Journal of Computer Science 7 (7): 997-1002, 2011
ISSN 1549-3636
© 2011 Science Publications

Secure Group Key Management for Dynamic Sensor Networks

[1]N. Suganthi, [2]V. Sumathi and [1]R.S. Mohanapriyha
[1]Depertment of Information Technology,
Kumaraguru College of Technology, Coimbatore, India
[2]Depertment of ECE, Government College of Technology, Coimbatore, India

Abstract: Problem statement: Key management is an important cryptographic technique for providing security in a dynamic environment of sensor networks. To provide a secure communication among a group of sensor nodes keys should established in an efficient manner. We introduced an efficient and dynamic key management system for dynamic sensor networks. **Approach:** We pre deployed some of the keying materials in all the nodes and keys are calculated using keying materials and some random matrix which is distributed by Group Controller node. If there is any change in the member ship then re-keying process will be performed to change the old group key. **Results:** Our key management system eliminates the storage of long keys and it will improve the network resilience. It provides the forward and backward secrecy among the group of nodes. It reduces the computation overhead of group controller and also the communication overhead. **Conclusion:** Experimental results show that the proposed method perform very well for improving the success ratio of key establishment and enhance security while reducing the communication overhead and resource consumption.

Key words: Sensor networks, key predistribution, shared secret key, pair wise key, group key, dynamic environment, random matrix, dynamic key, adversary compromises, sensor node

INTRODUCTION

As sensor networks edge closer toward wide-spread deployment, security issues have became a central concern and are increasingly important. In fact, sensor networks cannot be used in practice if they are not secure, for example, in applications like emergency rescue and battlefield communication (Akyildiz *et al*., 2002); if no security mechanism is used, an adversary can easily thwart the network establishment. Symmetric key systems are still the major tools for communication privacy and data authenticity in most networks. To provide secure communication for any group of nodes using symmetric key cryptography, these nodes need to share a common secret key. In fact, a secure key management scheme is the prerequisite for the security in sensor networks. However, none of the existing key management schemes seem to be satisfactory for sensor network due to the unique properties of sensor networks. The challenge of designing key management protocols for sensor networks thus lies in establishing a secure communication infrastructure, before any routing fabric has been established and in the absence of any trusted authority or fixed server, from a collection of mobile nodes which share no pre-initialized secret information and have no prior contact with each other.

In sensor networks, key agreement is used to set up secret keys between them. There are three classes of methods namely trusted-server scheme, self enforcing scheme, key pre-distribution scheme (Du, 2003).

Security in sensor network has six challenges: (i) wireless nature of communication (ii) resource limitation on sensor nodes, (iii) very large and dense nodes, (iv) lack of fixed infrastructure, (v) unknown network topology prior to deployment, (vi) high risk of physical attacks. As a result, the physical security of the node becomes an important issue. Security encompasses a number of attributes that have to be addressed.

Availability: The property of a system or a system resource being accessible and usable upon demand by an authorized system entity.

Integrity: The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay)

Authentication: The assurance that the communicating entity is the one that it claims to be.

Confidentiality: The protection of data from unauthorized disclosure.

Corresponding Author: N. Suganthi, Department of Information Technology, Kumaraguru College of Technology, Coimbatore, India

Non-repudiation: Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

Key management process entails four basic functions, namely analysis, assignment, generation and distribution of network keys.

Key analysis: Keying requirements are analyzed to determine the required number of keys for the network as well as the number of keys needed by each node.

Key assignment: Mapping keys to individual parties. This may be static or dynamic depending on the key management solution employed.

Key generation: The generation of administrative keys may take place once or multiple times over the life span of the network. The generation of communication keys is the responsibility of the communicating parties.

Key distribution: Delivery of keys to their designated nodes after they have been generated and assigned to nodes.

Related works:

Blom scheme: Our scheme builds on Blom's key pre-distribution scheme (Blom, 1985). Our results show that the resilience of our scheme is substantially better than Blom's scheme as well as other random key pre distribution schemes. Blom (1985), Blom proposed a key pre-distribution scheme that allows any pair of nodes to find a secret pair wise key between them. Compared to the (N-1) pair wise key pre distribution scheme, Blom's scheme only uses λ+1memory spaces with λ much smaller than N. The tradeoff is that, unlike the (N-1) pair wise key scheme, Blom's scheme is not perfectly resilient against node capture. Instead it has the following λ-secure property: as long as an adversary compromises less than or equal to λ nodes, uncompromised nodes are perfectly secure; when an adversary compromises more than λ nodes, all pair wise keys of the entire network are compromised. The threshold λ can be treated as a security parameter in that selection of a larger λ leads to a more secure network. This threshold property of Blom's scheme is a desirable feature because an adversary needs to attack a significant fraction of the network in order to achieve high payoff. However, λ also determines the amount of memory to store key information, as increasing λ leads to higher memory usage. The goal of our scheme is to increase network's resilience against node capture by efficiently using the available node energy.

Carpy scheme: In Blom's scheme (Blom, 1985), communications become insecure after more than λ sensor nodes are compromised. The reason for this is that the row vector in the Ai, in the sensor node i is directly related to the private matrix D. Hence, after collecting a sufficient number of row vectors of A, the adversary is able to construct the private matrix D by solving a system of linear equations since G is publicly known. An idea to enhance the security is to break the direct relation between D and A by adding certain random noise$_1$ on A to distort Blom's key (Yu *et al*., 2010). However, if improper random noise is applied, either additional computation and communication are needed to extract the common bits of distorted Blom's key between two sensor nodes, or the common key cannot be found anymore (Yu *et al*., 2010).

Key management scheme for Distributed sensor networks: Eschenauer and Gligor (2004) proposed a random key predistribution scheme which consists of three phases namely key pre-distribution, shared key discovery and path key establishment. The basic idea of their scheme is randomly selecting and storing a subset of communication keys from a very large size key pool into each wireless sensor node's memory before it is deployed. Each node uses a key discovery process to exchange key information with its neighbors after deployment. If two neighbor nodes share one or more common keys in their memories, they can establish a secure communication link between them. Otherwise, two communicating nodes need to setup a path key with other intermediate nodes' participation.

Efficient pair wise key establishment and management scheme: In EPKEM, pair wise communication key is established through four phases: Setup key pre-assignment phase, common keys discovery phase, pair wise key computation phase and key ring establishment phase (Cheng and Agarwal, 2005). The first two phases are similar to the previous Eschenauer and Gligor method.

Pair wise key computation phase: After the common key discovery phase, each sensor node knows its neighbor node's ID and their shared common keys. Since all the pre-loaded setup keys are picked from the same key matrix K, the same key may be stored in different nodes. That means, when some nodes are captured, keys stored in non-captured nodes may be compromised too. To address this problem, they establish a new pair wise communication key for each pair of neighbor nodes instead of using the shared common keys directly. The new pair wise communication key can be calculated based on the shared setup keys. Suppose node N_a and N_b are a pair of neighbor nodes and their shared setup keys are $K_{i,m}$ and $K_{l,j}$ (Cheng and Agarwal, 2005; Al-Talib *et al*., 2009; Maalla *et al*., 2009). To establish a private pair wise key which is unaware to other nodes, node N_a and node N_b compute their pair wise key using Eq. 1:

$$kN_a = k_{i,m} \oplus N_a \oplus k_{l,j} \oplus N_b \qquad (1)$$

Key ring establishment phase: Once a sensor node computed all corresponding pair wise communication keys with its neighbors, it erases all the pre assigned setup keys from its memory immediately to prevent the possible key compromising and node capture attack. Only the computed pair wise communication keys with its neighbors and the secret key shared with KDS are kept in the memory of each node, which compose the permanent key ring of a sensor node. A connected secure link network can be established when the above four phases are finished. Each sensor node can communicate with KDS and the calculated pair wise keys to authenticate and communicate with its proper neighbor nodes securely Cheng and Agarwal (2005).

Random key predistribution scheme using Probability Density Function (PDF): The Du *et al*. (2004) scheme uses deployment rectangles whose sizes strongly depend on the pdf of node deployment. This scheme exhibits better performance (connectivity and memory usage) and it keeps the sensor networks secure. However, it can only be applied to the group-based deployment model. Moreover, if the PDF of node deployment is not a two-dimensional normal distribution or if it changes in real time (e.g., the wind direction changes during deployment), this scheme does not appear to work. Advanced key pre-distribution scheme uses a key-position map and the PDF of node deployment. The key position map shows which key is assigned to which position (Huang *et al*., 2007); this is specified by coordinates in a two-dimensional coordinate system. The pdf of node deployment can be determined by physical laws or previous results. Similar to the Eschenauer-Gligor scheme, this scheme consists of three phases. The last two phases are the same as in the Eschenauer and Gligor (2004) scheme. Therefore, it was focused on the first phase: the key pre-distribution phase and this scheme can be applied to various deployment models (Huang *et al*., 2007) (e.g., deployment at irregular intervals, mixed deployment by helicopters and cars).

Polynomial based dynamic key generation: Polynomial based key pre-distribution scheme (Blundo *et al*., 1993; Sudha *et al*., 2009) distributes a polynomial share (a partially evaluated polynomial) to each sensor node by using which every pair of nodes can generate a link key (Huang *et al*., 2007). Symmetric polynomial P(x, y) (P(x, y) = P(y, x)) of degree λ is used. The coefficients of the polynomial come from GF (q) for sufficiently large prime q. Each sensor node stores a polynomial with λ + 1 coefficient which come from GF (q). Sensor node S_i receives its polynomial share of $f_i(y) = P(i, y)$. S_i (resp. S_j) can obtain link key $K_{i,j} = P(i, j)$ by evaluating its polynomial share $f_i(y)$ (resp. $f_j(y)$) at point j (resp. i). Every pair of sensor nodes can establish a key. The solution is λ-secure, meaning that coalition of less than λ+1 sensor nodes knows nothing about pair-wise keys of others (Camptee and Yenar, 2005).

MATERIALS AND METHODS

We assume that N low cost resource constrained sensor nodes are deployed over the sensing region and no prior deployment knowledge about the nodes location is known. Wireless Sensor Networks are dynamic in the sense that radio range and network connectivity changes by time. Sensor nodes dies and new sensor nodes may be added to the network. Each sensor node is assumed to have a unique ID, which could be arbitrarily chosen in a general purpose sensor node or fixed in a specific sensing hardware. Among these sensor nodes one is assumed to be a controller node which will act as temporary head. Whenever any new node is deployed it will register to this controller node. If any changes in the membership this controller node will initiate the random matrix generation and distribute that along with the hello messages to all the nodes. And individual nodes will calculate the new key and use it for communication. In addition to static networks, mobile nodes are also allowed in our methods so that partial or entire nodes could have mobility.

Secure and dynamic key generation system: In Blom scheme, the private key matrix which is kept secret is multiplied directly with the public vector or matrix. Such method is not much secure because linear equations can be formulated and by solving those equations the secret matrix could be obtained. In some of the schemes, a random noise is used to disturb the direct relation between the private secret matrix and the public vector.

By doing so the security can be increased but generating appropriate random noise is a difficult task and has computation overhead. The other methods need more complex calculations but the sensor nodes have limited processing power and hence it won’t be much efficient.

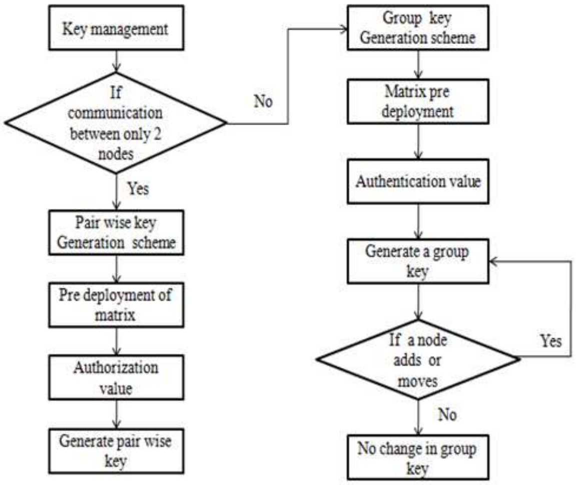

Fig. 1: System model

In our system, we consider two cases where communication is between two nodes and in other case the communication is among group of nodes. In both cases we consider a set of keying materials to be pre loaded in nodes. Then the authentication value is generated using hash function. It can be verified and the pair wise keys and group keys are generated. The flow of the model is shown in Fig. 1 in which two different scenarios are considered.

In our scheme, the direct use of matrix is not done and another linear independent matrix is introduced and certain computations are done to generate another second level secret matrix in a Galois finite field (GF (q)). An authentication code is also used which is kept secure for the nodes in the network and the nodes are assumed to be tamper proof. We also assume tamper-detection via sensor node shielding that erases the key matrix and authentication codes of captured nodes. In the single mission key scheme, all communication links are compromised, whereas in the pair wise private key sharing, all n-1 links to the captured unshielded node are compromised. Using all these we can generate the pair-wise secret keys.

For generating group keys we can consider a different scenario where the matrixes are preloaded and it can be deployed in nodes. Using that matrix we can generate an authentication value and a resultant value which is used to generate the group key. Since the network is dynamic or it changes often, rekeying should be done to provide forward and backward secrecy. A new matrix is generated for this purpose and is sent along with a hello message by the group controller node.

Algorithm: Pre loading of keying materials that is specified below are loaded into all the nodes in network before their deployment.

The four matrices that are loaded into the nodes are:

1.1 Symmetric matrix (A), where values of the matrix above and below the diagonal are similar
1.2 Linear independent matrix (B), where the value of each row and column does not depend on the other values of rows and columns
1.3 Random matrix(R) which consists of a column of random values
1.4 Public identifier (I1) which is a column vector, allotted to all individual nodes

Key establishment by nodes:

2.1 Compute F= (A*B), F1=F’
2.2 Final secret matrix S=F1*B
2.3 R1=sum(R), where R is a random matrix.R1 is obtained by adding the values of matrix R
2.4 Hash function of (authorization code), the resultant value of hash function and R1 is multiplied and is verified by neighbor nodes and controller node. The hash function involves certain ex-or and left shift operation
2.5 If authenticated then pair wise keys are generated in a finite Galois field GF (q):

(S*I1)’*I2, (S*I2)’*I1

Group key generation:

3.1 Symmetric matrix (A), linear independent matrix(B) and random matrix(R) are multiplied to generate a resultant matrix

Thus resultant matrix=A*B*R

3.2 Resultant matrix values are added to calculate the intermediate value. Hash function is applied to authentication code, which is given by Hash(authentication code)
3.3 After verification of authentication, group keys are generated using resultant value and hash value, which is given by product of sum of resultant value and hash value

Group key= sum (resultant matrix value) * Hash(authentication code)

3.4 If any node leaves, new random matrix is generated by group controller node and sent along with the hello message and re-keying is done. Using the new random matrix new resultant matrix is generated and in turn a new group key will be generated

Description: Initially we generate a secret symmetric matrix (A), of any number of rows and columns. This matrix is pre deployed to all the nodes that come into the network. The direct relation between the secret matrix and the public identifier is changed and we introduce another linear independent matrix(B). These two matrix are multiplied and transpose is taken for the resultant matrix and multiplied again with the linear independent matrix to generate a second level symmetric matrix. Each node is given a public identifier as matrix. To generate pair wise key between any pair of nodes they exchange their public identifiers and it is multiplied with the second level symmetric matrix. Then nodes are verified for authentication and if they satisfy the authentication conditions then the secure pair wise keys are generated and the nodes can communicate with one another. This method increases the network security when compared to the Blom scheme.

RESULTS

In Fig. 2 we compare the memory complexity of existing pair wise scheme and our proposed scheme. In pair wise scheme the memory used is 2(N-1) where N is the no. of nodes in network, where a pool of matrix values is used. In proposed scheme, only 2 (n X n) matrix where n is 4 or 5 and a random matrix with less number of values is used, which decreases the memory storage when compared to the previous scheme.

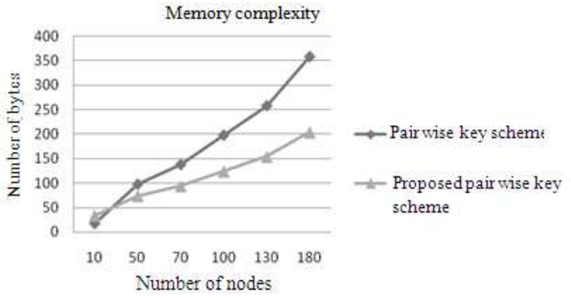

Fig. 2: Memory complexity

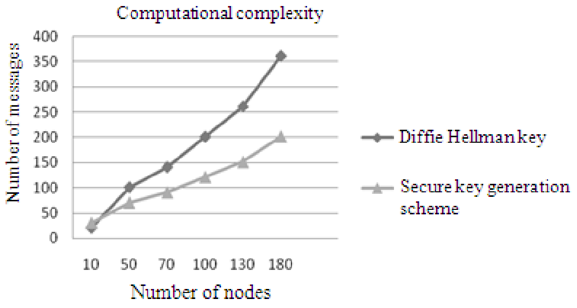

Fig. 3: Computation overhead

In our proposed work we use three pre defined matrix namely symmetric matrix, linear independent matrix and a random matrix. Hash function is applied to authentication code to obtain an authentication value. This value is manipulated along with the resultant value of all matrixes which provides the group key. In case of rekeying we generate a new random matrix and are sent along with the hello message by the group controller. This reduces the communication overhead. Whenever a node moves out of the network we generate a new group key.

Diffie Hellman key exchange uses (2N+1) no. of messages to calculate the group key which needs more computation power but in our proposed scheme we use (3+N) no. of messages to compute group key which reduces the computation power and is shown in Fig. 3.

Whenever a node leaves the network a new group key is computed for the network and whenever a node joins the network its authentication value is verified and group key is computed for it.

In pair wise key for each link it needs to establish a new key value which drastically increases the number of hops and in group key scheme no of hops will be reduced because a single key value is used for a set of nodes. This reduces the communication cost and is shown in Fig. 4.

DISCUSSION

Blom scheme the network resilience is low because the secret matrix is used directly and by obtaining the pair wise keys of some λ nodes certain equations can be formed and the secret matrix can be obtained. Hence the network can be compromised easily. But in our proposed scheme the initial secret matrix is computed with another matrix to obtain a second level symmetric matrix which provides more security and the secret matrix cannot be compromised. It increases the network resilience and can withstand up to N nodes.

By generating pair wise keys each and every individual link in the network becomes secure and if any link is compromised it doesn't affect the entire network.

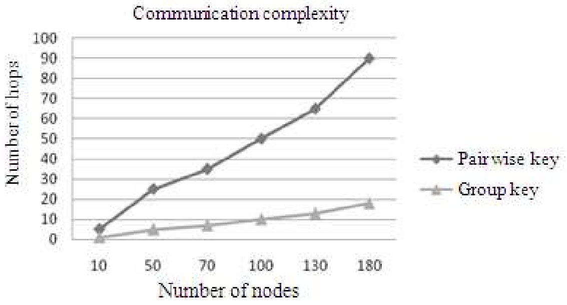

Fig. 4: Communication overhead

In pair wise keys, to transmit a data it should be passed among the secure channels which exist between the two nodes. By doing so, to reach the destination it needs more number of hops which increases the communication overhead. The number of encryptions and decryptions should be done for each pair of nodes which also increases the computation overhead. To overcome these problems group keys can be generated. Whenever a new node enters or leaves the network a new group key is generated.

Using Diffie Hellman key exchange method individual pair wise key established and using that group key is computed. Every node has its own private key. The public key of the node is exchanged among them to compute the secret pair wise key. This leads to more computation because it involves more exponentiation. Let na, nb be the private keys of node A and B respectively. P is the prime number and g is the primitive root of that number. Now using this information we can calculate the public keys as $Pa=g^{na}$ mod P, $Pb=g^{nb}$ mod P. Then by exchanging the public keys we can compute the secret group key in A as Pb^{na} mod P and in B as Pa^{nb} mod P. Here more computations are involved.

CONCLUSION

In this paper, we have seen various key pre-distribution schemes for sensor networks. We have discussed in various aspects like how those scheme works and what methods are used in it and the issues of those existing schemes and finally discussed the proposed method of generating a secure and dynamic key pair wise and group wise key which is efficient for the sensor networks.

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. A survey on sensor networks. IEEE Commun. Mag., 40: 102-114. DOI: 10.1109/MCOM.2002.1024422

Al-Talib, S.A., B.M. Ali and S. Khatun, 2009. An approach to improve the state scalability of source specific multicast. Am. J. Applied Sci., 6: 1347-1351. DOI: 10.3844/ajassp.2009.1347.1351

Blom, R., 1985. An optimal class of symmetric key generation systems. Proceedings of the EUROCRYPT 84, Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, Springer-Verlag New York, Inc. New York, USA., pp: 335-338. ISBN: 0-387-16076-0

Blundo, C., A.D. Santis, A. Herzberg, S. Kutten and U. Vaccaro *et al*., 1993. Perfectly-secure key distribution for dynamic conferences. Lect. Notes Comput. Sci., 740: 471-486. DOI: 10.1007/3-540-48071-4_33

Camptee, S.A. and B. Yenar, 2005. Key Distribution Mechanisms for Adhoc Networks-A Survey. Department of Computer Science, Rensselaer Polytechnic Institute, pp: 1-27. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.1246&rep=rep1&type=pdf

Cheng, Y. and D.P. Agarwal, 2005. Efficient pair wise key establishment and management in static wireless sensor networks. Proceedings of 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Nov. 7-7, Washington, DC., pp: 544-550. DOI: 10.1109/MAHSS.2005.1542842

Du, W., 2003. A pair wise key pre distribution scheme for wireless sensor networks. ACM Trans. Inform. Syst. Security, 8: 228-258. DOI: 1094-9224/05/0500-0228

Du, W., J. Deng, Y.S. Han, S. Chen and P.K. Varshney, 2004. A key management scheme for wireless sensor networks using deployment knowledge. Proceedings of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies, Mar. 7-11, USA., pp: 586-597. DOI: 10.1109/INFCOM.2004.1354530

Eschenauer, L. and V.D. Gligor, 2002. A key management scheme for distributed sensor networks. Proceedings of the 9th ACM Conference on Computer and Communication Security (CCCS’02), ACM, New York, NY, USA., 41-47. DOI: 10.1145/586110.586117

Huang, D., M. Mehta and D. Medhi, 2007. Modeling pair wise key establishment for random key predistribution in large scale sensor networks. ACM Trans. Networking, 15: 1204-1215. DOI: 10.1109/tnet.2007.896259

Maalla, A., C. Wei and H.J. Taha, 2009. Optimal power multicast problem in wireless mesh networks by using a hybrid particle swarm optimization. Am. J. Applied Sci., 6: 1758-1762. DOI: 10.3844/ajassp.2009.1758.1762

Sudha, S., A. Samsudin and M.A. Alia, 2009. Group rekeying protocol based on modular polynomial arithmetic over galois field GF(2n). Am. J. Applied Sci., 6: 1714-1717. DOI: 10.3844/ajassp.2009.1714.1717

Yu, C.M., C.S. Lu and S.Y. Kuo, 2010. Non interactive pair wise key establishment for sensor networks. IEEE Trans. Inform. Sec., 5: 556-569. DOI: 10.1109/TIFS.2010.2050140