ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

A Survey on Tree Structured Group Key Management Scheme to Provide Security for Sensor Networks

D.Lakshmi Anusha[1], Venkateswarlu Sunkari[2], Ato Daniel Abebe[3]

[1]Assistant Professor, Dept. of CSE, ASN Women's Engineering College, Tenali, Guntur Dist, Andhra Pradesh, India

[2]Assistant Professor, Centre for ITSC, Addis Ababa Institute of Technology, Addis Ababa, Ethopia

[3]HOD, Centre for ITSC, Addis Ababa Institute of Technology, Addis Ababa, Ethopia

ABSTRACT: Security is asignificant concern in sensor networks. Many applications in military, distributed information gathering etc., demand for Secure Group Communication (SGC) in sensor networks. The SGC requires common network-wide group key for confidentiality of control messages and data reports. The group key should be updated when a node is compromised. In this paper we propose a new key management scheme for group key computation and distribution which is based on tree structure. The proposed scheme minimizes storage as well as communication and computation cost of end user (i.e., sensor nodes). The complex encryption/decryption operations used to distribute new group key whenever a node is compromised are replaced by one way hash functions and simple XOR operations.

KEYWORDS: Secure Group Communication, Sensor Node, Hash Function, Group Key.

I. INTRODUCTION

Sensor networks are used in many applications like military sensing and tracking, environment monitoring, patient monitoring and tracking. Sensor networks usually consist of a large set of distributed low power sensors scattered over the area to be monitored. The sensors have the ability to gather data and process and forward it to a central node for further processing. The energy constrained nature of the sensor networks and deployment of sensor nodes in a hostile environment makes the problem of providing security to sensor networks challenging. In a sensor network operating in a battle field, we should encrypt each message from central node (sink node) and every data reports from sensor nodes to central node and messages exchanged among sensor nodes to protect the message from possible enemy eavesdroppers. This type of application demands Secure Group Communication (SGC) model in sensor net-works. Security of messages in such SGC model can beachieved by using common group key shared by the group of sensor nodes. One of the important feature of SGC in sensor networks is handling group dynamics (i.e., new nodes may enter the area at any time and some of the existing nodes may move out of the area or a node may be com-promised by adversaries).

When common group key is used for confidential communication, if a node is compromised, we need to change the group key, in order to achieve forward access control (i.e., the adversary should not be able to decrypt future messages). The process of updating the group key and distributing it to group members is called rekeying. Rekeying is required in SGC to ensure that only current sensor nodes in the group can communicate securely. Size of the rekey-ing message is nothing but number of encryptions required to distribute the new group key to the sensor nodes in the group securely.

The simple or naive method used to share the group key among the sensor nodes is by pre-loading it to each sensor node before deployment. But this method will not allow group dynamics.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

The other types of key management techniques which have been studied in general network environments are: a trusted-server scheme which depends on a trusted-server for key management between nodes, e.g., Kerberos [10], but this scheme is not suitable for sensor networks as it is difficult to assume the existence of such a trusted server in sensor networks. Limited computation and energy resources of sensor nodes make it undesirable to use public key algorithms, such as Diffie-Hellman key agreement [6] or RSA [16], as pointed out in [13].

The distributed group key management in sensor net-works is addressed in [18, 3, 11] which allow each sensor node to collaboratively generate and update local group keys and merged with other groups to generate a single network-wide group key. In distributed schemes as the size of the network increases then the communication and computation costs on each sensor node will also increase.

The topic of key management for multiparty communications in general networks are studied in [1, 5, 7, 8, 9, 12]. In [4] one of the efficient key tree based group key management technique called Logical tree Hierarchy (LKH) is discussed. A key update in this scheme requires $O(\log_2 N)$ messages where N is the size of the group. In this scheme each user has to store $\log_2 N$ keys (i.e., keys along the path from leaf to the root) and the key server has to maintain a tree of O(N) keys. The scheme proposed in [2] uses the LKH scheme and uses a binary tree, but with only two keys at every level. This reduces total number of keys at the server from O(N) to O(h) where h is the height of the tree. But storage at each user remains at $O(\log_2 N)$. The scheme discussed in [14] extends the scheme of [2] to m-ary tree instead of binary tree, which reduces the user side storage from $O(\log_2 N)$ as in [2] to $O(\log_m N)$. In tree based key management schemes each user shares a key called private key with the key server and key at the root of the tree is the group key which is shared by all users in the group. Other keys (other than private key and group key) are called auxiliary keys (key encryption keys) which are known only for certain subset of users and are used to encrypt new group key whenever there is a group membership change.

In this paper we propose a tree-based key management technique for SGC in sensor networks. The scheme uses m-ary tree and at each level m keys are maintained. When-ever a node is compromised new group key is selected and distributed to other nodes. The encryption keys that are required to send new group key GK^0 securely are computed as in [14]. The new group key is distributed to group members without performing any encryptions. The scheme proposed in this paper uses one way hash functions and simple XOR operations in order to distribute new group key (GK^0). The computation required at user level will be reduced since encryptions are replaced by hash functions and XOR operations. Also the auxiliary keys along the path are computed by the nodes on their own by using a simple computation as follows : F (auxiliarykey, newgroupkey) ← (Auxiliarykey) XOR (NewGroupkey). Since m-ary tree is used we are reducing number of levels of the tree which reduces storage at each user. Also at every level we are maintaining only m keys which reduces server side storage to $O(\log_m N)$. Since encryptions are replaced by hash functions and simple XOR operations, computation and communication cost incurred will also be reduced.

II.MOTIVATION

In [2] binary tree structure is used. When the group is large, the number of levels in the binary tree will be more which increases number of keys to be stored by each sensor node. Extending the scheme to m-ary tree will reduce the height of the tree reducing number of keys at each sensor node. At the same time we should consider storage at central node since the tree is maintained by central node i.e., number of keys at every level of the key tree. In [2] two keys are maintained at every level of the key tree, extending the scheme to m-ary tree will result in maintaining m keys.

For a group size n, if d is the height of the binary tree, it results in storing 2*d keys at the central node. For the same value of n, if d^1 is the height of the m-ary tree, then $m * d^0$ keys are to be stored at the central node. We can have therelation

$$n = 2^d = m^{d0}$$

$$\Rightarrow d^0 = d/\log_2 m$$

Number of keys at central node in m-ary tree in terms of d can be represented as $m * (d/\log_2 m)$, which illustrates that as m increases, number of keys at central node will increase, which violates our motto. Hence in order to maintain mini-mum number of keys both at sensor node and central node, following relation has to be satisfied:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

$(m * d/\log_2 m) \leq 2 * d$ which is true only if $m \leq 4$.
If central node is assumed to be a machine with large storage capacity then m value can be more which further reduces the storage at sensor nodes.

III.MODEL AND NOTATIONS

A Wireless Sensor Network is composed of n sensor nodes which are organized as m-ary balanced tree with sensor nodes at the leaf as shown in Fig.1. The tree is maintained by the central node. A GK (group key) which is at the root of the tree is used to encrypt the data traffic. Every sensor node shares a key with the central node called its private key used to communicate with the central node securely and other keys along the path (excluding private key and group key) are called as auxiliary keys (key encryption keys) which are used to encrypt the new group key. Whenever a node is compromised or new node enters the monitoring area central node updates the tree, computes new group key (GK^0) and distributes it to the existing nodes securely. The encryption keys (key encryption keys) required to distribute new group key GK^0 are computed using the method as in [14]. Using the encryption keys computed, new group key GK^0 is distributed to existing nodes without actually performing any encryption. The auxiliary keys will also change but new auxiliary keys are computed by each sensor node after receiving the new group key. Each sensor node is required to store all the keys along the path from leaf to the root.

Each node :

1. Has the capability to compute a one-way hash function H as in [17, 15].
2. Is able to update auxiliary keys after getting new group key using the function F as follows:

F (auxiliarykey, newgroupkey) ←− (Auxiliarykey)XOR (NewGroupkey).K_{10}, K_{11}, K_{12}, K_{13} are auxiliary keys at level 1.

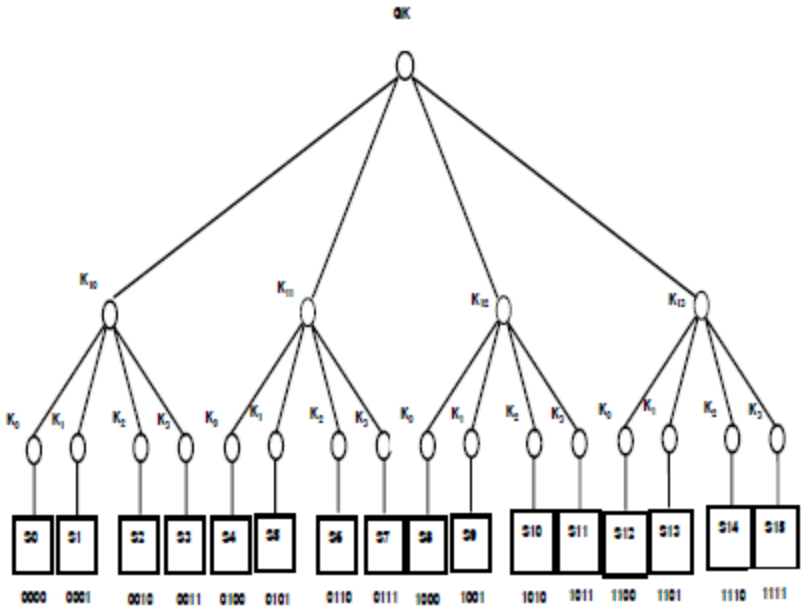

Fig 1. Key tree structure showing UIDs and keys of sensor nodes in the group, auxiliary keys and group key

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

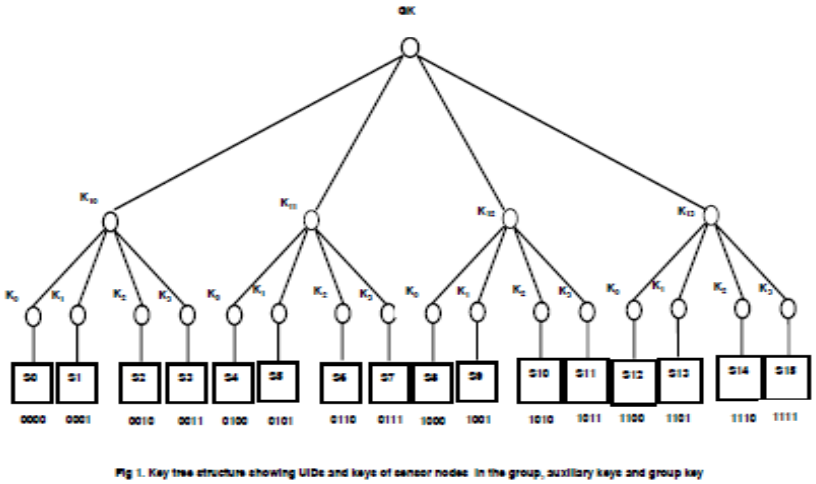

Fig 1. Key tree structure showing UIDsand keys of sensor nodes in the group, auxiliary keys and group key

GK is the group key shared by all the sensor nodes s_0 through s_{15}.m-ary tree: is a tree with the following properties:

- each interior node has at most m children
- each path from the root to a leaf has the same length

N: Total number of sensor nodes associated with the group and all nodes must be at the leaf level. Each node is assigned with Unique Identification Number (UID) which is a binary string of length x (where $x = \log_2 N$).

Subgroups: Each interior node containing at the maximum m children nodes forms one subgroup. Subgroups at level i are assigned with keys K_{i0} to $K_{i(m-1)}$ called Auxiliary keys at level i.

Keys: Individual keys of any subgroup are numbered from K_0 to K_{m-1} so that all nodes at position 0 of all subgroups are assigned with key K_0 and all nodes at position 1 of all subgroups are assigned with key K_1 and so on up to K_{m-1}.

From Fig. 1 the values of N, m, x, keys, auxiliary keys and group key are as follows: N=16 m=4 x=4

Keys:

Sensor nodes s_0, s_4, s_8, s_{12} are assigned with key K_0 Sensor nodes s_1, s_5, s_9, s_{13} are assigned with key K_1 Sensor nodes s_2, s_6, s_{10}, s_{14} are assigned with key K_2 Sensor nodes s_3, s_7, s_{11}, s_{15} are assigned with key K_3

IV.ENCRYPTION TO HASHING

By replacing encryption with Hash function the method used to communicate the new secret key is as follows: central node computes hash (as in [17, 15]) of a shared key (key known to central node and authorized node) say k_s i.e., $H(k_s)$ of the encryption key and XOR's with new secret key k^{new} to be communicated ($k^{comm} \leftarrow H(k_s) \oplus k^{new}$). Upon receiving k^{comm} nodes having k_s compute $H(k_s)$ and XOR's with k^{comm} which yields new secret key k^{new} ($k^{new} \leftarrow H(k_s) \oplus k^{comm}$).

In this method if two nodes with two different keys k_s and k_s^0 receive the same secret key in two individual communications, they can compute the hash of the other. For e.g., a node say g_x having k_s can recover $H(k_s^0)$ as follows : $H(k_s^0) \leftarrow (H(k_s^0) \oplus k^{new}) \oplus k^{new}$ where k^{new} is known to the node g_x (secret sent in earlier communication) and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

$H(k_s^0) \oplus k^{new}$ is eavesdropped. Using this hash i.e., $H(k_s^0)$ node g_x can decrypt the secret intended only for nodes having key k_s^0 that is sent in next communication. To avoid this every key that is hashed in the current communication is incremented by 1 and then used to take next hash value for next communication (i.e., $k_s^0 \leftarrow k_s^0{}_{+1}$ and then hashed to send next secret key). It is computationally infeasible for a user during the i^{th} application of this method to recover $k_s^0{}_{+i}$ even given $H(k_s^0{}_{+i}), \ldots H(k_s^0)$. The scheme is secure because of the properties of one-way hash functions explained in [17, 15].

V.HASH-BASED KEY DISTRIBUTION METHOD

The encryption keys computed using the method of [14] is used to communicate new group key to the existing nodes without actually performing any encryption.

For the tree in Fig.1, Let s_5 and s_6 are the nodes that are compromised, then the encryption keys computed us-ing the protocol 1 of [14] are KEK = { K_{10}, K_{12} , K_{13}, K_0, K_3}. Now new group key GK^0 is distributed to the reaining group members (i.e., nodes) using the hash method explained in the previous section. To communicate new group key GK^0 to nodes $s_0 \ldots s_3$ central node computes the hash of key k_{10} i.e., $H(k_{10})$ and XOR's this with new group key GK^0 which yields $K^{s0,\ldots,s3} \leftarrow H(k_{10}) \oplus GK^0$. Upon receiving this message nodes $s_0 \ldots s_3$ (knowing key k_{10}) compute $H(k_{10})$ and XOR's with $K^{s0,\ldots,s3}$ to get new group key GK^0 (i.e., $GK^0 \leftarrow K^{s0,\ldots,s3} \oplus H(k_{10})$). Similarly messages sent by central node to existing group mem-$_{bers\ are}K^{s8,\ldots,s11} \leftarrow_{H(k12) \oplus GK} 0,K^{s12,\ldots,s15} \leftarrow_{H(k13)}$
$\oplus GK^0$, $K^{s4} \leftarrow H(k_0) \oplus GK^0$ and $K^{s7} \leftarrow H(k_3) \oplus GK^0$. The nodes will compute the new group key GK^0 by XOR ing received message with the hash of the keys known to them. Group key GK^0 computed by nodes $s_8 \ldots s_{11}$ is

$GK^0 \leftarrow K^{s8 \ldots s11} \oplus H(k_{12})$, for nodes $s_{12}, \ldots, s_{15}$ it is $GK^0 \leftarrow K^{s12,\ldots,s15} \oplus H(k_{13})$, for node s_4 $GK^0 \leftarrow K^{s4}$
$\oplus H(k_0)$ and GK^0 computed by node s_7 is $GK^0 \leftarrow K^{s7}$
$\oplus H(k_3)$. The keys that are known to compromised nodes s_5 and s_6 are k_1, k_{11} (keys with s_5), k_2, k_{11} (keys with s_6). In the messages that are sent in clear by central

node to group members by using the hash of the encryption keys, none of the messages uses any of the keys that are known to compromised nodes. Hence using the keys of the compromised nodes it is not possible to get any information regarding new group key. In order to avoid attackers decrypting any message in the next time interval we per-form two operations. First, each remaining node along the path from the leaving point will compute new auxiliary key using the method F (auxiliarykey, newgroupkey) ←(Auxiliarykey) XOR (NewGroupkey). Second, every key used to compute the hash value is incremented by 1. In this scheme to communicate new group key securely we are not using any encryption instead all communications are by using hash values and XOR operations which will reduce the communication overhead i.e., rekeying cost is reduced.

VI.CONCLUSION

Sensor nodes move from one monitoring area to another very frequently and deployment of sensor nodes in a hostile environment allows node capture (node compromise). In order to achieve message confidentiality in such an environment we require a group key and the group key should be updated whenever a node is compromised. The tree based key management scheme proposed in this paper computes the new group key and distributes it to the current group members is efficiently in terms of storage, communication and computation. At central node storage is reduced to $O(\log_m N)$ from O(N) as in [4] and $O(\log_2 N)$ as in [2]. Since m-ary tree is used in our scheme it reduces the height of the tree there by reducing the number of keys to be stored at each sensor node. The storage at each sensor node is reduced to $O(\log_m N)$ from $O(\log_2 N)$ as in [2]. New group key is distributed to the existing nodes using hash functions and XOR operations.

REFERENCES

[1] A. Bellardie, "Scalable Multicast Key Distribution", RFC 1949, May 1996.
[2] I.Chang, R.Engel, D.Kandlur, D.Pendarakis and D.Daha. "Key management for secure internet multicast using Boolean function minimization

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

technique". ACM SIGCOMM'99, March 1999.

[3] A.Chandha, Y.Liu and S.K.Das, "Group key distri-bution via local collaboration in wireless sensor net-works", in IEEE International Conference on Sensor and AdHoc Communications and Networks(SECON), 2006.

[4] Chung Kei Wong, Mohamed Gouda, and Simon S Lam, "Secure Group Communication Using Key Graphs", Proceedings of ACMSIGCOMM, Vancouver, British Columbia, September 1998.

[5] Debby M. Wallner, Eric J. Harder, Ryan C. Agee, "Key Management for Multicast: Issues and Archi-tectures", Informational RFC, draft-Wallner-key-arch-ootxt, July 1997.

[6] W.Diffie and M.E.Hellman, "New directions in Cryp-tography",IEEE transactions on Information The-ory",vol.22, pp.644-654, November 1976.

[7] H.Harney, C.Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", RFC 2094, July 1997.

[8] H.Harney, C.Muckenhirn, "Group Key Management Protocol (GKMP) Specifications", RFC 2093, July 1997.

[9] D.McGrew and A. Sherman. "Key establishment in large dynamic groups using one way func-tion trees".. Available athttp://www.cs.umbc.edu/sher-man/papers/itse.ps, May 1998.

[10] B.C.Neuman and T.Tso, "Kerberos: An Authentica-tion service for Computer Networks",IEEE Commu-nications, vol. 32, no.9, pp.33-38, September 1994.

[11] B.Panja, S.K.Madria, and B.K.Bhargava, "Energy and communication efficient group key management pro-tocol for hierarchical sensor networks", in IEEE In-ternational Conference on Sensor Networks, Ubiqui-tow, and Trustworthy Computing(SUTC),IEEE Com-puter Society, 2006.

[12] A.Perrig, D.Song and J.Tygar, "ELK: A new protocol for efficient large-group key distribution".In Proceed-ings of the 2001 IEEE symposium on Security and Pri-vacy, 2001.

[13] A.Perrig, R.Szewczyk, V.Wen, D.Cullar and J.D.Tygar, SPINS: Security Protocols for Sen-sor Networks", in Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),Rome, Italy, pp.189-199, July 2001.

[14] A.S.Poornima, R.Aparna and B.B.Amberker. " Stor-age and Rekeying Cost for Cumulative Member re-moval in Secure Group Communication ", Interna-tional Journal of Computer Science and Network Se-curity,Vol. 7 No. 9 pp. 212-218, 2007.

[15] R.Rivest. The MD5 message-digest algorithm. RFC 1321, April 1992.

[16] R.L.Rivest, A.Shamir, and L.M.Adleman, " A method for obtaining digital signatures and Public-key cryp-tosystems",Communications of the ACM, vol.21, no.2, pp.120-126, 1978.

[17] N.F.P.180-1. Secure hash standard. Draft, NIST, May 1994.

[18] W.Zhang and G.Cao, "Group rekeying for filter-ing false data in sensor networks: a predistribution and local collaboration based approach", in Infocom IEEE,pp. 503-514,2005.