KEY MANAGEMENT SCHEME FOR SECURE GROUP COMMUNICATION IN WSN WITH MULTIPLE GROUPS

H.S.Annapurna[1] and M.Siddappa[2]

[1]Dept. of Computer Science & Engg.,
Sri Siddhartha Academy of Higher Education, Tumakuru, India.
hsassit@gmail.com
[2]Dept. of Computer Science & Engg.,
Sri Siddhartha Institute of Technology, Tumakuru, India.
siddappa.p@gmail.com

ABSTRACT

Security is one of the inherent challenges in the area of Wireless Sensor Network (WSN). At present, majority of the security protocols involve massive iterations and complex steps of encryptions thereby giving rise to degradation of quality of service. Many WSN applications are based on secure group communication. In this paper, we have proposed a scheme for secure group key management with simultaneous multiple groups. The scheme uses a key-based approach for managing the groups and we show that membership change events can be handled with less storage, communication and computation cost. The scheme also offers authentication to the messages communicated within and among the groups.

KEYWORDS

Group key, Key management, Sensing Node, Secure Group Communication, Key Tree.

1. INTRODUCTION

Wireless Sensor Network is a collection of sensor nodes with limited capabilities in terms of battery, computation, storage etc. The data that flows in among the sensor nodes in WSN consists of physically captured data from the readings of sensors, a mobile code, security using key management techniques, and location information of the sensor nodes. Owing to the lesser amount of obtainable of computational origin in the miniature sensor nodes and wireless communication social, WSN endures from probable security threat aspects [1]. There are basically two types of attacks in sensor network, e.g., active and passive attacks [2]. The malicious nodes can enhance their attacking capabilities by intruding the private information from mobile codes as well as by accessing the information pertaining to the positioning of the nodes [3]. Using various eavesdropping techniques, it is possible for the malicious node to incorporate malicious programs on the mobile code and thereby spreading the malicious mobile code in the entire network. The malicious node can also use the position information to identify

David C. Wyld et al. (Eds) : CSEN, SIPR, NCWC - 2016
pp. 91–101, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.61008

the best node to invoke their attacks thereby potentially making security breach. Owing to the wireless medium of communication in WSN, it is very challenging task to identify the malicious nodes and design a security policy to deny the access in the network. The malicious nodes are quite capable enough to access the entire network using potential computers and sophisticated communication equipments. The malicious nodes can also seed themselves in the network environment without even getting caught [4]. It is said that sink is considered as the most reliable core of the wireless sensor network that stores significant information about the security protocols, readings of sensors, and routing information. These are very critical in group communication. In small scale sensor network, it is easier to capture the data, process it, and forward to sink. But in random and dynamic network of large size, it usually doesn't go by single hop communication. The nodes are formulated in groups, where each group member interacts with other group member to forward the processed data from one point to another. The process of data aggregation completely fails without group communication. Hence, it is very important that a robust security technique is to be developed to address the security issues in group communication system in WSN. Cryptography [5] is the most frequently adopted technique to incorporate security while performing group communication in WSN.

However, conventional cryptographic algorithms like SHA, AES, although have good security features, suffer from limitations too. Hence, keeping all these issues in mind, the paper introduces a scheme for secure group key communication with multiple groups. Remaining part of the paper is ordered as follows: Section 2 discusses background of research work followed by key management scheme in section 3. Authenticated group communication is presented in section 4 and section 5 summarizes the paper.

2. BACKGROUND

The study towards secure group communication is more than a decade old and there are various techniques that have been introduced by the various researchers. This section discusses some of the recent studies found in standard research manuscript that focuses on i) secure group communication and ii) key distribution mechanism.

Cheikhrouhou et al. [6] have discussed a protocol for ensuring secure group communication using elliptical curve cryptography over ring based topology of wireless sensor network. The authors have discussed their outcomes considering storage cost which was found to be efficient compared to existing techniques. However, the limitation of the scheme is the dependency of key storage of size 160 bits. Wang et al. [7] have proposed a predistribution policy considering hexagonal grids consisting of groups and keys. Miettinen et al. [8] have presented a security protocol by incorporating an authenticated pairing system based on key context. Furtak and Chudzikiewicz [9] have used asymmetric key pair as well as electronic signature to provide secure authentication in wireless sensor network. Xi et al. [10] have presented a key estimating process that is done in faster manner as compared to attacker. However, various attackers have various patterns of generating attacks, the authors have not discrete mentioned the names of the attack. Moreover the outcomes of the study were not found to be benchmarked.

Hence, it can be seen that there exists various security protocols in the research papers with advantages and limitations. The prime trade-off found in all the study is dependency of broadcasting the key. We comment that broadcasting of the key is very sensitive operation and is highly prone to capture if proper encryption scheme is not implemented. Another trade-off found

is majority of the schemes are based on enhancement of conventional cryptographic scheme with less novelty in mathematical approaches. The third trade-off seen in all the studies is about the key sizes, which is 128, 216, 160, or 512 bits. Although the key sizes seem to be smaller but as majority of the existing approaches store this, grossly the sizes of the matrix holding the keys becomes eventually larger.

Many schemes for group key management have been proposed in the literature for WSN [11, 12, 13, 14]. But all these schemes consider a single group communication scenario. Aparna et al. [15] have discussed a scheme for secure group communication with multiple groups which is based on logical key trees. A combination of key-based and secret share-based approach is used for managing the group keys. Purushothama et al. [16] have proposed a group key management scheme for simultaneous multiple groups with overlapped membership. The scheme is based on key-user tree structure with substantial reduction in storage and rekeying cost. But both of these schemes are proposed for conventional networks. In this paper, we have proposed a scheme for secure group communication for WSN with multiple groups.

3. KEY MANAGEMENT SCHEME

We propose a scheme for group key management with multiple groups. A group consists of n sensing nodes and there are at most m simultaneous groups that need to be established. The nodes are numbered $s_1, s_2 \ldots s_n$ and groups are numbered $G_1, G_2 \ldots G_m$. A logical tree in constructed for each group G_i, for $i = 1,2...m$. The height of the tree for group G_i depends on the number of sensing nodes in G_i and it is $log_2 k$ if there are k ($k \leq n$) nodes in the tree. The tree is maintained by the central node. It constructs a separate key tree for each group. Each sensing node shares a private key with the central node which is used for confidential communication. The group key (GK) is at the root of the tree and is used for confidential communication with the group members. An interior node with two child nodes forms a subgroup and keys associated with the subgroup are called secondary keys. These keys are named either k_{ij} for $j=1,2 \ldots m$ or k_{p-l} depending on whether they have two child nodes or one child node. The key is named k_{ij} if it is the root of the subtree with leftmost child s_i and rightmost child s_j and it is named k_{p-l} if it is the root of the subtree with one child node (left or right). k_p is the leftmost or rightmost child (whichever exists) of this subtree and l is the level number. Secondary keys (keys along the path excluding group key and private key) are used to encrypt new group key. Next we discuss group formation phase followed by computation and distribution of group key.

3.1 Group Formation Phase

The proposed scheme uses Logical Key Hierarchy (LKH) scheme [17] and a binary tree with two keys at each level. The central node is responsible for group formation and rekeying operations. It assigns each sensing node a unique id (*UID*) which is a binary string of length p where $p = \lceil log_2 n \rceil$ where n is the number of sensing nodes. A sensing node s_i, which wishes to join the group G_j sends a join request of the form $JOIN\ (UID_i, G_j)$ to the central node where UID_i is the unique identification number of s_i. A node wishing to join more than one group sends individual join request to each group. A node can send a request to join more than one group in which case it will be a member of more than one key tree.

3.2 Rekeying Strategies and Protocols

We use key based approach for managing group keys and secondary keys. Whenever a node is compromised, it is evicted from the group(s) to which it belongs. Similarly, whenever a new node enters a monitoring area it is added to the group. In either case there is a membership change and hence the group key needs to be changed to prevent a new group member from reading past communications and old group member from reading current and future communications. Whenever there is a membership change, the central node updates the key tree, computes the new group key and distributes it to the existing nodes securely. In the following subsections we discuss the protocols for joining and leaving a group(s) represented by key tree(s).

3.2.1 Joining a key tree

A new node s_i $(1 \leq i \leq n)$ wanting to join a group G_j $(1 \leq j \leq m)$, sends a join request of the form *JOIN* (UID_i, G_j) to the central node (CN). Upon receiving this join request from s_i, the CN checks the node's identity and whether it is allowed to join the group G_j. If so, the CN updates the key tree by creating a node for s_i and ensures backward secrecy by changing the keys along the path from root till its parent and communicating them to appropriate users. The CN computes new group key GK_j' for group G_j and sends it to current members of G_j by encrypting it with old group key GK_j. For the new node, the CN sends the keys along the path by encrypting them with PK_i, private key of s_i. For example, consider an initial key tree with multiple groups shown Fig.1. In the figure s-nodes represent the sensing nodes and nodes labeled *PK* from PK_1 to PK_{11} represent the private keys of s_1 to s_{11}. The k-nodes represent the secondary keys and root nodes labeled GK_1, GK_2, GK_3 represent the group keys.

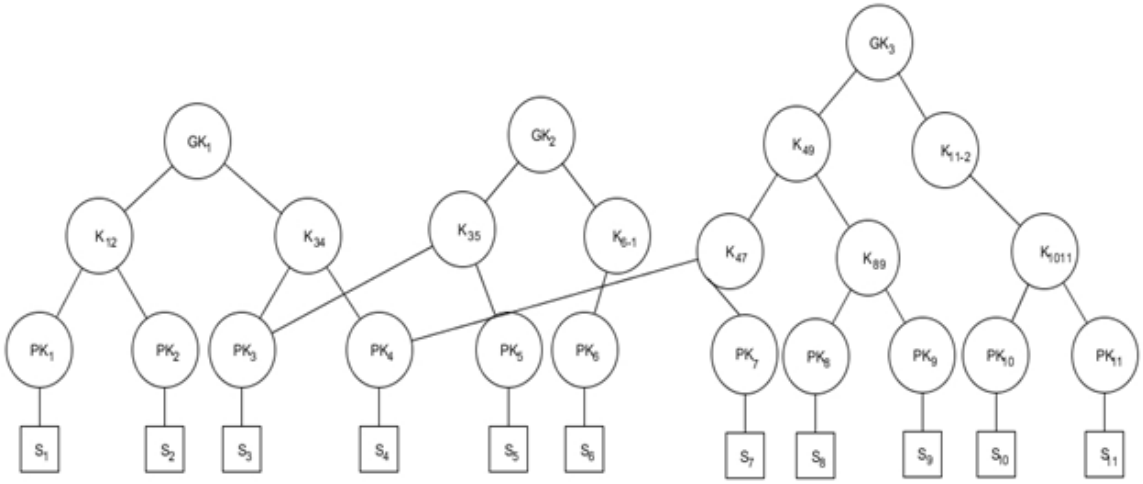

Fig 1 : Initial key tree with 3 groups

There are 3 simultaneous groups G_1, G_2, G_3 with four members, s_1, s_2, s_3, s_4 in G_1, 3 members s_3, s_5, s_6 in G_2 and 6 members s_4, s_7, s_8, s_9, s_{10}, s_{11} in G_3.

Now, suppose a new node s_{12} wants to join group G_3 in Fig.1, it sends a join request *JOIN* (UID_{12}, G_3) to the CN. If the requesting node is allowed to join, the CN updates the key tree as shown in Fig.2. The keys that must be changed are $K_{11\text{-}2}$ and GK_3. The CN changes $K_{11\text{-}2}$ to K_{1012}

and randomly selects a new group key $GK_3^{'}$. The changed keys and the new group key are communicated to appropriate nodes by sending the following rekeying messages:

1. CN $\rightarrow \{s_{12}\} : E_{PK_{12}} (K_{12\text{-}1}, K_{1012}, GK_3^{'})$
2. CN $\rightarrow \{s_{10}, s_{11}\} : E_{K_{1011}} (K_{1012}),\ E_{GK_3} (GK_3^{'})$
3. CN $\rightarrow \{s_4, s_7, s_8, s_9\} : E_{GK_3} (GK_3^{'})$

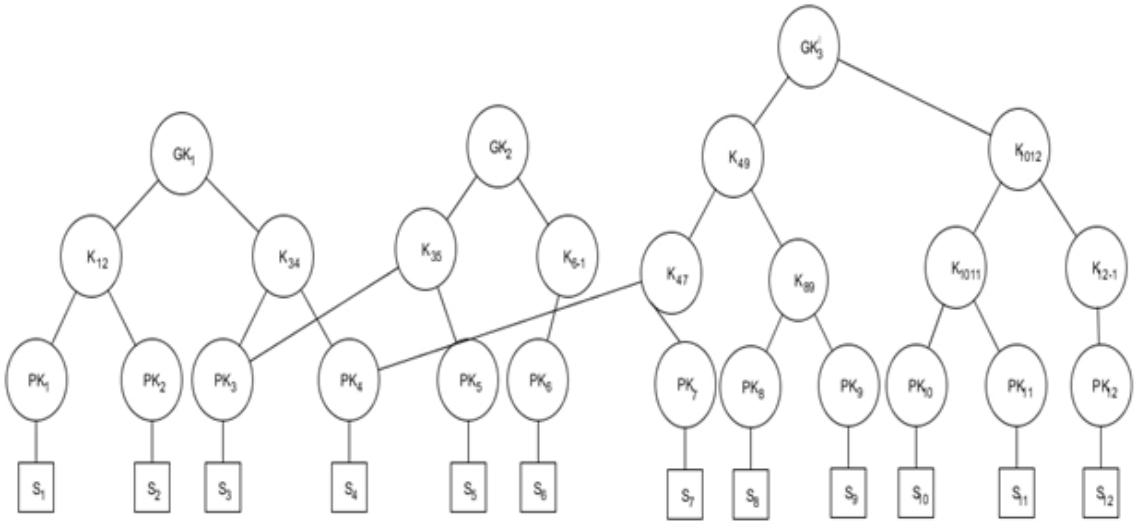

Fig.2 Key Tree after s_{12} joins G_3.

Suppose, now s_{13} wants to join both G_2 and G_3, it sends join requests to both the groups. It sends $JOIN(UID_{13}, G_2)$ and $JOIN(UID_{13}, G_3)$ to the CN. If the requested node is allowed to join G_2 and G_3, the CN updates the key tree as shown in Fig.3. The keys that must be changed are $K_{12\text{-}1}$, K_{1012}, $GK_3^{'}$, $K_{6\text{-}1}$ and GK_2. The new keys are sent to appropriate nodes by generating the following rekeying messages:

1. CN $\rightarrow \{s_{13}\} : E_{PK_{13}} (K_{613}, GK_2^{'}, K_{1213}, K_{1013}, GK_3^{''})$
2. CN $\rightarrow \{s_6\} : E_{GK_2} (GK_2^{'}, K_{613})$
3. CN $\rightarrow \{s_3, s_5\} : E_{GK_2} (GK_2^{'})$
4. CN $\rightarrow \{s_{12}\} : E_{PK_{12}} (K_{1213}),\ E_{K_{1213}} (K_{1013}),\ E_{GK_3^{'}} (GK_3^{''})$

5. CN →{s_{10}, s_{11}} : $E_{K_{1011}}(K_{1013})$, $E_{GK_3'}(GK_3'')$

6. CN →{s_4, s_7, s_8, s_9} : $E_{GK_3'}(GK_3'')$

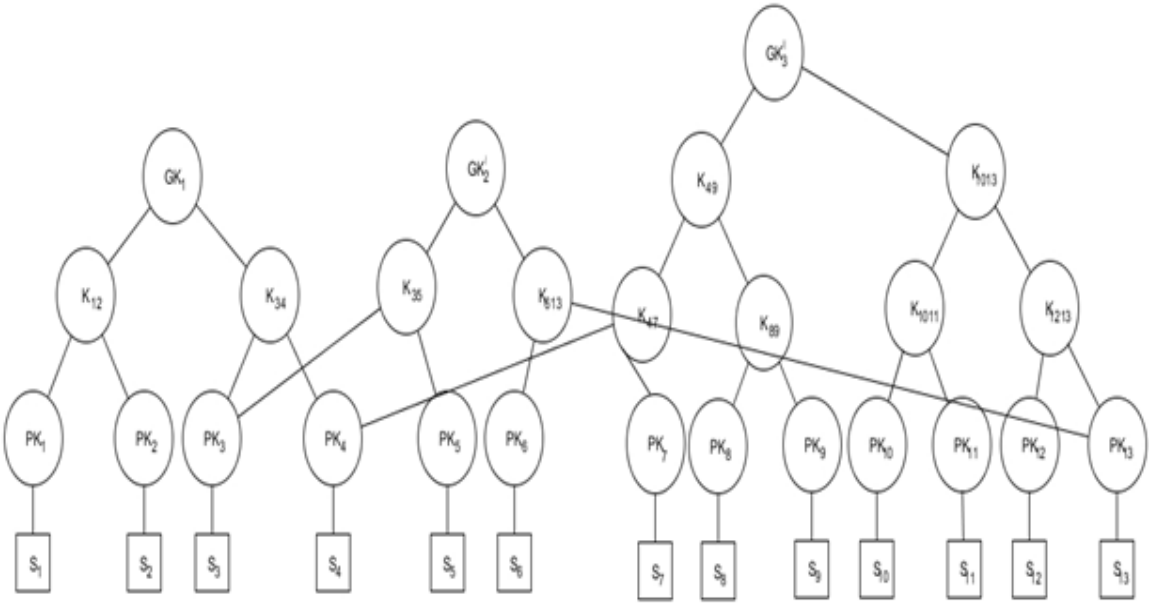

Fig. 3 : Key Tree after s_{13} joins G_2 and G_3.

When a member joins a group *Gj* with *k* members, then at most $log_2 k$ keys have to be changed, $\lceil 2log_2 k \rceil$ encryptions are required and $\lceil log_2 k \rceil$ rekey messages have to be built to communicate the changed keys to the appropriate members of the group. For a member joining *i* number of groups, the number of keys to be changed, number of encryptions required and number of rekey messages to be sent are $\sum_{j=1}^{i} log_2 n_j$, $2\sum_{j=1}^{i} log_2 n_j$, $\sum_{j=1}^{i} log_2 n_j$ respectively where n_j is the number of nodes in group *Gj*.

3.2.2 Leaving a key tree

After a node leaves a group, current group key can no longer be used for future communications and a new group key must be selected and distributed securely to the remaining group members. In addition, all other keys which are known to the leaving node must also be changed to ensure forward secrecy. A leaving node may be a member of a single group or more than one group. Depending on how many groups it belongs to and how many groups it wants to leave, the CN updates the key tree accordingly. Suppose a node s_4 which is a member of G_1 and G_3 wants to leave group G_3, it sends leave request of the form *LEAVE* (UID_4, G_3) to CN. Upon receiving this request CN removes it from G_3 and changes the keys along the path as shown in Fig.4. The keys that must be changed are GK_3'', K_{47}, K_{49}. GK_3'' Changes to GK_3''', K_{47} changes to $K_{7\text{-}1}$ and K_{49} changes to K_{79}. The new keys are conveyed to the existing members of the group by sending the following rekey messages :

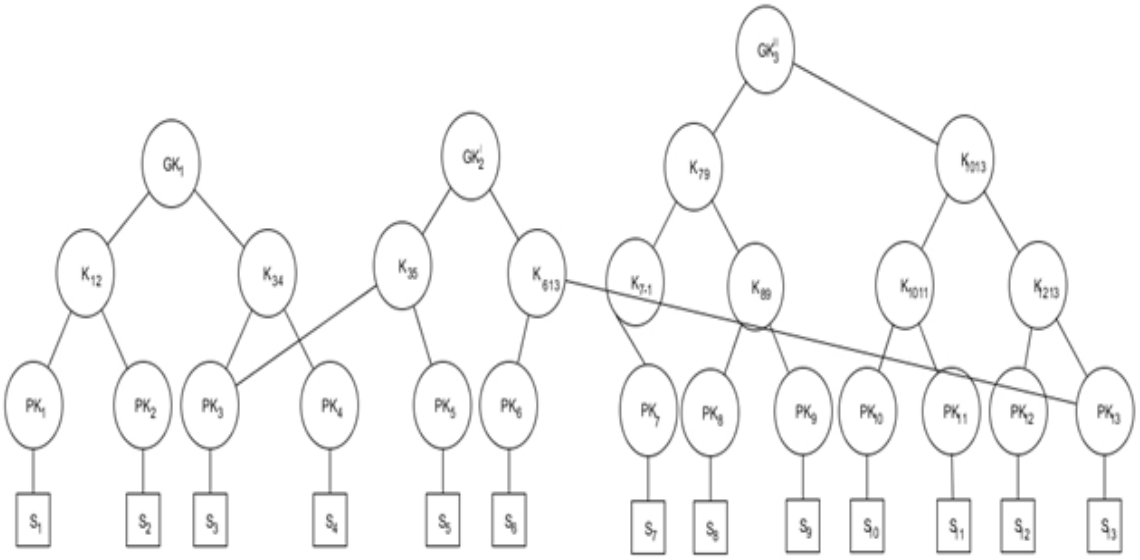

Fig. 4 : Key Tree after s_4 leaves G_3.

1. CN→$\{s_7\}$: $E_{PK_7}(K_{7\text{-}1})$, $E_{K_{7-1}}(K_{79})$, $E_{K_{79}}(GK'''_3)$

2. CN→$\{s_8, s_9\}$: $E_{K_{89}}(K_{79})$, $E_{K_{79}}(GK'''_3)$

3. CN→$\{s_{10}, s_{11}, s_{12}, s_{13}\}$: $E_{K_{1013}}(GK'''_3)$

Now, suppose s_3 wants to leave both G_1 and G_2, it sends leave request to both the groups. It sends *LEAVE* (UID_3, G_1) and *LEAVE* (UID_3, G_2) to CN. The CN removes s_3 from the trees representing G_1 and G_2. The key K_{34}, changes to $K_{4\text{-}1}$, K_{35} changes to $K_{5\text{-}1}$, GK_1 to GK'_1 and GK'_2 to GK''_2. The resulting key tree is shown in Fig. 5 below. The new keys are communicated to corresponding nodes by generating and sending the following rekeying messages :

1. CN →$\{s_4\}$: $E_{PK_4}(K_{4\text{-}1})$, $E_{K_{4-1}}(GK'_1)$

2. CN →$\{s_1, s_2\}$: $E_{K_{12}}(GK'_1)$

3. CN →$\{s_5\}$: $E_{PK_5}(K_{5\text{-}1})$, $E_{K_{5-1}}(GK''_2)$

4. CN →$\{s_6, s_{13}\}$: $E_{K_{613}}(GK''_2)$

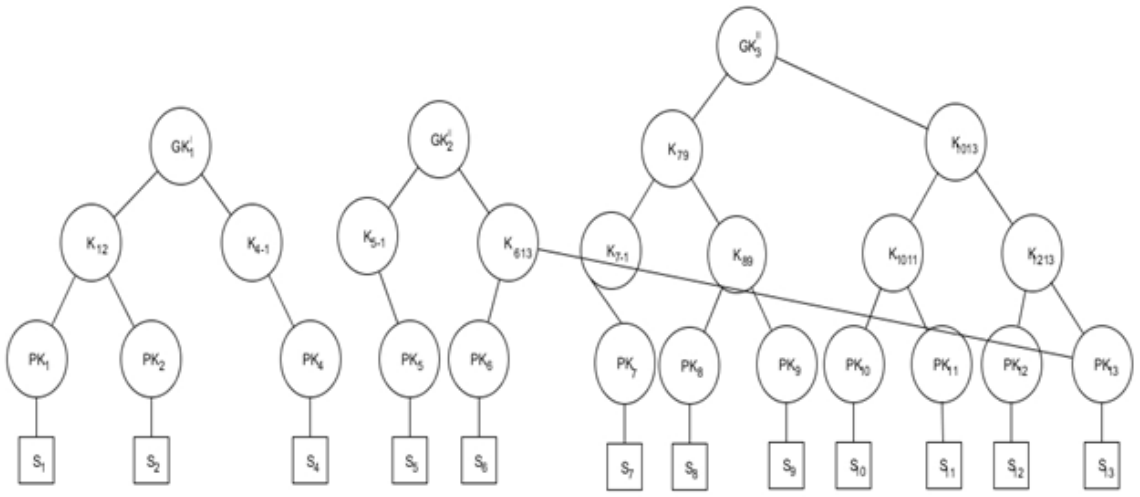

Fig.5 : Key Tree after s_3 leaves G_1 and G_2.

When a member leaves a group *Gj* with *k* members, then at most $log_2 k$ keys have to be changed, $\lceil 2log_2 k \rceil$ encryptions are required and $\lceil log_2 k \rceil$ rekey messages have to be sent to the appropriate members of the group to communicate the changed keys.

3.2.3 Changing the group membership

A node wishing to move from one group to another sends a move request to CN. Move request can be implemented as leave request followed by join request. For example, a node moving from G_i to $G_j$can be interpreted as leaving group G_i and joining group G_j. The CN must ensure forward secrecy for group G_i and backward secrecy for group G_j by changing the keys along the path in G_i and G_j. Consider the key tree in Fig.5. Now, suppose s_4 wants to move from G_1 to G_2, it sends a move request of the form $MOVE\ (UID_4, G_1, G_2)$ to the CN. The CN now removes the node for s_4 from G_1 and inserts it to G_2. The resulting key tree is shown in Fig.6. CN constructs the following rekeying messages and sends to the appropriate nodes:

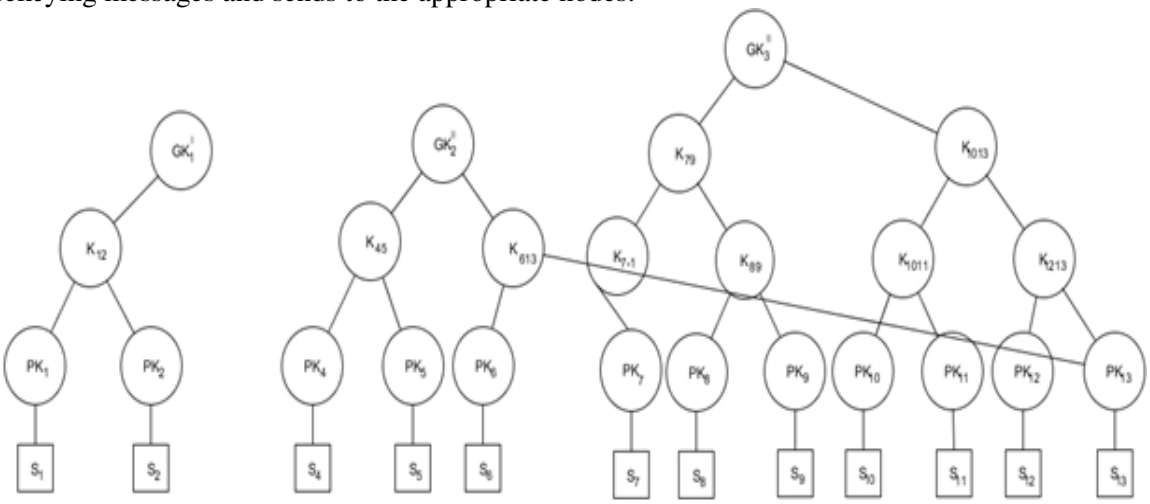

Fig.6 : Key Tree after s4 moves from G1 to G2.

1. CN $\rightarrow \{s_1, s_2\} : E_{K_{12}} (GK_1'')$
2. CN $\rightarrow \{s_4\} : E_{PK_4} (K_{45}, GK_2''')$
3. CN $\rightarrow \{s_5\} : E_{PK_5} (K_{45}), E_{K_{45}} (GK_2''')$
4. CN $\rightarrow \{s_6, s_{13}\} : E_{K_{613}} (GK_2''')$

For a member moving from group G_i to group G_j, the number of keys to be changed, number of encryptions required and number of rekey messages to be sent are $\lceil log_2 n_i + log_2 n_j \rceil$, $\lceil 2(log_2 n_i + log_2 n_j) \rceil$ and $\lceil log_2 n_i + log_2 n_j \rceil$ respectively where n_i and n_j are the number of nodes in groups G_i and G_j respectively.

Each member of the group needs to store $h_j - 1$ secondary keys and one group key where h_j is the height of the tree in group j for $j = 1$ to m. A node needs to store $\sum_{j=1}^{i}(h_j - 1)$ secondary keys and i number of group keys if it is a member of i number of groups.

4. VERIFYING AUTHENTICITY IN GROUP COMMUNICATION

Verifying authenticity of the sender is an important issue in secure group communication which provides protection against masquerade attack. For example, when a sender node s_i sends a message to group G_k, $(1 \leq k \leq m)$, the members of G_k must identify that the sender is s_i and it is not some other node s_j trying to impersonate s_i. In this section, we provide a protocol for authenticated group communication. When a node s_i wants to send a message to group G_k, it first sends a request to CN which includes the node's identity, group identity and a challenge C. The CN in turn sends a authentication key AK_i to s_i encrypted with private key PK_i and hash of AK_i to the members of G_k encrypted with group key GK_i. s_i, now computes hash of AK_i and sends the message M along with $H(AK_i)$ to group members encrypted with group key GK_i. Upon receiving this from s_i, group members decrypt it, compare the received $H(AK_i)$ with the one received from the CN. If both match the group members are sure of the sender and accept the message sent by s_i. Otherwise they discard the message. The use of challenge assures the group members that this is a fresh message and no old message has been replayed. Thus the protocol in Fig.7 provides authenticity as well as confidentiality in group communication. In the protocol we use the symbol || to denote concatenation operation.

1. $s_i \rightarrow$ CN : $[UID_i \,||\, G_i \,||\, C]$
2. CN $\rightarrow s_i$: $E_{PK_i} [AK_i \,||\, f(C)]$
3. CN $\rightarrow G_i$: $E_{GK_i} [H(AK_i) \,||\, f(C) \,||\, UID_i]$
4. $s_i \rightarrow G_i$: $E_{GK_i} [M \,||\, H(AK_i) \,||\, f(C) \,||\, UID_i]$

Fig.7: Protocol for authenticated group communication.

5. CONCLUSION

Secure group communication is an increasingly popular research area having received much attention in recent years. Group oriented applications in WSN demand for the security services to achieve the secure group communication. A common method is to encrypt messages with a group key so that entities outside the group cannot decode them. Therefore, key management is a fundamental building block for secure group communication systems. This paper introduces a key management scheme for WSN with multiple simultaneous groups. We have used a key-based approach for managing the groups and in case of membership change events the communication and computation costs are logarithmic in nature. The paper also provides a protocol for authenticated group communication.

REFERENCES

[1] N.Meghanathan, S. Boumerdassi, N. Chaki, D. Nagamalai, “Recent Trends in Network Security and Applications: Third International Conference”, The Third International Conference on Network Security and Applications, 2010.

[2] R.Shyamala, S. Valli, “Impact of Black hole and Rushing Attack on the Location-Based Routing Protocol for Wireless Sensor Networks”, Advance in Computer & Inform, Technology, pp. 349-359, 2012.

[3] T. Shimeall, J. Spring, “Introduction to Information Security: A Strategic-Based Approach”, Newnes Compute, pp. 382, 2013.

[4] J.Sen, “Security and privacy challenges in cognitive wireless sensor networks”, arXiv preprint arXiv: 1302.2253, 2013.

[5] G. Sharmaa, S. Balaa, A.K.Vermaa, “Security Frameworks for Wireless Sensor Networks-Review”, 2nd International Conference on Communication, Computing & Security, SciVerse Science Direct, 2012.

[6] C. Cheikhrouhou, A. Koubâa, G. Dini, and M. Abid, "RiSeG: a ring based secure group communication protocol for resource-constrained wireless sensor networks", Personal and Ubiquitous Computing, Vol. 15, No. 8, pp. 783-797, 2011.

[7] X. Wanga, P. Lia, Y. Suia, and H. Yanga, "A Hexagon-based Key Pre-distribution Scheme for Wireless Sensor Networks", Journal of Information & Computational Science, Vol. 11 (8), pp. 2479-2491, 2014.

[8] M. Miettinen, N. Asokan, T.D.Nguyen, A-R.Sadeghi, and M. Sobhani, “Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices”, In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 880-891, 2014.

[9] J. Furtak, and J. Chudzikiewicz, "The concept of authentication in WSNs using TPM", Computer Science and Information Systems, Vol. 3, pp. 183–190, 2014.

[10] W. Xi, X-Y. Li, C. Qian, J. Han, S. Tang, J. Zhao, and K. Zhao, "Keep: Fast Secret Key Extraction Protocol for D2D Communication", IEEE, 2014.

[11] Guorui Li; Ying Wang; Jingsha He, "Efficient Group Key Management Scheme in Wireless Sensor Networks", Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI), 2010

[12] Ju-Hyung, Jun-Sik Lee ; Seung-Woo Seo," Energy Efficient Group Key Management Scheme for Wireless Sensor Networks", 2nd International Conference on Communication System Software and Middleware, 2007.

[13] YuanZhang, Yongluo Shen ; SangKeun Lee, "A Cluster-Based Group Key Management Scheme for Wireless Sensor Networks", Web Conference (APWEB), 2010 12th International Asia-Pacific, 2010.

[14] A.S. Poornima, B.B. Amberker, "A Secure Group Key Management Scheme for Sensor Networks", Fifth International Conference on Conference: Information Technology: New Generations, 2008. ITNG 2008

[15] R. Aparna and B. B. Amberker, "Key management scheme for multiple simultaneous secure group communication," in Proceedings of the IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA '09), December 2009.

[16] B R Purushothama , B B Amberker, "Group key management scheme for simultaneous multiple groups with overlapped membership", Third International Conference on Communication Systems and Networks (COMSNETS 2011), 2011.

[17] C.K.Wong, M. Gouda, and S.S. Lam. Secure Group Communication Using key Graphs. IEEE/ACM Transactions on Networking, Volume 8,No.1, pp.16-30, Feb.2000.

AUTHORS

H.S Annapurna is currently working as Associate Professor in the department of Computer Science &Engg., Sri Siddhartha Institute of Technology, Tumkur. She has obtained her Bachelor of Engineering from University of Mysore, Mysore. She has received Masters degree in Software Systems from BITS, Pilani. She is currently pursuing Doctral degree in the area of cryptography and network security from Sri Siddhartha Academy of Higher Education, Tumakuru India.

M.Siddappa received B.E and M.Tech degree in Computer Science & Engineering from University of Mysore, Karnataka, India in 1989 and 1993 respectively. He has completed doctoral degree from Dr.MGR Educational Research Institute Chennai under supervision of Dr.A.S.Manjunatha, CEO, Manvish e-Tech Pvt. Ltd., Bangalore in 2010. He worked as project associate in IISc, Bangalore under Dr. M.P Srinivasan and Dr. V.Rajaraman from 1993 – 1995. He has teaching experience of 26 years and research of 10 years . He published 45 Technical Papers in National, International Conference and Journals. He has citation index of 113 till 2015 and h-index of 3 and i10-index of 1 to his credit. He is a member of IEEE and Life member of ISTE. He is working in the field of data structure and algorithms, Artificial Intelligence, Image processing and Computer networking. He worked as Assistant Professor in Department of Computer Science & Engineering from 1996 to 2003 in Sri Siddhartha Institute of Technology, Tumkur. Presently, he is working as Professor and Head, Department of Computer Science & Engineering from 1999 at Siddhartha Institute of Technology, Tumakuru. He has visited Louisiana university Baton rouge and California university.