key manager maintains the detail of compromised nodes that will be updated by the cluster Head.

5 Security Analysis

Assumption: Base station never get compromised and all information are stored on the Base station \( nL_i, nU_i, gL_i \) and \( gU_i \) matrix for \( 0 < i \leq m \). Security analysis shows better result than existing methodology when Node, Cluster head and Group Key Manager are compromised.

A. When Sensor Node gets compromised

One row and one column of \( nL_i, nU_i \) matrix will get revealed. This will not affect the communication between the remaining nodes within a cluster, because each link will use different key for Node-to-Node connectivity. In that case group key will get revealed, which won’t affect the rest of the sensor groups, only the group in which sensor node is present will get affected, that can also be solved by GKM. GKM will send new row and column to all the nodes in the group except the compromised node, by unicasting the message using individual key \( K_{j,GKM} \) between GKM and node.

B. When Cluster Head gets compromised

One row of \( nL_i \), one column of \( nU_i \), public and private key of Cluster head \( C_i \) will get revealed when \( C_i \) gets compromised, but it will not affect other communication between the nodes in a cluster. Since Public Key infrastructure communication is used between base station and cluster head, it won’t affect the remaining links between base station and cluster head. When compared with scheme [9], since LU matrix is used so there is a possibility of having same Keys between base station and Non-compromised cluster head. This is the added advantage of the proposed scheme.

C. When Group Key Manager gets compromised

None of the communication links will get affected. But it can be able to change the group key. But this also can be overcome by following the procedure to check whether GKM has changed group key of nodes or not:
1. Cluster head sends nonce encrypted by using previous group key and broadcast the message.
2. Sensor nodes within sensor groups will decrypt the message and again gets encrypted by individual key thereby sends that message to Cluster head.
3. If the Cluster head receives the same message, just replace the GKM as it is compromised; else replace the GKM and change Group row and column of each group.

For the security purpose, the key should be changed periodically, which can be done by refreshing the row and column of nodes using previous key.

6 Performance Analysis

A. Node-to-Node Connectivity

Compared with E-G scheme, proposed scheme has 100% node-to-node connectivity, since each node shares its row elements to get shared secret key of any other node. In E-G scheme, probability of sharing same key between two nodes, having Key pool P, and no of keys stored in each node is m, is given by,

$$\text{Probability} = 1 - \frac{((P-m)!)}{P!(P-2m)!}$$

In Fig.2 for P=1000, if m=90 the probability of key sharing will be 0.99, if m=40 it will be 0.811 but in proposed scheme connectivity does not depend on the number of keys stored in node and the probability of connectivity will be 1.

![Fig. 2. Key Connectivity](image)

B. Scalability

When the number of nodes increases, proposed scheme provides good scalability than other protocols without compromising the security. When nodes are added in the scheme using LU matrix, either size of matrix should be changed or randomly pick...
row and column and assign to nodes. In the scheme given in [8] when new cluster is introduced, size of the matrix should be changed so computation and communication cost is increased. In proposed protocol, all the clusters are having different matrix for node-to-node connectivity and group communication, and different private and public key for communicating with the Base station. So we can introduce a new cluster with new set of $nL_i$, $nU_i$, $gL_i$, $gU_i$ matrix, and public and private key, which reduce computation and communication cost, without compromising the security.

C. Resilience

Perfect resiliency is achieved. In scheme [8] when number of cluster head compromising increases, possibility of compromising communication links will be increases. While randomly picking up keys, possibility of assigning same key will get increases because there will be single LU matrix for cluster head and base station communication. But in proposed scheme since PKI is incorporated, compromising cluster head will not affect the remaining communication links between other cluster head and base station as shown in the Fig.3.

![Fig. 3. Cluster Head Compromised](image)

When a node is compromised it will not affect the communication between the non-compromised nodes up to certain threshold.

Conclusion

An efficient and Hybrid KMS for WSN is discussed. LU matrix will play a major role to achieve 100% Node-to-Node connectivity. In this architecture, secured ElGamal Public Key Cryptosystem is used between Cluster Head and Base Station and also between Cluster Head to GKM in order to improve the security. Re-keying is done using GKM. It is also good in Resilience and Scalability. Security analysis and Performance analysis show that this scheme is the most suitable one in real time active Wireless Sensor Networks applications.
References

An Efficient Authentication Protocol Based on Congruence for Wireless Sensor networks

D. Manivannan, B. Vijayalakshmi, P. Neelamegam
School of Computing, SASTRA University, Thanjavur, India.
dmv@cse.sastra.edu, bvijibalu@rediffmail.com, neelkeer@ele.sastra.edu

Abstract - Authentication in Wireless Sensor Networks (WSNs) is a challenging process. Providing authentication for the Nodes in WSN is a vital issue in Secure Group communication among WSNs. Massive group of tiny sensor Nodes form WSNs and these are placed in open, unattended milieu. Due to this reason, Nodes in WSN can endure exclusive encounters. WSNs are more vulnerable to active and passive attacks than wired ones due to their broadcasting nature, limitations in resources and unrestrained environments. However, security will be a significant factor for their complete implementation. In this paper, a new protocol based on congruence equations and number theory concepts have been introduced to achieve secure authentication among Nodes in WSNs. The proposed protocol uses Fermat Number Theorem (FNT) and Combinations of Chinese Remainder Theorem with Fermat Numbers (CRT_FN) to enhance the strength of authentication mechanism among the sensor nodes. In between Node to Node authentication FNT is used, Cluster head to Node and Base station to Cluster head CRT_FN is used. Comparison of the proposed protocol with existing protocols is done. It reflects that it provides instant authentication, minimum memory utilization and withstands DoS attack, cloning attack, replay attack, eavesdropping and Man-in-the-middle attack.

Keywords - Authentication, Congruence, Chinese Remainder Theorem, Fermat Number.

I. INTRODUCTION

The recent technological revolution has made a drastic change in data communication. The advent of laptops, mobile phones, and wireless sensor Nodes has also added value to it. This leads to the use of wireless communication anywhere needed. Out of all wireless based communications, wireless sensor Nodes are emerging rapidly and used in many applications like military, agriculture, health, traffic analysis etc. and in near future they will become part and parcel of human life[6][7]. In all WSN based applications, WSNs are projected to sense, process and amass some specific environmental features like temperature, pressure, humidity, etc and provide the sensed data to the data service that need it. In any data communication, authentication of entities involving in communication is very essential; since it alone ensures the original identity of each entity involving in the communication and it is very essential for the security of the networks. In WSN, authentication of Nodes plays a vital role.

Mainly three types of communication take place within a sensor network. First, a Node can relay its data to another Node, second, a Node can dispatch its data to Base station and the third is, the Base station can spread over its commands to the Nodes near the vicinity. All data from base station is sent to data service which needs that through some external network. All these are depicted in Fig. 1.

Since all these communications happen in open milieu or even in hostile environment, malicious packet injection is easy in these networks. Many passive attacks like eavesdropping, man-in-the-middle attack and passive attacks like replay attack, DoS attack, and cloning attack can easily be made in WSNs due to their wireless nature. The Nodes or the Base stations that receive the packets want to make sure that the packets they have received are really originated from the claimed source. A false data from a compromised Node may lead to false prediction by base station which leads to drastic result and the real purpose of WSN may be wasted. In other case, false commands from phony base station may lead to wrong working of all Nodes within the network which may also affect the actual principle of WSNs. All these abnormalities can be detected with the help of only an authentication protocol. An authentication protocol can enable the receivers to confirm that the packet was truly sent by the claimed sender. Thus in open milieu it is indispensable to enable sensor Nodes to authenticate broadcast messages/commands from base station and enable base station to receive authenticated data from sensor Nodes.
Simply deploying symmetric schemes and hash functions based authentication mechanisms are not suitable for WSNs. The main reason is, any receiver with secret key can counterfeit data and mimic the sender. As a result, it is natural to seek solutions based on asymmetric cryptography to prevent this attack. Traditional asymmetric cryptography based authentication techniques like public key based digital signatures are not suitable for resource constrained sensor Nodes because it has high overhead in terms of time requisite to sign and verify each packet and also bandwidth needed for this is more, which is impractical for WSNs.

The rest of the paper is as follows. The next section reviews the related works and basic requirements of authentication protocols for WSN. In section 3, few mathematical backgrounds needed for this protocol is explained. In section 4, the proposed protocol is explained along with the notations used. In section 5, the security and efficiency of the proposed protocol is analyzed with other protocols.

II. RELATED WORKS

Authenticated data transfer between Nodes and base station is one of the core challenges in WSNs and several schemes were proposed for that. Timed Efficient Stream Loss-tolerant Authentication (TESLA) [1] is a Broadcast Authentication protocol in which loose time synchronization between sender and receiver is required. Here one -way hash chains are used to obligate to a sequence of random values and asymmetry is forced by the postponed exposure of keys. Albeit, TESLA [1] removes computational overhead of digital signature based schemes, secure distribution of key chain and loose time synchronization between Nodes are still overhead. µTESLA and Multi-level µTESLA [2], [3] are variants of TESLA protocol. µTESLA divides the time period for broadcast into several intervals and separate keys are used for each interval. All packets broadcasted in a specific interval are authenticated with same key assigned to that particular time interval. Even though this protocol mitigates the problem of secure key chain distribution by unicasting the key chain, this still increases the communication cost. Despite of proposal of multiple multi-level µTESLA [3] schemes, scalability, prevention from DoS attacks and revocation of Nodes, still remains a problem. Practical implementation of these protocols is also complicated. Chinese Remainder Theorem-Based Broadcast Authentication in Wireless Sensor Networks (CRTBA) [4] was designed to overcome DoS attacks by providing instant authentication, but still this too use one way hash function. The main drawback of using one-way hash chain is at least minimum of 64-bit key chain is desirable for efficient result, but some Time- Memory -Data tradeoff techniques can be pertinent, that leads to discovery of future keys, this may endanger the secrecy of entire network[5] and this is not suitable for resource constraint sensor Nodes [8]. In CRTBA, each Node needs to store the two keys and seed value in addition. This is suitable only for fixed group of users, scalability remains problem because of predetermined key chain length for fixed users and it concentrates only on broadcast authentication. Most of the existing techniques concentrate on single entity authentication and one way communication authentication. Considering all these factors, to improve authentication and to minimize the limitations of existing techniques, to overcome attacks, to provide authentication on either side of communication and to improve the secrecy, a new congruence based authentication scheme is proposed. To overcome all these drawbacks, in the proposed scheme, Congruence equations are combined with Fermat Number theorem for Node to cluster head/ cluster head to base station authentication. Fermat Number sequences are used to authenticate Nodes, in Node to Node communication. Some of the requirements are considered as basic requirements of any authentication protocol for WSN and they are discussed in Table 1.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Confidentiality</td>
<td>Each Node’s private information must be kept secure to guarantee data privacy [9]</td>
</tr>
<tr>
<td>Mutual Authentication</td>
<td>Any two entities involving in message transfer must be sure about the original identity of each other. [9].</td>
</tr>
<tr>
<td>Anonymity</td>
<td>Information on the Node must be made anonymous.</td>
</tr>
<tr>
<td>Data Integrity</td>
<td>This property guarantee the receiver that the data received is not changed in transmit by an adversary. [9].</td>
</tr>
<tr>
<td>Instant authentication</td>
<td>Time taken to authenticate a request must be very less, and much time should be utilized for data transfer.</td>
</tr>
<tr>
<td>Data Freshness</td>
<td>Each data received from a Node is recent data and it ensures that no attacker relayed old data’s. [9].</td>
</tr>
</tbody>
</table>

III. BACKGROUND

In this section some mathematical primitives and notations used in this protocol is explained. In addition, the importance of the proposed scheme related with mathematical logics are discussed. In congruence, If x and y are any two integers and n is a positive integer that divides x-y, then it can be stated that, x is congruent to y modulo n and it can be denoted as \( x \equiv y \mod n \). This is called as congruence, and used instead of equality relation in cryptography. Here n is the modulus of the congruence and y is the residue of x (mod n). The result of y mod n is always a nonnegative integer less than n and in modular arithmetic it is referred as set of least residues modulo n, or \( Z_n \). The congruence operator is mainly many - to – one mapping that maps infinite members of Z (set of integers) to one member of \( Z_n \).

In a congruence equation \( x \equiv y \mod n \), even if y and n are known, precise x value cannot be found, because x will take

---

**TABLE I**

**BASIC REQUIREMENTS OF AUTHENTICATION PROTOCOLS**

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Confidentiality</td>
<td>Each Node’s private information must be kept secure to guarantee data privacy [9]</td>
</tr>
<tr>
<td>Mutual Authentication</td>
<td>Any two entities involving in message transfer must be sure about the original identity of each other. [9].</td>
</tr>
<tr>
<td>Anonymity</td>
<td>Information on the Node must be made anonymous.</td>
</tr>
<tr>
<td>Data Integrity</td>
<td>This property guarantee the receiver that the data received is not changed in transmit by an adversary. [9].</td>
</tr>
<tr>
<td>Instant authentication</td>
<td>Time taken to authenticate a request must be very less, and much time should be utilized for data transfer.</td>
</tr>
<tr>
<td>Data Freshness</td>
<td>Each data received from a Node is recent data and it ensures that no attacker relayed old data’s. [9].</td>
</tr>
</tbody>
</table>
the form \( nk + y \) and \( k \) can take the values 1, 2, 3,…. Hence it is very intricate to find the exact \( x \) value. The main property in congruence is, if \( a, b \in \mathbb{Z} \) with \( n>0 \), then \( a \equiv b \mod n \) is a linear modular equation or linear congruence equation over Z. With known \( b, n \) and unknown \( a \), there cannot be only one solution to this problem. If \( v \in \mathbb{Z} \), has the property \( av \equiv b \mod n \) then \( v \) is the solution, but then the integers of the form \( kv + n \), \( k \in \mathbb{Z} \) are also solutions and there can be infinite number of these solutions. This property can be stated as property 1, and can be used to enhance the security of the identity of each Node in WSN. For solving simultaneous congruence equation, Chinese Remainder Theorem (CRT) is used, and in CRT, given a system of congruence’s to different moduli:

\[
X \equiv a_1 \mod m_1,\\
X \equiv a_2 \mod m_2,\\
\ldots\\
X \equiv a_i \mod m_i.
\]

And if each pair of moduli are relatively prime, and if each pair of moduli are relatively prime, i.e.: \( \gcd (m_i, m_j) = 1 \) for \( i \neq j \), has exactly one common solution \( M = m_1*m_2*\ldots*m_i \) and any two solutions are congruent to one another modulo \( M \). For generating the sequence, which is used in the proposed protocol to identify the authentication of Nodes, the Fermat Numbers are used. Fermat Numbers are integers of the form \( F_n = 2^{2^n} + 1 \) where \( n \) is a non-negative integer. Most of the Fermat Numbers are prime numbers.

### IV. PROPOSED PROTOCOL

When the Base station receives any data from sensor Nodes, it must be sure that data is only from the claimed node and not from any attacker. Similarly the sensor Node must be sure about the true identity of base station from which command is received. Having these considerations, the proposed protocol proposes a method of combining congruence based technique with Fermat number theory for authenticating Node and cluster head in Node to cluster head and cluster head to base station communication respectively and usage of Fermat number theory with the circular convolution product of sequences for authenticating Nodes in Node to Node communication.

Even though many previous works use congruence as the base of their work CRTBA was the first scheme to use it in authentication procedure. The negative aspect of this scheme is that, each time when the equations are solved the original key is revealed in the network. This endangers the total secrecy of WSN. This drawback is overcome and more security is incorporated in this proposed protocol.

Following notations are used in this paper:

**Notations**

- \( KB_{pi}, KN_{pi} \): Relative prime numbers stored in base station and sensor Node respectively.
- \( KB_{ri}, KN_{ri} \): Residue of congruence equation in base station and sensor Node respectively.
- \( X_n \): k-bit random number.
- \( H_n \): k-bit random Number.
- \( X_n[ ] \): Sequence derived from \( X_n \).
- \( H_n[ ] \): Sequence derived from \( H_n \).
- \( FT \): Fermat Number.
- \( N \): Length of sequence \( X_n[ ] \) or \( H_n[ ] \).
- \( IFNT \): Inverse FNT.
- \( \mid \): Concatenation operator.
- \( H_s() \): Hash function.

**Explanation**

### A. Initialization Phase

A secret code is randomly generated using a random generator. Two different congruence relations for the secret code are constructed with the help of two relative prime numbers \( KB_{pi} \) and \( KN_{pi} \). The residues of each equation \( KB_{ri} \) and \( KN_{ri} \) respectively are calculated. Using property 1, Only \( KN_{pi} \) and \( KN_{ri} \) are stored in a sensor Node. From this it is clear that a Node capture won’t reveal the secret code. In the other side, the hash of the secret code, \( KB_{pi} \) and \( KB_{ri} \) are stored in base station. \( KN_{pi} \) and \( KN_{ri} \) are secret code of each sensor Node and it is unique to each Node and stored in each Node.

### B. Node to Cluster Head Authentication

In Node side, the authentication process follows these steps,

**Step 1:** Each sensor Node generates two, k-bit random number \( X_n \) and \( H_n \).

**Step 2:** From two random numbers generated, two sequence \( X_n[ ] \) and \( H_n[ ] \) of length \( N \) is produced respectively.

**Step 3:** The circulation convolution product \( Y_n[ ] \) of the two sequences \( X_n[ ] \) and \( H_n[ ] \), is calculated using:

\[
Y_n(n) = \sum_{m=0}^{n-1} H_n(m) X_n((n-m)mod N), \quad \text{for } n = 0 \text{ to } N-1 ,
\]

and this is considered as result 1.

**Step 4:** Each Node concatenates its id with hash of \( X_n \), \( Y_n \), the obtained the result \( Y_n[ ] \) (result1) and with the private secret stored in each Node \( KN_{pi} \) and \( KN_{ri} \) to the base station in the form, \( (Id)\mid Hs(Xn)\mid Hs(Yn)\mid result\mid Hs(KN_{pi}KN_{ri})\mid m) \) and this is sent to the cluster head as authentication request. To avoid replay attack, the \( X_n \), \( H_n \) and \( Hs \) can be stored in either side and each time along with new values, previous \( X_n \), \( H_n \) values are also attached. Each time, both the sides check whether previous values on both ends get matched, if so it is confirmed that data freshness is maintained or if not some intruder have resed the old packet and it can be rejected.

In the cluster head, the instant the request is received, authentication verification is done in the sequence of steps,
Step 1: The Id in the requested packet is checked with the list of Id’s id of active Nodes near its vicinity, if it doesn’t match, the request is rejected and negative acknowledgement is sent.
Step 2: If the Id gets matched, original Xn and Hn is obtained from Hs(Xn) and Hs(Yn) using reverse hash function.
Step 3: From Xn and Hn, the two sequence Xn[ ] and Hn[ ] are generated.
Step 4: The largest number in the two sequences is identified, and the number of bits (α) used to represent that largest number is calculated.
Step 5: Using alpha (α) value Fermat number Ft is calculated using Fermat Theorem \( Ft = 2^\alpha + 1 \).
Step 6: Using above values, \( \alpha \) is calculated as \( \alpha = \frac{N-1}{\alpha} \times \mod{Ft} \).
Step 7: From the result \( \alpha \), FNT of the sequence Xn[ ] is calculated as,
\[
FNT(Xn[\{a\}]) = \left[\sum_{n=0}^{N-1} Xn(n) \times \alpha^k \right] \mod{Ft} \quad \text{for} \quad k=0, 1, \ldots, N-1.
\]
Step 8: Likewise FNT of the sequence Hn[ ] is calculated as,
\[
FNT(Hn[\{a\}]) = \left[\sum_{n=0}^{N-1} Hn(n) \times \alpha^k \right] \mod{Ft} \quad \text{For} \quad k=0, 1, \ldots, N-1.
\]
Step 9: Point wise multiplication of FNT (Xn[ ]) and FNT (Hn []) is done and the sequence Xf[ ] is found.
Step 10: From the above result IFNT (Xf[ ]) is calculated as,
\[
IFNT(Xf[\{a\}]) = \left[\sum_{n=1}^{N-1} Xf(n) \times \alpha^k \right] \mod{Ft} \quad \text{For} \quad n=0, 1, N-1. \quad \text{The} \quad N^{-1} \quad \text{is found using the condition,} \quad NN^{-1} = 1 \quad \text{(mod} \quad M)\text{).}
\]
Step 11: The Result of IFNT (Xf[ ]) is considered as Result2.
Step 12: The Result 1 from the client is compared with the Result 2. If both are equal the secret identity KN pi and KN ri of Id’s id of active Nodes near its vicinity, if it doesn’t match, the request is rejected and negative acknowledgement is sent.
Step 13: The reverse Hash function is applied to the received packet to get I1 and I2. If these values matched with the shared secret value between the corresponding neighbors, then further process is done, if it doesn’t match, the request is rejected and negative acknowledgement is sent.

C. Node to Node Authentication

When data is transferred between Nodes, each Node must guarantee the authenticity of Nodes that transferred the data. This is essential since, even single false data when aggregated with other results can magnify the falsification of data which leads to false prediction by base station. To avoid this, before accepting each message from a Node, each Node must check the pure identity of the sender Node from which the data is received.

Before proceeding with this procedure, it is assumed that each Node shares two random k-bit integer (I1, I2) for only its neighbor from which it can get data. Since these values are stored only between neighbors and not for all Nodes this will occupy only negligible amount of memory.

In Node to Node authentication, when a Node wants to send data to other Node it sends message along with the authentication request,

Step 1: Each sensor Node generates two, k-bit random number Xn and Hn.
Step 2: From two random numbers generated, two sequence Xn[ ] and Hn[ ] of length N is produced respectively.
Step 3: The circulation convolution product Yn[ ] of the two sequences Xn[ ] and Hn[ ], is calculated using,
\[
Yn(n) = \sum_{m=0}^{N-1} Hn(m) Xn((n-m) \mod{N}, \quad \text{for} \quad n = 0 \to N - 1. \quad \text{and this is considered as result R1.}
\]
Step 4: Each Node concatenate Hash of (I1, I2), corresponding to particular neighbor with hash of Xn, Yn, the obtained result R1 and message m. (Hs(I1, I2) || Hs(Xn)||Hs(Yn)||R1||m) and this is sent to the other Node as authentication request.

The Node which receives the message along with the request, process the request in sequence of steps,

Step 1: The reverse Hash function is applied to the received packet to get I1 and I2. If these values matched with the shared secret value between the corresponding neighbors, then further process is done, if it doesn’t match, the request is rejected and negative acknowledgement is sent.
Step 2: Original Xn and Hn is obtained from Hs(Xn) and Hs(Yn) using reverse hash function.
Step 3: From Xn and Hn, the two sequence Xn[ ] and Hn[ ] are generated.
Step 4: The largest number in the two sequences is identified, and the number of bits (α) used to represent that largest number is calculated.
Step 5: Using alpha (α) value Fermat number Ft is calculated using Fermat Theorem \( Ft = 2^\alpha + 1 \).
Step 6: Using above values, \( \alpha \) is calculated as \( \alpha = \frac{N-1}{\alpha} \times \mod{Ft} \).
Step 7: From the result \( \alpha \), FNT of the sequence Xn[ ] is calculated as,
\[
FNT(Xn[\{a\}]) = \left[\sum_{n=0}^{N-1} Xn(n) \times \alpha^k \right] \mod{Ft} \quad \text{for} \quad k=0, 1, \ldots, N-1.
\]
Step 11: The Result of IFNT(Xf[]) is considered as R2.
Step 12: The Result R1 from the Node 1 is compared with the Result R2. If both are equal, it is clear that the data is not corrupted in the middle and it is from authorized Node and the message is accepted. The new Xn and Hn value plays the role of (I1, I2) for next communication. Since all the values are hashed, anonymity is maintained. The secret values between Nodes are also maintained. The receiver Node can either follow FNT procedure or even circular convolution product based on the procedure loaded in the Nodes.

For Broadcast authentication, the base station needs to combine FNT with CRT and concatenate Hash of secret code in addition. All Nodes check the initial values and perform the FNT-CRT procedure and final result is Hashed and compared with the received value, if both gets matched the command from base station is accepted as authentic.

V. SECURITY AND EFFICIENCY ANALYSIS

In this section, the security analysis and the comparison of the efficiency between the proposed protocol and the previous protocols are discussed,

A. Security analysis

The proposed protocol is analyzed to check whether it satisfies the requirements described in Table I.

| TABLE III
Comparision of basic requirements of authentication protocols with different schemes |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>µTESLA</td>
<td>CRTBA</td>
</tr>
<tr>
<td>Data confidentiality</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Instant Authentication</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>Anonymity</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>Data Integrity</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>Data Freshness</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

S – Satisfied, N – Not Satisfied

Confidentiality: Each message in the authentication procedure is protected against the illegitimate eavesdropper. In the proposed protocol, each message won’t provide any usable information since everything is the result of the hash function. Therefore, the proposed protocol satisfies the confidentiality.

Anonymity: The authentication process should not reveal any usable information about the node identity. In The proposed protocol, even the part of secret is captured, without the other half secret stored in base station, original secret cannot be found. Thus, anonymity is satisfied.

Mutual Authentication: The proposed protocol provide node to node authentication, node to cluster head authentication and cluster head to base station authentication. Each procedure provides different level of security with the multiple combinations of number theory concepts.

Instant Authentication: Authentication needs to be done as soon as request is provided without waiting for keys or buffer. The proposed protocol provides instant authentication, without need for any buffer or time synchronization between any two entities.

Data Integrity: The data transferred in the WSN must not be altered by the intruder, if it is done then it must be known to the receiver. In the proposed protocol if R1 matches with R2, then receiver can be sure about data integrity if not the receiver can neglect the message.

Data Freshness: Each time when a data is received the receiver must be sure that only recent data is received. In the proposed protocol, since each data is sent with k-bit random number and the previous number is also maintained, Data freshness and continuity between communications is maintained.

Few attacks that can occur in WSNs have been identified, and the proposed protocol is compared with some previous protocols, in the basis of resistance against attacks in WSN and that is shown in Table IV.

Eavesdropping: The adversary cannot get any constructive data through eavesdropping. As a result, the proposed protocol is secure against eavesdropping.

Replay attack: Each time a new random number is generated and attached along with the message. The intruder cannot request with some old message since it won’t match up with the previous random number which is maintained at each node. Thus, the replay attack is impossible.

Cloning attack: The attacker cannot guess the random number generated in each message. Moreover, he does not know the unique secret key of each Node that is shared with base station. Thus, the adversary cannot make the fake request.

DoS Attack: The instant request is received, authentication is done and since there is no need for buffer and no possibility of fake messages DoS attacks are impossible.

Man-in-the-middle attack: In the proposed protocol man-in-the-middle attack is impossible, because the attacker does not know unique secret. Therefore, he cannot generate the hash using the random number. Even if the Node is captured, from the part of secret stored in Node no useful information can be gained. This is clearly explained in property 1.

| TABLE IV
Comparision of various attacks with different authentication schemes |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacks</td>
<td>µTESLA</td>
<td>CRTBA</td>
</tr>
<tr>
<td>Eavesdropping</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>Cloning attack</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Replay attack</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>DoS attack</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>Man-in-the-middle attack</td>
<td>N</td>
<td>S</td>
</tr>
</tbody>
</table>

S – Secure, N - Not Secure
B. Efficiency Analysis

The proposed protocol is compared with the previous WSN authentication protocols in terms of memory required to store initial values required for authentication and the probability of obtaining secret from each Node. In each protocol certain values must be stored before deployment of sensors. In that case, the proposed protocol needs only two values to be stored in node memory for authentication procedure when compared with other protocols. This is shown in Fig. 2.

In CRTBA, congruence equation is constructed by combining, secret key of the session and MAC M. Since \( n_0, n_1 \) values are stored in each Node; this is similar to solving single congruence equation and it will have only one solution. Hence any Node capture can reveal the secret key immediately. This affects the security of the protocol, whereas in the proposed protocol only, one prime value and residue is stored. For this, multiple solutions exist (property 1); hence it is impractical to detect one correct solution from multiple solutions.

Fig. 2 Number of basic keys stored in Node memory for different schemes.

When the proposed protocol is compared with existing protocol it shows that it provide instant authentication, with minimum memory utilization and withstand many attacks.

This protocol is implemented in Java in windows Xp environment. Few Nodes and cluster heads are created and based on the proposed protocol, congruence values are stored and authenticity between Nodes and cluster Heads are verified. It shows that Nodes/ cluster heads are authentic for correct input values and Not authentic for other values. All the analysis is made based on this implementation.

VI. CONCLUSIONS

Based on the results, this protocol proves to be an efficient, secure and scalable protocol to establish authentication between Nodes, Node to cluster head and cluster head to base station. It is suitable for both static and dynamic WSNs. Any pair of Nodes can authenticate themselves in secure group communication and even if any new Node is introduced, just congruence calculations need to be calculated and loaded into Nodes. Hence this is scalable and with the use of Fermat number the security increases. After comparing with some well-liked and accepted authentication protocol, the proposed protocol is efficient against multiple attacks and probability of obtaining Node information is nearly zero, computation overhead is minimized and also limited memory is utilized.

REFERENCES

7. Elaine Shi and Adrian Perrig,“ Designing Secure Sensor Networks,” IEEE Wireless Communications, December 2004
10. Online: http://www.maths.gla.ac.uk/~wws
12. Sarad A.V aka Data “Applications to Chinese Remainder Theorem”.

554
A Secure Group Communication and Rekeying using Rabin’s Squaring Trapdoor Function in Multicasting

D. Manivannan  
School of Computing  
SASTRA University  
Tanjore, Tamil Nadu  
India  

A.R.Shloka  
School of Computing  
SASTRA University  
Tanjore, Tamil Nadu  
India  

P.Neelamegam  
School of Computing  
SASTRA University  
Tanjore, Tamil Nadu  
India  

ABSTRACT
In today’s world of internet, secure communication among group of members has become vital, where multicasting plays an important role. In secure multicasting, security and scalability are the two important checks. Sometimes there is tradeoff between security and scalability. In order to transmit the data in a secure and scalable way, a suitable key management protocol should be implemented which reduces the number of rekey messages generated during the join or leave of any member thereby preserving forward and backward secrecy. In our proposed work, a new key management protocol has been proposed where a hierarchical structure has been implemented to improve the scalability. Derivation key is used to generate the new keys from the existing keys and Rabin’s Squaring Trapdoor Function is used as the key derivation function to distribute the rekey messages. The security of the key derivation function lies in the hard mathematical problem of integer factorization which cannot be solved in a polynomial time. In order to increase the security, a salt value has been used along with the derivation keys for key stretching so that original keys used in the derivation function cannot be found out. The proposed protocol reduces the rekey messages by 1/d, in a d-degree tree and the number of modular exponential operations is only 2h compared to 3h in TGDH (Tree-based Group Diffie-Hellman) protocol, where h is the height of the tree.

General Terms
Key Management protocol for Group Communication.

Keywords
Rekeying, Key Derivation, Rabin’s squaring Trapdoor Function, Pseudo Random Number Generator.

1. INTRODUCTION
With the widespread use of the internet and other web related activities in all the fields, the importance of group communication has grown into prominence. Members belonging to a group can remain in contact with each other and also one member can remain in contact with the other members of the group. This kind of communication is mostly used in some of the widely prevalent applications like teleconferences, distance education programmes and chat room applications. Once the member has moved away from the group it does not remain in contact with the other group members and the communication among the group members is kept as a secret from that non-member. These kinds of communication can be unicast, when only two members need to communicate, or broadcast or multicast, when a member wants to communicate with many other members. In multicast communication there is only single message packet transmission which reaches all the destined group members. The replication of the packets to be delivered to the group members is taken care by the nodes in the network. As the members are added to the group the messages are delivered to them also which helps in taking care of major issue namely scalability. Thus all multicast communications address the issue of scalability to some extent.

When a member joins the group it is given a multicast address ranging from 224.0.0.0 to 239.255.255.255. A member joining a multicast group can be communicated via UDP (Uniform Datagram Protocol) multicast only as the current technology permits. This makes it necessary to concentrate on two major factors namely reliability and security. Securing the data sent through multicasting network is very important because the multicast network is “open” in nature since the multicast address is public. So, anyone can access the multicast address and pose as a member and get hold of all secure data. The data sent through the multicast network are susceptible to attack by the intruders who may be within the group or outside the group. Hence there is needed to take care of the security issues namely Data confidentiality, Data integrity, and Data authenticity.

Simple cryptographic techniques of encrypting a message with a key and decrypting it does not suffice in today’s environment in which the members in each multicast group is very large. Every time a member joins or leaves the group, the group key needs to be changed and communicated secretly to all the members thereby maintaining perfect forward and backward secrecy. There are several schemes namely Probabilistic schemes [11], Hierarchical schemes [3, 4, 5, 6], Conference Keying Schemes [10]. Of all the methods hierarchical methods offer a simple method to communicate data efficiently and secretly without much computations involved. While the probabilistic methods have huge storage overheads and the conference keying methods give unnecessary burden to the members who need to do huge exponential calculations. The hierarchical methods maintain a balance between both computation and storage overheads. We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material.

1.1 Evaluation Criteria:
System Architecture must not be too complex that the implementation becomes difficult. The architecture considered is a binary tree which uses a key derivation function which is cryptographically secure and has a very simple implementation.
The number of rekey messages generated should be minimized. The hierarchical system is known to have minimum number of rekey messages whenever there is member addition or deletion. With the implementation of the key derivation function the rekeying cost is still reduced. Computational complexity should not be too high. The key derivation function used has an exponentiation of power 2 which is not very complex. The security of the system should be high. Rabin’s Squaring function is known to be very secure since the security of the function lies in the mathematically hard problem of Integer factorization. Moreover finding the exact root of a number modulo n, where n is a very large composite number is infeasible.

This paper is organized as follows: In section 2 some of the existing protocols are analyzed, in section 3 some of the concepts that forms the background of the paper is discussed, in section 4 working of the proposed protocol is discussed, in section 5 performance of the proposed protocol is compared with the existing protocols, in section 6 the results are discussed.

2. RELATED WORKS

With the increase in need for multicasting with growing applications using internet, where data communication needs to be maintained secretly, there is always research in multicasting and key distribution. A simple approach includes Pair-wise keying protocols in which there exists a session key between each user pair using which they both communicate with each other. Though this protocol provides at most security, this does not scale well with large number of users since each user has to store n-1 session keys, where there are n members in the group. In Probabilistic methods [1] each user in the group is given random set of keys selected from the key pool. The users having the same key can communicate with each other. In case of member deletion, all the users sharing the key with the evicted member should change their key set. This protocol involves large storage overheads. In Hierarchical methods [3,4,5,6] a tree like structure is being followed in arranging the users. The root node is the group key which is obtained by all the group members. In Hierarchical Node based methods [1] the users of different efficiency are placed in different hierarchical levels. There is a group controller which designates the members of higher power to be sub-group controller (SGC). Each sub-group controller has a set of sub-group members who know the sub-group key (SGK) and communicate using this key. Intergroup communication is done by SGC by using Group key (GK). During member addition/deletion only the corresponding SGKs change and the rekeying cost is O (log m). In Hierarchical Key based methods [5,6], the server maintains a tree structure and all the users are added in the root node. Each node is given a key called Key Encryption Keys (KEK) and the root node has GK. Each leaf node has a member with its individual key. Each member has all the keys from leaf to the root. In case of member addition/deletion all the keys on the path from root to the leaf have to be changed. If the degree of the tree is d then the rekeying cost is dlog n. The hierarchical protocol scales well when there is large number of users. In one-way function tree [3], the keys of both the child nodes are used for calculating the keys. For this both the nodes store the blinded node key of their siblings. When there is member addition/deletion all the keys on the path of that member is changed and cost is O (log n). But there is possibility of collusion attack which is tackled in [4]. In improved OFT [4], all the blinded node keys on the path are also changed in case of member addition and deletion. This method avoids collusion attack but the rekeying cost becomes h². In the Tree Based Group Diffie-Hellman [10], which is the extension of 2-party Diffie-Hellman, all the group members calculate KEKs and the GK by themselves. This method can be efficiently used as conference keying protocol. Though there is no load on the server to distribute the keys, the members need to do large exponential calculation. It involves 2 rounds and 3 messages at the worst case during join/leave. Though the number of messages transmitted and the bandwidth used is minimum, it involves a total of 3h modular exponential calculations. In the shared key derivation method [12], one-way functions are used to find the new keys from the existing keys. Some of the members calculate the new key by themselves while for the remaining members the Server sends the updated keys. Compared to simple LKH this method reduces the rekeying cost.

3. BACKGROUND

The base structure of the proposed work is a simple binary tree. The server maintains the binary tree for storing the keys. All the members are added as the leaf nodes. The key in the root node is the group key. The keys in the leaf nodes is the individual key (IK) unique to each member. All the other keys are the Key Encryption Keys (KEK) which is used for sending the encrypted keys and not for sending the data. The data are communicated among the group by using the group key (GK). All the members have the keys which fall on the path from the leaf to the root. When a member is added or deleted all the other members sharing the keys with joining/relieving member have to update their keys.

The proposed protocol uses concept of key derivation. Instead of the server generating the new keys, some of the old keys are used to generate the new keys. These keys are called the derivation keys. Any of the cryptographic function f (.), can be used as the key derivation function. The basic idea behind using the derivation keys and the key derivation functions is that some of the members can compute the new key by themselves and need not depend upon the server to give them those keys. Thus the computation and the transport overhead on the server have been reduced. The cryptographic function to be used as a key derivation function it should satisfy the following two lemma:

**Lemma:** Given f(x) it is not computationally possible to compute x. This property is called the one-way property.

**Lemma:** Given the different x1, x2,.... which have been derived from the derivation function f(.), it is not possible to find f(.). This property is called the pseudo randomness property.

There are various cryptographic functions which could be used as key derivation functions namely the one-way hash functions,
pseudorandom number generators and the one way trap door functions. Our protocol uses the Rabin’s squaring one-way trapdoor function. The trapdoor one-way function is the one in which there exists trapdoor information, and only using that information, the function can be reversed.

Rabin’s squaring one-way trapdoor function is used in the proposed protocol as pseudorandom number generator to update keys. The Rabin’s function and its security can be explained as follows:

Let \( p, q \) be odd primes. Let \( n = pq \). Then the squaring function \( \text{SQUARE: } \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^* \) is given by,

\[
f(n, x) = x^2 \mod n
\]

The trapdoor information here is \( n = pq \). This is a hard mathematical problem. The strength of the squaring trapdoor function lies in the fact that even when \( n \) is known, finding the factors of \( n \) is not possible and it is a hard mathematical problem and not possible to solve in polynomial time. When the congruence relation, \( x^2 \equiv a \mod n \) is taken, \( a \) is a quadratic residue. Using this congruence relation the solution for \( x \) can be \( a^2m \), where \( m \) is any integer. Finding square root modulo \( n \) is as hard as factoring \( n \). When the quadratic residuosity problem is hard, Squaring trapdoor one-way function can be used as perfect pseudorandom number generator.

The quadratic residuosity problem can be stated as follows: Let there be two very large distinct odd primes, \( p \) and \( q \). Then \( n \) is an odd composite integer obtained as \( n = pq \). \( \mathbb{Z}_n \) is a group having numbers from 1 to \( n-1 \), both numbers are inclusive. It can be said that \( a \in \mathbb{Z}_n \) is a quadratic residue modulo \( n \) if and only if \( a \in \mathbb{Z}_p \), where \( a \) is a quadratic residue in the group 1 to \( p-1 \), and \( a \in \mathbb{Z}_q \), where \( a \) is quadratic residue in the group 1 to \( q-1 \). Thus Given a number ‘a’, finding whether it is a quadratic residue or a non-quadratic residue is possible only when the factors of \( n \) are known, which in turn is hard problem.

Even if it is known that a given number is quadratic residue, finding the square roots is difficult. There are four square roots for a given, \( x^2 \equiv a \mod n \) and finding the exact square roots is possible only when the factors are known. Alternatively when the four exact square roots are known the factors can be found in polynomial time. Thus finding the exact square roots of a square modulo \( n \) and the factors of \( n \) are computationally equal and both cannot be possible in a polynomial time when the other is unknown. Thus both are considered to be cryptographically hard.

Finally inside the key derivation function, a salt value \( k_s \) is used for key stretching along with the derivation keys to strengthen the key derivation function. This leaves the attacker to try every possible combination to find the original key thus increasing the security of the system further.

4. PROPOSED PROTOCOL:
4.1 Join/Leave Operations:
4.1.1 Initial Setup:
After a group of members offer their willingness to join the group, the server does the initial set up by authenticating all the members who wish to join the group and generates a binary tree. All the group members are added to the leaf nodes of the tree. The server then provides the members with the KEKS, GK and the IK.

4.1.2 Member Join:
When a member wants to join the group, the member sends a message to the server asking for its permission to join the group. The Server authenticates the member and gives it an individual key through a secure channel. The member is added to the rightmost shallowest node. If the tree is balanced then a new node is created and added. Then the server updates the entire key on the path from the leaf to the root of the added member. For key generation and updation the server uses derivation keys and key derivation function. The key derivation function used is the Rabin’s Squaring trap-door function which can be used as the pseudorandom number generator. At each level \( i \) the server selects a key among the keys which is not to be changed, which act as the derivation key for the node at the previous level \( (i-1) \). This message is broadcast to all the users. All the users who have these keys calculate the new keys by themselves and the server has to send the messages only to the remaining members thus reducing the number of rekey messages.

![Figure 2: Key updation using key derivation](image)

For example suppose user u8 joins the subgroup, rekeying takes place in the following steps.

Step 1: Selection of the derivation keys by the server from one of the keys which is not to be changed.
Step 2: Calculation of $n = pq$, Where $p$ and $q$ are odd primes and $n$ is a large composite. It is mathematically hard to factorize the integers even after knowing $n$.

Step 3: The server generates the salt value, $k_s$, which is some random value in order to avoid repetitions in the keys that are generated.

Step 4: The server concatenates all these values and broadcasts the message as in Figure 3.

$$\begin{array}{cccc}
k_7 & K_{5,6} & K_{1,4} & n & k_g \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{Derivation keys} & \text{Large composite number, product of two odd primes} & \text{Salt value}
\end{array}$$

**Figure 3: Broadcasted message for key derivation**

Step 5: (in the member side)
The members look into their database and all the members having the derivation keys calculate the new keys themselves.

Now the user $u_7$ can calculate

$$K_{7,8} = (k_7 \oplus k_g)^2 \mod n$$

$u_5$ and $u_6$ can calculate,

$$K_{5,8} = (k_{5,6} \oplus k_g)^2 \mod n$$

$u_1$, $u_2$, $u_3$, $u_4$ can calculate $G K$,

$$K_{1,4} = (k_{1,4} \oplus k_g)^2 \mod n$$

Step 6: (in the server side)
For the remaining users the server sends the appropriate keys as follows:

- Server: $E_{u_8}[K_{7,8}] \rightarrow u_8$
- Server: $E_{u_7,4}[K_{5,8}] \rightarrow u_7,u_8$
- Server: $E_{u_5,4}[K_{1,4}] \rightarrow u_5,u_6,u_7,u_8$

### 4.1.3 Member Leave:
Whenever a member leaves the group it sends the message to the server. The server grants permission to leave and updates all the keys in the path of the leaving members. The procedure for key updation is same as for member addition.

## 5. COMPARISONS AND ANALYSIS

### 5.1 Transportation Overhead:
The method of key derivation substantially reduces the number of rekey messages. The number of rekey messages is given by $\log m$, where $m$ is the number of members in the group. For normal LKH2 it is $2\log m$ (Figure 4, Figure 5). The rekey messages can be reduced to almost half when a key derivation function is used in a binary tree. This protocol can be extended to LKH of any degree $d$ and it has been analyzed and found that almost there is reduction in $1/d$ rekey messages during join and leave (Figure 6).

### 5.2 Computational Overhead:
The Squaring trapdoor function which has been used as key derivation function is an exponentiation function, but of power 2, hence feasible to compute. When the key derivation function need to be periodically changed the server can just compute the different $n$ values by changing the $p$ and $q$ values instead of constructing a whole new function. Since strength of the squaring function lies on the hardness of the integer factorization, Rabin’s squaring function can be used as a key derivation function. The number encryptions done by the server is also reduced since some of the members calculate the updated keys by themselves and the server need not send any encrypted messages to them.

The implementation of the Tree Based Group Diffie-Hellman protocol though drastically reduces the rekey messages it involves calculation of huge exponentiations on the member side which becomes unnecessary burden on the member side. The total number of modular exponentiations is also reduced to $2h$, $h$ is the height of the tree from $3h$ for TGDH (Figure 7). With the assumption that the server has greater power than the members it can take the burden of computing the new key and passing on to the section of the users as in our protocol.

### 5.3 Storage Overhead:
In the proposed protocol the number of keys stored in each member is same as LKH protocol i.e., all the keys from the member to the root node. But in the Tree Based Group Diffie-Hellman protocol the storage in each member is $\log m$ keys from the nodes in the path and also the blinded node keys of the siblings on the path, totally, $2\log m$. Thus the storage overhead is very high in this protocol compared to the proposed protocol.

## 6. RESULTS
The following table summarizes all the results in terms of storage and number of rekey messages during join and leave.

### Table 1 Comparison of performances of various protocols

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Simple LKH2</th>
<th>Tree based Group Diffie-Hellman</th>
<th>proposed protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total no. of keys in the server</td>
<td>$2m-1$</td>
<td>$4m-3$</td>
<td>$2m-1$</td>
</tr>
<tr>
<td>No. of keys in each member</td>
<td>$\log_2 m$</td>
<td>$2(\log_2 m)$</td>
<td>$\log_2 m$</td>
</tr>
<tr>
<td>Rekey messages during join</td>
<td>$2\log_2 m$</td>
<td>$3$</td>
<td>$\log_2 m$</td>
</tr>
<tr>
<td>Rekey messages during leave</td>
<td>$(2\log_2 m)-1$</td>
<td>$3$</td>
<td>$(\log_2 m)-1$</td>
</tr>
</tbody>
</table>

From the table it is seen that our protocol has less number of rekey messages without trading off on storage much. LKH has same storage but higher number of rekey messages. Tree-based
Group Diffie-Hellman protocol has only 3 rekey messages and 2 rounds but has a huge storage cost.

The number of modular exponential operations of our protocol are compared with Tree-based Group Diffie-Hellman protocol. It is found that our protocol has less modular exponential operations than TGDH and hence the computational complexity of our protocol is also balanced. The exponential operations are only of the power 2 and hence less computational overhead on the memberside whereas in TGDH members need to perform huge exponential operations.

<table>
<thead>
<tr>
<th>Modular Exponential operations</th>
<th>Tree-Based Group Diffie-Hellman</th>
<th>proposed protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join</td>
<td>3h (2h by the sponsor node and h by other nodes)</td>
<td>2h (h by members and h by server)</td>
</tr>
<tr>
<td>Leave</td>
<td>3h (2h by the sponsor node and h by other nodes)</td>
<td>2h (h by members and h by server)</td>
</tr>
</tbody>
</table>

Figure 4 Comparison of rekey messages during join

Figure 5 Comparison of rekey messages during leave
7. CONCLUSION

In this paper an efficient group key management protocol has been proposed which is also scalable. The proposed protocol uses hierarchical framework which is known to have minimum number of rekey messages. Though the protocol can be extended to a tree of any degree, binary tree has been implemented which is known to be the simplest structure and very easy to implement. The proposed work uses the Rabin’s squaring trapdoor function as the key derivation function and breaking this function is a hard mathematical problem. This function is thus used as a secure pseudorandom number generator. The salt value that is being used for key stretching along with the derivation key increases the randomness and also making the function more secure. The new key obtained is the quadratic residue modulo n, and it is very hard to find the roots of the quadratic residue modulo n. Even if there is a remote chance of finding the roots, the original derivation key is impossible to find with the use of the salt value in the squaring function. Thus the proposed protocol is secure and also scalable. The storage overhead is also balanced. The computation of the new key is also not very complex with only exponential modular calculations of power 2.

8. ACKNOWLEDGEMENT

Our thanks to the experts who have contributed towards the development of this research work. Our sincere thanks to our family and friends who supported and encouraged us.

9. REFERENCES


Abstract - Database security has paramount importance in industrial, civilian and government domains. Organizations are storing huge amount of data in database for data mining and other types of analysis. Some of this data is considered sensitive and has to be protected from disclosure. Challenges for security in database are increased due to the enormous popularity of e-business. In recent years, insider attacks gathered more attention than periodic outbreaks of malware. Database systems are usually deployed deep inside the company network and thus insiders has the easiest opportunity to attack and compromise them, and then steal the data. So data must be protected from inside attackers also. Many conventional database security systems are proposed for providing security for database, but still the sensitive data in database are vulnerable to attack because the data are stored in the form of plaintext only. Database encryption is the only solution for avoid the risk posed by this threat.

This paper focuses on a security solution for protecting of data-at-rest, specifically protecting the sensitive data that resides in databases by using TSFS algorithm with three keys thus it provide more security for database. This algorithm improves the efficiency for executing the queries in database by encrypting only the sensitive data.

Keywords - Database Encryption, Key expansion, Transposition, Substitution, Folding, Shifting.

I. INTRODUCTION

Many organizations increase their dependency on database systems for day-to-day operations and for making decision. With a networked database in the complex multi-tiered applications, multiple parties such as customers, partners, and internal and external users will share the information inside the database. The security of data managed by these systems becomes crucial. Damage and misuse of sensitive data stored in the database not only affect a single user also affect entire organization. The recent development of web based applications and information systems have further increased the risk exposure of databases. The available security policies cannot provide a secure support for the sensitive data, which reside in the database, as the illegal and unauthorized users may obtain the readable data.

There are four methods of enforcing database security: physical security, operating system security, DBMS security, and data encryption. Out of our survey, the first three methods however are not totally satisfactory solutions to the database security problem, for the following four reasons. First, it is difficult to control the attack on raw data because the raw data exist in readable form inside a database. Second, it is impossible for the operating system and DBMS security to the disclosure of sensitive data, because the sensitive data must be backed up in storage median in case of system failure. Third, it is hard to protect the confidential data in a distributed database system. Fourth, it is hard to verify that the origin of a data item is authentic, because an intruder may have modified the original data. Encryption of the data has the potential to solve all the previously mentioned problems, If the data are not in a readable form, obtaining the data will be of no advantage to a person without the proper key to decrypt it. Thus the problem of data disclosure can be eliminated and the data authenticity problem can also be largely solved by encryption.

In this paper, we propose an efficient light-weight database encryption techniques using TSFS (Transposition, Substitution, Folding, Shifting) algorithm, only the sensitive data in the database are encrypted by using this algorithm, so it will provide efficient execution of queries and give quick response to the users. TSFS is the symmetric - key block encipherment algorithm, for symmetric encryption, same key is used for encryption and decryption and security is dependent on the length of the key. Here we use three keys for the process of encryption and decryption. For providing effective and more security for the database these three keys are expanded in into 12 sub keys by using the key Expansion Technique.

The remainder of this paper is organized as follows: Section 2 discusses the existing system for securing the database and analyzes the strength and drawbacks. Section 3 presents how the three keys are expanded into sub keys. Section 4 briefly discusses the design and implementation of the proposed approach. Section 5 presents some strength of the TSFS algorithm. Section 6 explain the security analysis of the algorithm. Section 7 draws the conclusion.
Many methods having more creativity and efficient implementation have been proposed for database security research field. Using efficient keys and sub keys a database encryption scheme based on the Chinese Reminder theorem \[4\] also implemented. In that theorem, a record oriented cryptosystem in the database, which enables encryption at all the level of rows and decryption at all the level of cells are implemented. Extension of the sub key in encryption by supporting multilayer access control is proposed by Hwang and yang to enhance the security level. They also introduced a two-phase algorithm for database security \[5\]. Another scheme called chip secured data access \[6\] principles as a solution to data confidentiality problem. This solution is quite secure and effective, but it is still too complicated and the cost is too expensive.

These database encryption mechanisms provide an effective way to keep the sensitive data in security by store the data in encryption form. But once the database is encrypted, the efficiency of DBMS will fall, as the query cannot execute over the encrypted attribute directly. In order to get the query results, the users have to decrypted the whole data first and then conduct the query over the plaintext data. The process of decryption will not only affect cost of time but also leak the sensitive data to the attackers. There are several methods try to solve the problems on the efficiency and secure on the sensitive database. In privacy homomorphism technology \[7\], the queries can execute directly in the encrypted database. But this schema has weak encryption strengthen. The attacker could get some sensitive information through simple comparison of parts of cipher text and plaintext.

In the OPES approach \[8\] the queries are directly applied on encrypted data, without decrypting the operands. This scheme allows comparison operation to be directly applied on encrypted data without decrypting the operands. Thus equality queries as well as the max, min, and count queries can be directly processed over encrypted data. But it is not sufficient for executing the complex queries ant it is not inherently secure for straightforward attack. Encryption for indexing in a column-oriented DBMS \[9\] is another method for database encryption that encrypt only the specified columns so that only the sensitive data are selected for encryption thus the process of encryption and decryption time is reduced. Another scheme numeric to numeric database encryption for encrypting the numeric data \[10\], but this approach is not fit for the character data as the character data has its specific features which is different from the numeric data and this scheme work only for integers not for decimal point numeric data, Sometimes character data could be a very sensitive one so that we have to provide security for both numeric and character data.

Out of those surveys, we present TSFS algorithm to overcome some drawbacks in the existing methodologies and also ensure the confidentiality and integrity of the data in the database.

In this step, each key is expanded to many sub keys to be used in each round. In general the keys are expanded by shifting the rows \[11\] and by using Add round key technique in AES and many techniques are available for expanding the key. In the proposed scheme, three keys are used and each key is expanded to four sub-keys. The keys are in any format consist of numbers, alphabets, combination of both but not accepting the special characters and symbols. In order to increase the strength of the algorithm, the given keys are stored in the form of 4 x 4 matrix so the length of the key must be in 16 digits. If the user gave the keys below 16 digits, first the keys are converting in to 16 digits by using padding technique and then stored the keys in the matrix. After that shifting the rows for key expansion and it will be used in real time process to expand the keys.

The following example describes how the three keys are expanded into 12 sub keys.

For example when the

Key1 value is 66FF3CE3491C5EDA
Key 2 value is 95EFFBE191E22DB4
Key 3 value is 9CC98A29456677A6

Here, we get these keys by using a random key generator. It is not necessary that a random key generator must only be used for obtaining the key values. Key values are specified by the users as they wish.

First the keys are converted into numbers based on the position in the alphabets a-z (a-0, b-1-------z-25). Then the keys are stored in 4*4 matrix form.

The keys are expanded based on shifting the rows.

Key1 is expanded into key$_{10}$, key$_{11}$, key$_{12}$, k$_{13}$.

For key$_{10}$ - row 0 is not shifted, row 1 is shifted one time, row 2 is shifted two times and row 3 is shifted three times

For key$_{11}$ - row 0 is shifted one time, row 1 is shifted 2 times, row 2 is shifted three times and row 3 is not shifted

For key$_{12}$ - row 0 is shifted two times, row 1 is shifted three times, row 2 is not shifted and row 3 is shifted one time

For key$_{13}$ - row 0 is shifted three times, row 1 is not shifted, row 2 is shifted one time and row 3 is shifted two times

Figure.1 shows the key expansion process
In this paper, we propose an efficient lightweight database encryption scheme (called TSFS). The main objective of this paper is to propose a secure database encryption scheme that provides maximum security, whilst limiting the added time cost for encryption and decryption. Only the sensitive data are encrypted, so it is very effective for executing the queries. This algorithm is working for numeric data, decimal point numeric data and also for the alphanumeric data.

To provide security, TSFS algorithm uses four types of transformations: Transposition, Substitution, Folding and Shifting.

Transposition and substitution ciphers are still the most important kernel techniques in the construction of modern symmetric encryption algorithms. The benefit of these two ciphers is that they have two factors of cryptology and security, diffusion and confusion. In this algorithm, we mostly use transposition and substitution ciphers techniques.

Let’s see how TSFS uses four types of transformations for encryption and decryption. In the standard, the encryption algorithm is referred to as the cipher and the decryption algorithm as the inverse cipher. This algorithm is a non-Feistel cipher, which means that each transformation or group of transformations must be invertible. In addition, the cipher and the inverse cipher must use these operations in such a way that cancel each other. The keys must also be used in the reverse order. The following figure shows the overall view of the algorithm.

A. Overview of the algorithm

IV. PROPOSED ENCRYPTION SCHEME

In this paper, we propose an efficient lightweight database encryption scheme (called TSFS). The main objective of this paper is to propose a secure database encryption scheme that provides maximum security, whilst limiting the added time cost for encryption and decryption. Only the sensitive data are encrypted, so it is very effective for executing the queries. This algorithm is working for numeric data, decimal point numeric data and also for the alphanumeric data.
If user wants to encrypt private data that can be seen and altered only by himself, a randomly generated working key will be used to encrypt the private data with a conventional encryption algorithm. After the data entered by the user immediately that data is encrypted by using the algorithm and the encrypted data is stored in the database. This algorithm is working for both numbers and characters and also it accept the alphanumeric data also (for example the account number is asdf48723498), if decimal point numbers are entered then that number is multiplied by 100 and then applied to the algorithm. Why we multiply by 100 means because for Indian Rupees the Pisa column have only two integers so that for covert the decimal point numbers to normal numbers, we use this technique. After the algorithm encrypted the data and the data is divide by 100 then it stored in the database. Thus the decimal point number is stored in the same data type in the form of cipher text.

A. Transposition

Transposition ciphers are and important family of classical ciphers. It does not substitute one symbol for another; instead it changes the location of the symbols. A symbol in the first position of the plaintext may appear in the tenth position of the cipher text. In other words, a transposition cipher reorders the symbols. In this algorithm we use diagonal transposition cipher the entered data are stored in 4 x 4 matrix form and then the data are taken diagonally and stored in the another matrix for the consequence. The following figure show how the data are taken diagonally. For example the entered data is the account number:  asdf48723498. After getting the data, pad the input data with 0s and stored in the matrix form.  Just like this (left matrix) and the data’s in the matrix are read in the route of zigzag diagonal starting at the upper left corner. The following figure shows the result of the transposition.

![Fig.3 Transposition](image)

B. Substitution

A substitution cipher replaces one symbol with another. If the symbols in the plaintext are alphabetic characters, we replace with one character with another. For example, we can replace letter A with letter D, and letter T with letter Z. If the symbols are digits (0 to 9), we can replace 3 with 7 and 2 with 6. Substitution ciphers can be categorized as either monoalphabetic ciphers or polyalphabetic ciphers. In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the cipher text is always one-to-one. In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a symbol in the plaintext to a symbol in the cipher text is one-to-many. Here we use a new modified affine cipher for encryption. It is one of the monoalphabetic ciphers available.

Normally affine cipher is a combination of additive and multiplicative cipher. For this we have to use two keys one for the additive cipher and another for multiplicative cipher.

By using this cipher the Encryption process is

\[ C = (P \times k1+k2) \mod M \]

Decryption process is

\[ P = ((C – k2) \times k1^{-1}) \mod M. \]

In this cipher the multiplicative inverse of k1 only exists if k1 and M are co prime. Hence without the restriction on k1 decryption might not be possible. What is the key domain for any multiplicative cipher? The key must be in the range from 0 to 26.

This set has only 12 members :1,3,5,7,9,11,15,17,19,21,23,25. Considering the specific case of encrypting messages in alphanumeric in English (i.e. M=26), So there are 12 x 26 or 312 possible keys. So it is easy for the cryptanalyst to find the key.

To overcome this draw back we slightly changes this affine cipher i.e. here we eliminate the process of multiplicative cipher and add one more additive cipher. In this cipher we give more importance for selecting the two keys for encryption. We expand the three keys into twelve keys and stored in the form of 4 x 4 matrix and also the entered data are also stored in the form of matrix. For encrypting the 0th row and 0th column data in the matrix we take the k1 from the same row and column of the expanded keys key_{10} and the k2 from key_{11} and the same format is used for encrypting the other data’s in the matrix. Here for the first round we use the key_{10} and key_{11} and for the second round we take the key k1 from key_{11}, and k2 from key_{12} and the same process used up to the 11th round, in the 12th round we take k1 from key_{13} and k2 from key_{10}. Based on this method keys are selected for encryption process.

The encryption function E, for any given letter x is

\[ E(x) = (((k1+p) \mod M) +k2) \mod M \]

Where modulus M is the size of the alphabet and k1 and k2 are the key of the cipher. In this cipher k1 is not need to be prime number.
The decryption function \( D \) is

\[
D(E(x)) = (((E(x) - k_2) \mod M) - k_1) \mod M
\]

The main strength of the encryption is in selecting the keys. Let’s see how it works, the input of this cipher is the result of the transposition cipher, after getting the input, the modified affine cipher is applied to each data in the matrix and that data is covert to another data. The following fig shows the result of the substitution.

![Substitution](image)

**C. Folding**

After substitution the result is taken as input to the folding technique. Folding is one of the transposition cipher, just like the paper fold, the matrix is folded horizontally, vertically, and diagonally. This folding technique shuffles the data from one position to another position. The following figure shows the result of folding.

![Folding](image)

**D. Shifting**

This is a simple shifting cipher which provides a simple way to encrypt and numbers by using 16- array element of numeric digits. Each element must contain the 26 numeric characters from 0 to 25. Each digit must appear only once in each element of the array. The digits can appear in any order you like.

The input of this cipher is the result of the folding cipher. In the shifting cipher the program replaces each digit of the number by its position within its array element. For decryption the position is given as an input based on the position the data is taken and that data is plaintext of the given cipher text.

This process is illustrated in the following figure.

![Shifting](image)

**VI. STRENGTH OF TSFS ALGORITHM**

The number of keys is more so that the key combination increases which makes guessing of keys harder. The main strength of the algorithm is in the substitution transformation because selecting the key for finding the cipher gave more security to the data. In this algorithm the numeric plaintext have numeric cipher text, character plaintext have character cipher text and if the input data is alphanumeric type then the output cipher text also in alphanumeric, so there is no need for change the data field type when the encrypted data are stored in the database. Only the sensitive data in the database are encrypted so that for getting one or two data from the database, it is not need to decrypt all the data’s in the database. Thus the proposed technique increases the speed of executing the queries in the database.
VII. SECURITY ANALYSIS

Generally there are two kinds of attacks such as passive attack and active attack. Statistical attack comes under passive group. The cipher index value is computed with the help of TSFS algorithm and stored. It is too difficult to recover the data from the cipher index value by applying statistical attack. The benefit of the proposed scheme in terms of security is the data type of plain text and cipher text is same. It is difficult to analyze the recovered data by the attacker, it’s encrypted or not. For example the credit card number of the customer in bank sector is encrypted by using TSFS algorithm is shown in the following figure.

<table>
<thead>
<tr>
<th>Before Encryption</th>
<th>After Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>2546321756985412</td>
<td>8754123659874521</td>
</tr>
</tbody>
</table>

Fig. 7 credit card encryption

Even though the adversary get the authorization to read the data in the database, he cannot identify whether the credit card number is encrypted or not. Another main strength of the proposed scheme is the number of keys are more and hence the key combination increases to \(10^{64}\) which makes guessing of keys harder to the attacker. For the key expansion technique, the various keys and their values in the rows were plotted against randomness and key ranges are shown in the graph. The following graph depicts the variation of values in the seed key and their sub keys. Here we take a sample of key1 and their sub keys \(k_{10}, k_{11}, k_{12}, k_{13}\). For each round two sub keys are used in the substitution transformation. \(k_{10}\) and \(k_{11}\) used in the first round, \(k_{11}\) and \(k_{12}\) used in the second round, \(k_{12}\) and \(k_{13}\) used in the third round and the continuous pair of keys are used for each round. Variations between values of rows in keys provide more security for the data.

Fig. 8 variation of keys
VIII. CONCLUSION

Database attacks are increases in the risks of data disclosure. Many organizations must deal with legislation and regulation on data privacy. In this environment, our security planning must include a strategy for protecting sensitive databases against attacker. In this paper we propose a new light-weight and effective database encryption algorithm for encrypting the sensitive data that reside in the database. If sensitive data are encrypted before storage in the database, risks from security leaks can be eliminated and the security issues of the database will reduce by using three cryptographic keys for protecting the data. Our proposed scheme is considered as efficient because it provides maximum security to the database and also increases the process of encryption and decryption.

The proposed algorithm can be implemented for securing any corporate related accounting information which contains numeric data, decimal point numeric data and alpha numeric data but it does not work with any symbols. Suppose if we want to encrypt the user email id, it is not possible to encrypt by using this algorithm. So it is possible to enhance this algorithm for encrypting the symbols and other special characters.

REFERENCES

PROTOCOL FOR ADVANCED WIRELESS SENSOR NETWORKS

1 TULASI DWARAKANATH.V, 2 D.MANIVANNAN

1M.Tech, Embedded Systems, SASTRA University, Thanjavur, India – 613402.

2SOC, SASTRA University, Thanjavur, India – 613402.

E-mail: 1 dwaraka.tulasi@gmail.com, 2 dmv@cse.sastra.edu.

ABSTRACT

The Wireless Sensor Network (WSN) comprises of the nodes with the low power processors having very limited resources and is deployed in field to sense and analyze the data. The protocols used in the sensor network must be simple. The data collected by the node is transmitted over the sensor network is encapsulated with the IP satisfying the requirements of 6LoWPAN [3]. The computation overhead on the sensor node must be less in order to increase the life of a Sensor Node. The Simple Data Transmission (SDT) Protocol is used to encapsulate the data collected by the node. The SDT protocol has the provision of choosing the type of data to be collected by the node at various instances. The SDT protocol is developed to collect the data from the various sensors on the board to frame the data and to actuate a sensor on the board according to the requirement of the application.

Keywords: WSN, 6LowPAN, SDT.

1. INTRODUCTION:

WSNs have a range of applications in various fields [2]. The WSNs are developed with the devices consuming low power, operating at a low data rate, comprises of many individual nodes which perform a specific task and the data is collected by the coordinator. Each node can work independently and in combination with the neighbors to fulfill the requirements like increasing the overall performance of the network. The traditional and proprietary Sensor Networks has limitations in terms of network scalability and interoperability among the nodes with a different communication standard. The 6LoWPAN network is shown in Figure 1 indicates various types of Networks according to the field of application. The 6LoWPAN specification has called for the standardization on some of the critical issues of WSNs [3]. To solve these issues a new protocol must be developed to extend the livability of a WSN in the future. This requires a unique global identity to enhance the global connectivity of the node for transferring the data. The following work provides a solution for the limitations of traditional WSN nodes and makes compliant to IPv6 and the later communication standards.

6LoWPAN finds its scope in various areas like Industry Automation, Transport, Health monitoring, Home Automation etc. thus forming internet of objects. The application of Sensor Network is dependent on the area its going to be deployed. The protocols used in the WSN must be designed to satisfy the requirement where the Sensor Network is designed for an application. The SDT is designed to encapsulate the data with the IP. The SDT was tested with C-DAC’s Ubimotes.

Section 2 explains the SDT, its Scope and Command base. Section 3 explains the implementation of SDT on Ubimotes.

Figure 1: 6LoWPAN
2. SIMPLEx DATA TRANSMISSION PROTOCOL:

2.1. Scope of SDT:

The Simple Data Transmission protocol is developed for the low power applications, which suit the WSNs. The SDT protocol has a set of commands to transmit the data to and fro the node and edge router of the sensor network. The SDT protocol is based on the client server architecture in which a command is sent to the sensor node from the coordinator or edge router and the reply is sent in the same fashion from the node to the coordinator. The request and reply comprises of a command indicating the type of data and mode of collection. The type of data collected is dependent on the type of sensor attached to the sensor node and the mode is the way the data is collected by the node from the sensor i.e the communication interface the sensor is connected. The SDT protocol is designed to work on layer 3 networking protocols such as TCP/IP or any other low power IP standard like Berkeley Low power IP (BLIP).

The Simple Data Transmission Protocol lies in the Application Layer of the Sensor Node Architecture. The Network layer comprises of BLIP which supports UDP/IP and the bottom layer is based on 802.15.4.

![Application Network Physical SDT UDP/IP IEEE 802.15.4](image)

**Figure 2**: Scope of SDT

2.2. Structure of SDT:

The first field is protocol byte which indicates that the packet belonging to the Sensor Network.

The second and third fields indicate the length of payload and CRC respectively. The fourth and fifth fields indicate the source and destination address of the nodes. The sixth field holds the command and data to be transmitted from the coordinator to node and vice versa. The last field has the CRC code to check for the errors.

![Command Base](image)

**Table 1**: Command Base

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Query</td>
<td>C → N</td>
</tr>
<tr>
<td>R</td>
<td>Reply with Sensor Info</td>
<td>N → C</td>
</tr>
<tr>
<td>S</td>
<td>Sensor Data query</td>
<td>C → N</td>
</tr>
<tr>
<td>D</td>
<td>Reply with Sensor Data</td>
<td>N → C</td>
</tr>
<tr>
<td>A</td>
<td>Actuate Sensor</td>
<td>C → N</td>
</tr>
<tr>
<td>B</td>
<td>Sensor Actuated</td>
<td>N → C</td>
</tr>
<tr>
<td>E</td>
<td>Error</td>
<td>N ↔ C</td>
</tr>
</tbody>
</table>

C – Coordinator, N – End Device

The SDT commands combined with the possible other fields. Here is seen the ‘Payload’ part only.

**Query – ‘Q’**

![Addr Q](image)

Reply with Sensor info – ‘R’

![Addr R Sensors](image)

Sensor Data Query – ‘S’

![Addr S Sensors Int](image)

S – Query Sensor Data
Int – Sample Interval
The Coordinator transmits query messages to the Nodes. Nodes will reply accordingly to the coordinator. If any mismatch between the query and corresponding reply the Error message will be transmitted from the coordinator and if the query is not available then error will be transmitted to the coordinator.

3. IMPLEMENTATION:

The Simple Data Transmission protocol is developed on the Hardware Independent Layer of TinyOS. The underlying two layers are following the functionality of the BLIP to access the Hardware Abstraction and Hardware Presentation Layers.

```
<table>
<thead>
<tr>
<th>HIL</th>
<th>SDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAL</td>
<td>TinyOS</td>
</tr>
</tbody>
</table>
```

**Figure 4:** Scope of SDT in TinyOS

The typical packet of SDT has 7 fields which consist of various data to transmit to the coordinator. The coordinator which collects the data maps the data payload with the original IP standard and transmits to the network administrator where he can collect the sensor data and can manipulate according to the requirement.

The Simple Data Transmission Protocol is implemented on the motes comprising of MSP430F1611 and CC2520 radio. The SDT protocol is merged with BLIP on TinyOS [5] such that it satisfies the application standard of 6LoWPAN. The BLIP stack has been modified in order to reduce the memory overhead created by unnecessary applications running along with SDT.

The Sensor Board has various sensors to measure the physical parameter and has onboard ADC to choose a particular Sensor. The Sensor Board is connected to the Node with many communication standards like UART, I2C so that the programmer can choose the way the Sensor Board need to be connected to the Wireless Sensor Node.

The interfaces for the sensor board have been developed in TinyOS. The TinyOS 2.1.1 has been modified such that it will work for MSP430F1611 and CC2520.

**Figure 5:** (a) Sensor Board (b) Processor Board (c) Sensor Node

4. CONCLUSIONS:

The SDT is compared with the other application protocols like SNMP [1], SIP. The protocols SNMP, SIP are for network management and Session Initiation in the Sensor Networks and not for Data framing. The Simple Data Transmission Protocol has been tested on the Sensor Node with MSP430 and CC2520 radio and the Data has been transferred from the End Device to the Coordinator.
5. **Citations:**

This is a part of research project on Developing Protocols for Advanced WSNs in UCHR, C-DAC Bangalore.

**REFERENCES:**


http://nescc.sourceforge.net/papers/nesc-ref.pdf

[6] TI's 2.4 GHZ IEEE 802.15.4/ZIGBEE® RF Transceiver, CC 2520 Data Sheet

Wireless Accident Information System Using GSM and GPS

R. Rathinakumar and D. Manivannan,
School of Computing, Sastra University, Thanjavur, India -613401

Abstract: This study discusses about designing a Smart Display and Control (SDC) which will monitor the zone and maintains the specified speed in the zone levels, which runs on an embedded system. This system includes three modules; automatic speed control module, accident detection and information sending module and security enabling module. Automatic speed control module includes RF transmitter placed in specific location and RF receiver in the vehicle. Accident detection module includes GSM and GPS technology. Security enabling module includes sensor units which ensures the condition of seat belt and the driver. This module includes alcohol sensor and eye sensor. The smart display and control is composed of two separate units: Zone status Transmitter unit and Receiver (speed Display and Control) Unit.

Key words: Alcohol sensor, eye sensor, GPS, GSM, PIC microcontroller, vibration sensor

INTRODUCTION

Nowadays accidents occur in all the places but major accidents occur in school zone and college zone. Because of high speeding of vehicle. The main objective of the system is to provide security for the vehicle user and also detects the accident if occurred and informs the respective authority through wireless technologies. If any accident occurs in highway or any other place, the accident information system will get activated and message will be transmitted to respective authority (Rajesh et al., 2010). Statistical report says that the accident occur due to the following reasons; drunk driver not using the seat belt properly. This system will check all these things before the vehicle starts. This automatic accident detection system will overcome the above mentioned problems in an effective way. Present system checks only the seat belt condition and lacks much security constrains.

According to this system, whenever a person sits in driver seat of the vehicle, the system checks for the following parameters with the driver. The Alcohol sensor, which checks if the person has consumed alcohol or not. The eye sensor makes sure that the person in driver seat does not falls asleep. In case of any accident, the vibration in vibration sensor increases beyond the limit and information is sent to GSM module. The GSM can send message to respective authority. Thus this system ensures the life security.

Overview of the system frame work: In this design process three modules were used, automatic speed control module, accident detection module and security enabling module. In automatic speed control module the RF transmitter and RF receiver are to be used. The wireless Transmitter can send the data up to 100 feet away from the vehicle. The RF module used here operates with a carrier frequency of 418 MHZ within the 260 MHZ to 430 MHZ RF Spectrum (unlicensed Spectrum) thus avoiding any FCC (Federal Communication commission) charges or regulations. The RF transmitter is placed in a specific location and RF receiver is placed in vehicle. When the vehicle reaches the zone like school zone or U turn, it will automatically reduce the speed and when it leaves the zone it will automatically regain its speed. From this the occurrence of accidents will get reduced. In information sending module GSM, GPS and vibration sensors are used. In this system, the vibration sensor, GPS and GSM is placed in the vehicle. If an accident occurred the vibration sensor senses the vibration level and if it exceeds the threshold limit, the system will consider that there is an occurrence of accident in that particular location. Then the system will activate the GPS to gather the location detail and sends the location of the vehicle through GSM to the control station. Security enabling module includes eye sensor and seat belt detector which will ensure the security condition of the driver.

Nowadays the accident occurs due to a drunken driver and improper use of belt. The main cause of accident is due to the driver drowsy and tired condition, which will be noticed by the help of eyes sensor which will not start and also informs the status of the driver to the base station along with the vehicle ID.

PROTOTYPE DESIGN

Transmitter design: Figure 1 shows the transmitter module. Transmitter module is placed in the specific zone. Transmitter and receiver both operate at a frequency of 430 MHZ. transmitter receives data serially and sends the data to the receiver continuously. The RF transmitter is
sensor (vibration sensor) will indicate the controller and controller will in turn transmit the message to the hospitals and police stations through GSM technology.

The function of Global Positioning System (GPS) is the most promising technology to acquire the position of the accident occurs due to drunken driver and improper use of seat belt. Before the vehicle starts the driver will be wearing the seat belt it will also indicate it and also will not allow the driver to move the vehicle.

**Design process:** In automatic speed control module RF nodes are used, the RF module consists of RF transmitter and RF receiver. The RF transmitter is used for the transmission of data the a rate of 1 to 10 kbps it will operate at a frequency range of 430 MHZ. RF transmitter is placed in specific zone and RF receiver is placed in the vehicle. The RF receiver also works with same frequency. The transmitted data which is sent by RF transmitter is received by the receiver and is validated. Accelerometer sensor (vibration sensor) is connected to the port A, which will provide analog value to the ADC port of controller. The GSM is interfaced to PIC microcontroller.

PIC controller used here is PIC16F877A. It has got five ports namely PORTA, PORTB, PORTC, PORTD and PORTE. PORTA and PORTE are meant for analog input data. PORTS B, C, and D are used for input and output purpose. The purpose of GSM is for sending the message to police, ambulance, and relatives. The GSM will communicate via the UART communication through RS232 standard. The GPS suits best for vehicle location or tracking. To know the location of vehicle GIS software can be used. Alcohol sensor is connected to port A which will indicate if the driver is in drunken state and will not allow the vehicle to move. Seat belt detector is also used to detect whether the driver is wearing the seat belt or not and an eye sensor will monitor the drivers cautiousness constantly.

LCD is connected to port B of the controller to which all data pins of LCD are connected. The alcohol sensor is connected to port pin RA0. The eye sensor is connected to RA1. Vibration sensor is connected to RA2. RA0, in that 1, 3 pins take as the analog inputs from sensors and convert them to digital values. The alarm circuit is connected to port D. GSM, GPS and RF modules are connected to TX and RX of PIC microcontroller.

The Fig. 3 shows the flow diagram of the overall system. First step in that process is alcohol detection and seat belt detection. The alcohol sensor and seat belt

---

**Fig. 1:** Transmitting module

**Fig. 2:** Prototypic design of accident detection, speed control and security module

placed in zones like school zone, college zone, U turn. When the vehicle reaches those zones, it will automatically reduce the speed (Ben Carroll et al., 2010) to 20 KM. When the vehicle leaves the zone it will regain its speed. The receiver module is placed in the vehicle. When the signal from transmitter is received it will decode the encoded data and indicate the controller to reduce or limit the speed of the vehicle.

When the vehicle leaves the zone it will regain its speed. The receiver module is placed in the vehicle. When the signal from transmitter is received it will decode the encoded data and indicate the controller to reduce or limit the speed of the vehicle.

**Receiver design:** Figure 2 shows the receiver prototypic design. The RF module consists of RF transmitter and RF receiver. The RF module has an encoder in transmitter and decoder in the receiver. The encoder is used for encoding the parallel data for transmission while the reception is decoded by decoder in the receiver. The RF receiver is connected with PIC microcontroller. The RF receiver will be always in listening state, if it receives any signal of same frequency as of receiver, it will automatically indicate the controller which in turn reduces or limits the speed of the vehicle until the vehicle leaves that particular zone. From this, the accident in school and college zone will get reduced. The accelerometer is connected with microcontroller and placed in vehicle. If any accident occurs in the highways, the accelerometer
controller which will immediately activate the GPS (Wu and Shenzhangyi, 2011) and collects the location detail and sends the message to the respective numbers stored in that controller (the number may be hospital or police stations number).

EXPERIMENTATIONS AND RESULTS

Figure 4 shows the prototypic model of the vehicle enabled with RF module, Alcohol sensor, GSM, GPS and Seat belt detector. Figure 5 and 6 shows the alcohol level indication and the seat belt off indication.

CONCLUSION

This work solves the issues like automatic speed control mechanism, accident detection and information sending. From this we conclude that this system will reduce the accidents and save the human lives. On the whole this system proves to be very cost effective and efficient. The experimentations and results prove that the system is easily implementable in real time. This system can also be extended by inducing automation concepts like automatic driverless vehicle system, inter vehicular communication etc.

ACKNOWLEDGMENT

Author wishes to acknowledge Dr. P. Swaminathan Dean School of computing SASTRA UNIVERSITY for his time and support.
REFERENCES


