International Journal of Computational Intelligence and Informatics, Vol. 4: No. 2, July – September 2014

Matrix based Key Pre – Distribution Scheme for Wireless Sensor Networks

T A Tharani
Dept. of Information Technology
kumaraguru college of technology
Coimbatore
09tharanicbe@gmail.com

N Suganthi
Dept. of Information Technology
Kumaraguru college of technology
Coimbatore
suganthi.n.it@kct.ac.in

R Srinithi
Dept. of Information Technology
Kumaraguru college of technology
Coimbatore
emailtonithi@gmail.com

***Abstract*-Wireless sensor networks are used in various applications now-a-days. As they are deployed in open area, there is a need for key management in order to protect the information stored in sensor nodes. To address this problem, we use key pre-distribution scheme. In this paper, we propose a new scheme based on symmetric matrix and maximum rank distance (MRD) codes where the size of the symmetric matrix is kept constant to reduce the memory requirement at each node. Some information about the matrix G and matrix A is stored in sensor nodes to generate secret key between them and for secure communication. Only two messages are required to generate a secret key between two nodes and thus it reduces the communication overhead. This scheme has greater network connectivity and scalability. Newly deployed nodes can generate a key without changing any information on previously deployed nodes. To provide additional security, the final result from key generation scheme is applied to the division remainder hash function and the resultant value is used as the secret key between the nodes.**

Keywords- Key Pre-Distribution, Matrix, Security, Sensor Node, Less Memory, Scalability.

I. INTRODUCTION

A wireless sensor network (WSN) is a network formed by a large number of sensor nodes, each equipped with sensors to detect environmental changes, security and health monitoring. Generally wireless sensor networks have limited energy and memory capabilities. Such resource limitations lead to many problems which have been studied by researchers. WSN are vulnerable to many attacks such as black-hole attack, Sybil attack, etc. as described in [1] which is a threat to the information stored in sensor nodes. Today security must be enforced in WSN using secret keys for communication between the nodes. Many schemes available today do not satisfy the requirements such as low storage, computation, and transmission overheads. This scheme is based on the idea proposed by Blom [2] but we use maximum rank distance (MRD) codes instead of maximum distance separable (MDS) codes.

II. BACKGROUND AND RELATED WORKS

Key management has been considered the most fundamental mechanism since the beginning of the research on network security. Based on the characteristic of WSNs, the practical key management scheme for WSNs would be key pre-distribution approach. Key pre-distribution approach which belongs to symmetric encryption algorithm is that key information is distributed to all sensor nodes prior to deployment. There are several research work going on in key pre-distribution schemes. Some of the already existing schemes are discussed below.

In [2], Blom proposed a symmetric key generation scheme in which any two nodes can establish a pairwise secret key. The major issue in this scheme is, as long as an adversary compromises less than or equal to k nodes, uncompromised nodes are perfectly secure; when an adversary compromises more than k nodes, all pairwise keys of the entire network are compromised. This scheme consists of (k × N) matrix G of MDS codes over a finite field GF(q), whereas N is the size of the network and q > N. The matrix G has k linearly independent columns and it is public. During the key generation phase, (k × k) symmetric matrix D is calculated from finite field and it is kept secret. Then (N × K) matrix A is calculated as $A = (D.G)^T$. Key space is calculated as K = (A.G). As D is a symmetric matrix, K is also symmetric. During the key assignment phase, one row from A is randomly assigned to each node with its ID. If two nodes want to communicate with each other, they establish a key by exchanging the information stores in them. For example, consider two nodes; S_i and S_j want to establish a common key between them. The i^{th} row of matrix A is assigned to node S_i and j^{th} row to node S_j. These nodes will obtain public information, that is, $column_i$ and $column_j$ of generator matrix G during the key establishment phase. Node S_i will calculate its key as K_{ij} = rowi × $column_j$ and node S_j will calculate its key as $K_{ji} = row_j \times column_i$, as K is a symmetric matrix that means $K_{ij} = K_{ji}$. In [3], Eschenauer and Gligor proposed a random key pre-distribution scheme in which each sensor node receives a random subset of keys from a large key pool as the node's key ring before deployment, and stores them in its memory. After nodes have been deployed, two neighbouring nodes can find at least one common key in their key rings and use the key as their shared key. Based on this scheme, Chan,

Perrig and Song [4] proposed a q-composite random key pre-distribution scheme in which two nodes can establish a secret key if they have at least q common keys in their key rings. In [5], variant of [2] is proposed. It uses multiple D matrices to generate 'i' key spaces and then out of these 'i' key spaces, 'j' key spaces were assigned to each node. Those nodes that share common key spaces can establish a direct link between them. In [6], matrix-based random key pre-distribution scheme was proposed in which the keys are assigned to nodes on the basis of their positions. In [7], the authors have proposed a symmetric key generation and pre-distribution scheme, using a symmetric matrix and generator matrix of maximum rank distance (MRD) codes based on [8]. In [9], the authors have proposed a combination of matrix decomposition technique and polynomial-based key pre-distribution approach. It guarantees that any two sensor nodes can find a common key between themselves by using a pool of polynomial formed in the symmetric matrix format and matrix decomposition.

III. GENERATOR MATRIX

A generator matrix G_k of a MRD code is defined by,

$$G_k=\begin{bmatrix} g_1 & g_2 & \dots\dots & g_n \\ g_1^{[1]} & g_2^{[1]} & \dots\dots & g_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{[k-1]} & g_2^{[k-1]} & \dots\dots & g_n^{[k-1]} \end{bmatrix} \quad (1)$$

Where $g_1, g_2, \dots, g_n$ are any set of elements from F_q that are linearly independent over F . The notation $g^{[i]} := g^{q^{\wedge}}$ (i mod N) means the i^{th} frobenius power of g. A code with the generator matrix as shown in (1), is referred to as a (n, k, d) code, where n is the code length, k is the number of information symbols and d is the code distance.

IV. KEY GENERATION SCHEME

The symmetric key generation scheme is divided into three phases. The first two phase's namely key generation, key assignments are done before the node deployment whereas the third phase which is key establishment is done after node deployment.

A. Key Generation

The first step in this phase is to generate a finite field F_q of q elements which is used to generate different matrices from which key spaces for each node will be calculated. This phase is composed of four steps which are given below.

Step 1: A symmetric matrix D of size (k × k) is generated from the finite field F_q where the value of k is kept constant. Depending on the user application, the value of k can vary. Reason for keeping the value of k as constant is to reduce the memory consumption which is explained in Section 5.1.

Step 2: Let the total number of nodes in the network be 'n'. Divide n nodes into t groups having (k -λ) nodes in each group where λ is any number ≥1. Reason for choosing group size (k – λ) is to prevent attackers from gaining information stored in the nodes which is explained in Section 5.2.

Step 3: For each group generator matrices is generated as shown in (1). In the generator matrix, only the element of first row is linearly independent and the rest of the rows are generated by taking the frobenius power of the corresponding element in the previous row. For generating generator matrix $G_{1\ of}$ size (k × N) for the first group, a generating vector g_1 of N elements will be generated. For the rest of (t - 1) matrices, we randomly choose β_i elements from the finite field where i = 1, 2... t – 1 and multiply it with any one of the already existing generating vectors, that is

$g_2 = \beta_1 g_1$

$g_3 = \beta_2 \{g_1, g_2\}$

.

.

$g_t = \beta_{t-1} \{g_1, g_2, \dots, g_{t-1}\}$

After calculating t generating vectors, generator matrices will be calculated for each group. All the generating vectors g_i and multiplying coefficients β_i will be kept secret.

Step 4: For each group, A_i matrices is calculated as

$A_i = (DG_i)^T$

Where i = 1, 2, . . . , t.

B. Key Assignment

In this phase, one row from the matrix A_i will be randomly assigned to each node in the i^{th} group where i =1, 2. . . t and one element from the matrix G is also stored in each node. As there are (k - λ) nodes in each group and also N ≥ k, there will be always (N – k + λ) unassigned rows in each of the A_i matrices. Some of the information about matrix G will also be stored at each sensor node. It is shown in (1) that any column of generator matrix G can be calculated if first element of that column is known. The column position of the stored element of G_i must be the same as the row number of the A_i, for example, if a node S_i in the i^{th} group is assigned m^{th} row from the A_i matrix, then it will also have the first element of the m^{th} column of the G_i. To calculate the link key, each node from the i^{th} group will have one row from the matrix A_i and first element of the column of the matrix G_i, that means each node is needed to have (k + 1) × τ bits, τ is the number of bits required to store one element of the GF(q).

C. Key Establishment

After the deployment of nodes, secret key must be established between two nodes if they want to communicate with each other. This process consists of four steps.

Step 1: Each node will send its node ID and the seed element to the node to which it has to communicate.

Step 2: Once the node receives the ID and the seed element, it will calculate the column of matrix G from the seed value, by raising it to k^{th} element.

Step 3: After calculating the column, the node will multiply this column with its own row from matrix A. The resultant value is the secret key between the communicating nodes. Suppose, there are two nodes, S_i and S_j, that want to establish a link key between them. Node S_i has row_i, an i^{th} row of matrix A_i and a seed of the i^{th} column from the matrix G_i stored on it and node S_j has row_j, j^{th} row of matrix A_j and a seed of the j^{th} column from the matrix G_j stored on it. These nodes will first exchange their node IDs and seeds for the columns with each other. Node S_i will calculate $column_j$ by raising the seed it received to k^{th} element and then it will calculate the link key as $K_{ij} = row_i \times column_j$. Similarly, node S_j will calculate $column_i$ by raising the seed it received to k^{th} element and then will calculate the link key $K_{ji} = row_j \times column_i$. As D is a symmetric matrix, therefore matrix K will be symmetric as well that means $K_{ij} = K_{ji}$. In this way, any two nodes in the network, irrespective of their groups can generate a common link key between them by just exchanging only one message containing their seed for column from matrix G and node ID. Mathematically,

$$\begin{aligned} K &= (AG) \\ &= (DG)^{TG} \\ &= G^T D^T G \\ &= G^T DG \end{aligned} \tag{2}$$

$$\begin{aligned} K^T &= (AG)^T \\ &= G^T A^T \\ &= G^T ((DG)^T)^T \\ &= G^T (G^T D^T)^T \\ &= G^T DG \end{aligned} \tag{3}$$

As right-hand sides of (2) and (3) are equal, therefore K is a symmetric matrix.

Step 4: The resultant key value from the above step is applied to a Division Remainder hash function

$H(k) = k \% M$

Where M is a prime number and M < k. The result of this function is treated as a secret key for communication between two nodes.

V. ANALYSIS AND DISCUSSION

A. Memory analysis

As wireless sensor networks have severe constraints in their memory, only minimum amount of information must be stored in sensor nodes. Compared with other pair wise scheme, our scheme is very efficient in terms of memory consumption. To calculate a secret key for any node in the network, only one row from a (N × k) matrix A and one element of matrix G and one hash function are needed to be stored at each node. Therefore only (k + 2) × τ bits are required to be stored at each node, where τ is the number of bits required to store one element of GF(q). In [7] as the value of k increases, memory required at each node increases so we kept the value of k as constant.

Fig. 1 shows the memory comparison of the proposed scheme with [5,6,7]. We have compared our scheme with those matrix based key pre-distribution scheme that use Blom's idea. The results show that proposed scheme requires less memory as compared with other schemes. The memory required at each node is absolutely static and it does not increase as the network size increases.

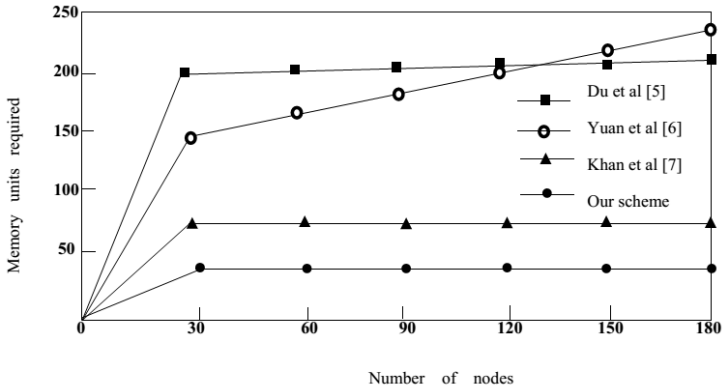

Figure 1. Memory required at each node

B. Security Analysis

One of the major problems with Blom scheme [2] is that communication among the uncompromised nodes is secure only when less than k nodes are compromised. When at least k nodes are compromised then whole network is compromised because each node carries a row from (N× k) matrix A = $(D.G)^T$ and matrix G is public as well. When k nodes are compromised then an adversary may construct a new (k×k) A_{adv} matrix and then multiply it with (k ×N) public matrix G to get a (k ×N) matrix K_{adv}. If an adversary compromises k nodes and he/she is able to compromise all nodes from same group then still he will have only k columns of matrix G and he/she will be able to get only a (k× k) matrix K, so communication among the rest of the n - k nodes in that particular group is still uncompromised. To compromise a whole group an adversary would need to compromise all n nodes. Even if an adversary has compromised one whole group and recovered the generating factor for matrix G, the first step he/she needs to do is to re-order the elements of generating vector because we have randomly assigned the rows from matrix A and seeds from the matrix G to each node and there are N! Different ways with which we can choose n linearly independent columns from the finite field. Furthermore in order to find other G matrices, he/she needs to find the t multiplying coefficients from finite field consisting of 2^{N-1} elements which is difficult for a large N.

The process of obtaining D matrix if there are k nodes in each group is given below,

1. Construct two (k × k) matrices A_{adv} and G_{adv} from the information stored on k nodes.

2. Take the transpose of A_{adv}, which will give

$$A_{adv}^T = (DG_{adv})$$

3. As generator matrix G is a rectangular matrix of order (k × N). The inverse of rectangular matrices does not exist but G_{adv} is a (k × k) square matrix and its inverse does exist. So the adversary needs to find inverse of G_{adv}, which can be easily calculated. Suppose it is G_{adv}^{-1}

4. Multiply G_{adv}^{-1} with A_{adv} and it will give the matrix D.

$$A_{adv}^T = (DG_{adv}) \quad (4)$$

From (4), D can be calculated as

$$D = (A_{adv}^T G_{adv}^{-1})$$

It means that the only way an adversary can break the system without capturing all nodes in the network is to compromise at least k nodes from same group, but we have only k –λ nodes in the each group.

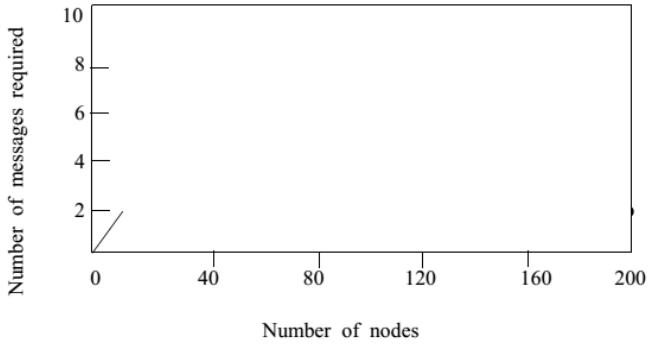

Figure 2. Number of messages required to calculate link key

C. Network Scalability

A good pre-distribution scheme must be scalable and adaptive to changes in the network. The most important thing is that the information which is already stored in the sensor nodes must not be changed and it must work flawlessly when new nodes are added to the network. Newly deployed nodes must be able to establish secret keys with the previously deployed nodes and vice-versa. Our scheme supports greater network scalability and connectivity. Since wireless sensor networks have limited energy, communication between the nodes must be minimised. Our scheme requires only two messages to be transmitted in order to exchange information between the nodes even if network size increases as shown in Fig. 2. This reduces the communication overhead.

VI. CONCLUSION

We have proposed a symmetric key generation and pre-distribution scheme based on MRD codes. The size of symmetric matrix is kept constant which considerably reduce the memory required at each node as compared with other matrix – based key pre-distribution schemes. Our scheme provides 100% global connectivity and network scalability as well. New nodes can be added at any time and previously deployed nodes can generate key with newly deployed and vice-versa without any information update. The key advantages of our scheme are less memory usage, 100% network connectivity, excellent scalability and low communication overhead without compromising on authenticity, integrity and confidentiality of security.

REFERENCES

[1] Zoran S. Bojkovic, Bojan M. Bakmaz, and Miodrag R. Bakmaz: "Security Issues in Wireless Sensor Networks", International Journal of Communications, vol. 2, pp. 42-47, 2008.

[2] Blom, R,"An optimal class of symmetric key generation system", Advances in Cryptology Proceeding EUROCRYPT, vol. 8, pp. 335–338, 1985.

[3] Eschenauer L, Gligor V,"A key management scheme for distributed sensor networks", Proceeding ninth ACM Conference on Computer and Communication Security, vol. 3, pp. 41–47, 2002.

[4] Chan, H, Perrig A, Song, D."Random key pre-distribution schemes for sensor networks", IEEE Symposim on Research in Security and Privacy, Carnegie Mellon University, vol. 27, pp. 197–213, 2003.

[5] Du W, Deng J, Han, Y.S Varshney, P.K,"A pairwise key pre-distribution scheme for wireless sensor networks", ACM Transaction Information System Security, vol. 8, pp. 228–258, 2005.

[6] Yuan T, Zhang S, Zhong Y,"A matrix-based random key pre-distribution scheme for wireless sensor networks", Proceeding Seventh IEEEInternational Conference on Computer and Information Technology, vol. 7, pp. 991–996, 2007.

[7] E. Khan, E. Gabidulin, B. Honary, H. Ahmed,"Matrix-based memory efficient symmetric key generation and pre-distribution scheme for wireless sensor networks", IET Wireless Sensor System, vol. 2, pp. 108–114, 2007.

[8] Gabidulin E, "The theory of codes with maximum rank distance", Problems of Information Transmission, vol. 21, pp. 1–12, 2007.

[9] MinghuiZheng, HuihuaZhou, Guohua Cui, "A LU Matrix-based Key Pre-distribution Scheme for WSNs", IEEE Symposim on Researchin Wirless Sensor Networks, vol. 23, pp. 345-357, 2008.