International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.05, May 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

GROUP KEY GENERATION USING COLLABORATIVE KEY AGREEMENT PROTOCOLS IN CLIENT-SERVER ARCHITECTURE

[1] Dharmendra, [2] Lalit Tripathi, [3] Swati Gupta

[1]M.Tech Student, Department of Computer Science & Engineering, SCET, Palwal (India)

[2]Assistant Professor, Department of Computer Science & Engineering, SCET, Palwal (India)

[3]Assistant Professor, Department of Computer Science & Engineering, HCST, Mathura (India)

ABSTRACT

We consider several distributed collaborative key agreement protocols for dynamic peer groups. This problem has several important characteristics which make it different from traditional secure group communication. They are distributed nature in which there is no centralized key server, collaborative nature in which the group key is contributory; i.e., each group member will collaboratively contribute its part to the global group key, and dynamic nature in which existing members can leave the group while new members may join. Instead of performing individual rekey operations, i.e., re-computing the group key after every join or leave request, we consider an interval-based approach of rekeying. In particular, we consider two distributed algorithms for updating the group key: (1) the Rebuild algorithm, (2) the Queue-batch algorithm. Performance of these distributed algorithms under different settings, such as different join and leave probabilities, is analyzed. We show that these three distributed algorithms significantly outperform the individual rekey algorithm, and that the Queue-batch algorithm performs the best among the three distributed algorithms. Moreover, the Queue-batch algorithm has the intrinsic property of balancing the computation communication workload such that the dynamic peer group can quickly begin secure group communication. This provides a fundamental understanding about establishing a collaborative group key for a distributed dynamic peer group.

Keywords: Authentication, dynamic peer groups, group key agreement, rekeying, secure group, communication, security.

1. INTRODUCTION

With the emergence of many group-oriented distributed applications such as multi-player games and tele/ videoconferencing, there is a need for security services to provide group-oriented communication privacy and data integrity. To provide this form of group communication privacy, it is important that members of the group can establish a common secret key for encrypting group communication data. For example, consider a group of

people in a peer-to-peer ad hoc network having a closed and confidential business meeting. Since they have not previously agreed upon a common secret key, communication between group members is susceptible to eavesdropping. To solve the problem, we need a secure distributed group key agreement protocol such that the group of people can establish the common group key for secure and private communication. Note that this type of key agreement protocols is both distributed and contributory in nature: each member of the group contributes its part to the overall group key. It is important to point out that the type of distributed group key agreement protocols we study is very different from more traditional centralized group key distribution protocols. Centralized protocols rely on a centralized key server to efficiently distribute the group key. An excellent body of work on centralized key distribution protocols exists in various researches. In those approaches, group members are arranged in a logical key hierarchy known as a key tree. Using the tree topology, it is easy to distribute the group key to members whenever there is any change in the group membership (e.g., a new member joins or an existing member leaves). For distributed key agreement protocols, however, no centralized key server is available. This arrangement is justified in many situations – e.g., in a peer-to-peer or ad hoc network where centralized resources are not readily available. Moreover, an advantage of distributed protocols over the centralized protocols is the increase in system reliability, since the group key is generated in a shared and contributory fashion and there is no single point of failure. [1][2][3].

In this paper, we consider a dynamic communication group in which members are located in a distributed fashion. We extend the Diffie-Hellman key exchange protocol to more than two members in the communication group. The membership of the communication group is dynamic so that members can leave and new members can join the group at any time [4][5]. The contributions of our work are:

- The key agreement protocol is distributed in nature and does not require a centralized key server.
- The key agreement protocol is contributory; each member contributes its part to the overall group key.
- We illustrate that instead of performing individual rekeying operations, one can use an interval-based approach to significantly reduce the computation and communication costs of maintaining the group key.
- We propose three distributed interval-based rekey protocols, and carry out quantitative and simulation-based analysis to illustrate their performance merits [6].

2. PROBLEM DEFINITION

The mainstay of the paper is to collaboratively generate a common key for group communication and to dynamically perform re-keying operation using various algorithms and to share resources using the generated group key in client server architecture. In this project we need to create a system which can provide the members of a group with secure common group key [7]. We implemented Queue Batch algorithm to reduce the rekey complexity. This algorithm uses an interval-based rekey approach so that we can group multiple join/leave requests and process them at the same time. This reduction enables a more efficient way to manage secure group communication. It will help to increase the utilization of resources in existing system. This

algorithm is implemented by using RSA algorithm to generate public and private key in order to create the "secret group key" and then this generated secret group key is distributed among all the group members. Also instead of performing individual rekey operations, i.e., re-computing the group key after every join or leave request, we consider an interval-based approach of rekeying. The key agreement protocol is contributory that is each member contributes its part to the overall group key [8]. Also instead of performing individual rekeying operations, one can use an interval-based approach to significantly reduce the computation and communication costs of maintaining the group key. This system will help to generate a group key with the involvement of all the group members and also its distribution among the group members [9][10].

One major advantage is that this system would not be vulnerable to Man-in-the-Middle attack as attacker would not be able to find the secret key between the group members. This algorithm involves collaborative key agreement in which all nodes become a part of the secure group key. Moreover, rekeying is done after a batch of join or leave operations. The protocol remains efficient even when the occurrences of join/leave events are very frequent. Computational and communication cost is less. Resources used for rekeying is minimized because it is being done for batch of join/leave operations [11].

Next we implemented Rebuild algorithm which is same as Queue Batch algorithm but here rekeying is done at every join/leave requests. The motivation for this algorithm is to minimize the final tree height so that rekeying operations for each group member can be reduced [12].

The main motivation behind comparing these algorithms is that many group-oriented communications require security services. For example: a closed and confidential business meeting in a network. We therefore need a secure group key agreement scheme so that the group can encrypt their communication data with a common secret group key. The scope of this project is in corporate world, people need to communicate with secure medium which will be provided by our system [13][15]. Confidentiality and data integrity is maintained hence can be used by Defense Agencies Intelligence Bureaucrats. These algorithms provide intrinsic property that help dynamic group to quickly begin secure group communication and thus will help saving time.

2. MATHEMATICAL MODEL

2.1. Rebuild Algorithm

The motivation for the Rebuild algorithm is to minimize the final tree height so that the rekeying operations for each group member can be reduced. At the beginning of every rekey interval, we reconstruct the whole key tree with all existing members who remain in the group, together with the newly joining members. The resulting tree would be a complete tree [14]. The pseudo-code of the Rebuild algorithm to be performed by every member is shown below:

Algorithm

1. Obtain all members from T and store them in Ml
2. Remove the L leaving members in Ml from Ml
3. Add the J new members in MJ to Ml
4. Create a new binary tree Tl based on members in Ml and set T = Tl
5. Rekey the key nodes and broadcast the new blinded keys in T.

2.2. Queue Batch Algorithm

Queue batch algorithm, an interval based algorithm is used for re-keying at equal intervals. Queue-batch algorithm performs the best among the interval-based algorithms. The algorithm reduces the latency and the workload created due to re-keying operation that is performed at the beginning of the re-keying intervals.

In Queue batch algorithm, as and when members join, they are stored as in a temporary tree and at the beginning of a re-keying interval this tree is attached to the tree with existing members. It is attached to the highest departed position, so that the height of the tree does not increase much [16][17].

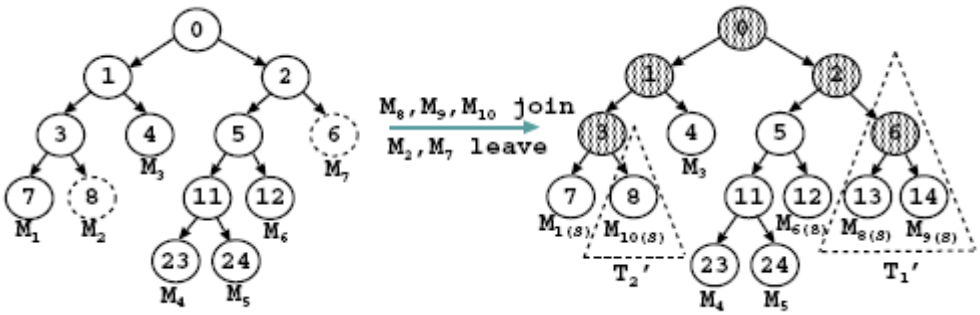


FIGURE 2.1: QUEUE BATCH ALGORITHM

The Queue batch algorithm is illustrated in Figure-1, where members M8, M9, M10 wish to join the communication group, while M2 and M7 wish to leave. Then in the Queue-sub-tree phase, the three new members M8, M9, M10 will form a tree. In the Queue merge phase, the tree is added at the highest departed position, which is at node 6. Now group key is computed for the new group structure and the computed group key is broadcasted to all the members.

Phase 1: Queue-sub tree formation (T0)

1. if (a new member joins) f
2. if (T0 == NULL) /* no new members in T' */
3. create a new tree T0 with the only one new member;
4. else f /* there are new members in T' */
5. find the insertion node;
6. add the new member to T0;

7. Select the rightmost member under the sub-tree rooted at the sibling of the joining node to be the sponsor;
8. if (sponsor) /* sponsor's responsibility */
9. Re-key the key nodes and broadcast the new blinded keys to the communication group;

The node ID of root node is set 0. Each non-leaf node v consists of 2 child-nodes, with ID 2v+1 2v+2.Based on this protocol, the secret key of v can be generated by the secret key of one child node blinded node of another child-node [18].

Phase 2: Queue-merge (T, T0,M*l* ;*L*)

1. if (L == 0) f /* there are no leave */
2. add T0 to either (a) the shallowest node (which need not be the leaf node) of T such that the merger would not increase the resulting tree height, or

(b) the root node of T if the merge to any locations would increase the resulting tree height;

3. else /* there are leaves */
4. add T0 to the highest leave position of the key tree T;
5. elect members to be sponsors if they are (a) the rightmost member of the sub-tree rooted at the sibling nodes of the departed leaf nodes in T, or (b) the rightmost member of T0;
6. if (sponsor) /* sponsor's responsibility */
7. re-key the key nodes and broadcast the new blinded keys to the communication group;

The Queue-batch algorithm has the intrinsic property of balancing the computation/communication workload such that the dynamic peer group can quickly begin secure group communication.

3. PERFORMANCE EVALUATION

In this section, we present the mathematical analysis of the two proposed algorithms. We consider two performance measures, namely:

1. *Number of renewed nodes:* a node is said to be *renewed* if it is a non-leaf node and its associated keys are renewed. This metric provides a measure of the communication cost since new blinded keys of the renewed nodes have to be broadcast to the whole group [19].

2. *Number of exponentiation operations:* this metric provides a measure of the computation load for all members in the communication group. For simplicity, we assume the following in the analysis:

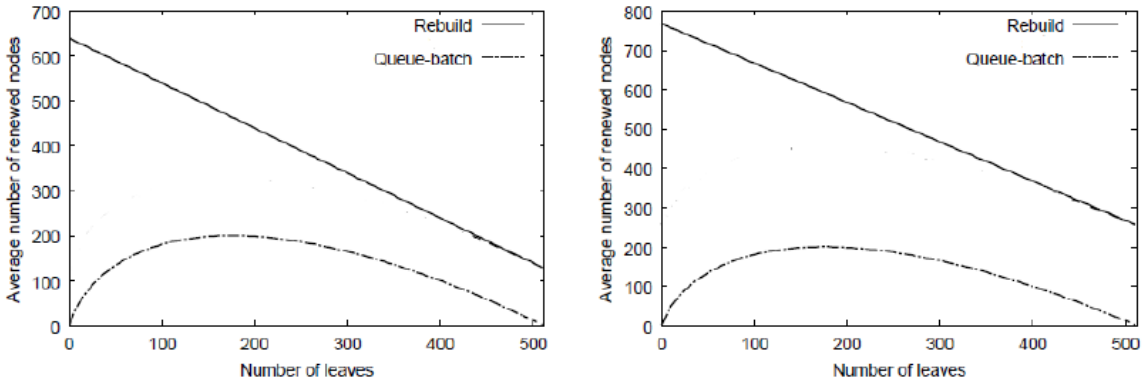
- The existing key tree T is a completely balanced tree before the interval-based rekeying event.
- Each member has a homogeneous leave probability.
- The number of blinded key computations simply equals that of renewed nodes, provided that the blinded key of each renewed node is broadcast only once [20][21][22].

4. RESULTS

The above experiments show that Queue-batch offers the best performance in terms of computation and communication costs among the two interval-based algorithms. The superior performance of Queue-batch is

more obvious when the occurrences of join and leave events are highly frequent. Queue-Batch algorithm require more time as compared to Rebuild algorithm in the generation of group key. This is due to the fact that in queue batch algorithm, group is generated after a predefined batch of time irrespective of joining and leaving operation of group members whereas in Rebuild algorithm, group is generated after every join and leave operation of group members.

Load on system resources is more while implementing Rebuild algorithm as compared to Queue-Batch algorithm. This is due to the fact that in Rebuild algorithm, group key is generated after every join and leave operation of group members so more work has to be done.



(a) $J = 128$ (b) $J = 256$

FIGURE 4.1: AVERAGE NUMBER OF RENEWED NODES AT DIFFERENT NUMBERS OF JOINS (J) WHEN THE ORIGINAL TREE IS COMPLETELY BALANCED

5. CONCLUSION

We have implemented two algorithms i.e. 'Queue-batch algorithm' and 'Rebuild algorithm' for generating common group key using collaborative key agreement protocols for dynamic groups in client server architecture. We found that interval-based rekey approach is the better than rebuild algorithm which reduces rekeying complexity so that multiple join/leave requests can be grouped and processed at the same time. In particular, we show that the Queue-batch algorithm is better than rebuild algorithm and can significantly reduce both computational and communication costs. This reduction enables a more efficient way to manage secure group communication. We conclude that our group key server using any of the two rekeying strategies is scalable to very large groups with frequent joins and leaves. In particular, the average server processing time per join/leave increases linearly with the logarithm of group size.

REFERENCES

[1] BURMESTER, M., AND DESMEDT, Y. A secure and scalable group key exchange system. IEEE Trans (2010).

[2] CANNY, J. Collaborative filtering with privacy. In *Security and Privacy, 2006. Proceedings. 2002 IEEE Symposium on* (2002), IEEE, pp. 45–57.

[3] HOU, R. Y.-K. *Improving reliability and performance of redundant disk arrays by improving rebuild time and response time.* PhD thesis, 1994.

[4] KIM, Y., PERRIG, A., AND TSUDIK, G. Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC) 7*, 1 (2004), 60–96.

[5] LEE, P. P., LUI, J. C., AND YAU, D. K. Distributed collaborative key agreement and authentication protocols for dynamic peer groups. *Networking, IEEE/ACM Transactions on 14*, 2 (2006), 263–276.

[6] LUCANTONI, D. M. New results on the single server queue with a batch markovian arrival process. *Communications in Statistics. Stochastic Models 7*, 1 (1991), 1–46.

[7] RAJARAM, M., AND THILAGAVATHY, D. An interval based contributory key agreement. In *Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on* (2010), IEEE, pp. 16.

[8] RATHGEB, C., AND UHL, A. An iris-based interval-mapping scheme for biometric key generation. In *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on* (2009), IEEE, pp. 511–516.

[9] WONG, C. K., GOUDA, M., AND LAM, S. S. Secure group communications using key graphs. *Networking, IEEE/ACM Transactions on 8*, 1 (2000), 16–30.

[10] YANG, Y. R., LI, X. S., ZHANG, X. B., AND LAM, S. S. Reliable group rekeying: a performance analysis. In *ACM SIGCOMM Computer Communication Review* (2001), vol. 31, ACM, pp. 27–38. 44189-198. 2003.

[11] Robust Location Privacy Scheme for VANET," IEEE J. Sel. Areas Commun., vol. 25, no. 8, pp. 1569-1589, Oct. 2007.

[12] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," in Advances in Cryptology– EUROCRYPT'94, LNCS, vol. 950, pp. 275-286, 1995.

[13] M. Waldvogel, G. Caronni, D. Sun, N. Weiler and B. Plattner, "The VersaKey Framework: Versatile Group Key Management," IEEE J. Sel. Areas Commun., vol. 17, no. 9, pp. 1614-1631, Sept. 1999.

[14] M. Steiner, G. Tsudik and M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 8, pp. 769-780, Aug. 2009.

[15] A. Sherman and D. McGrew, "Key Establishment in Large Dynamic Groups Using One-way Function Trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.

[16] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stanton, and G. Tsudik, "Secure Group Communication Using Robust Contributory Key 0Agreement," IEEE Trans. Parallel Distrib. Syst., vol. 15, no. 5, pp. 468- 480, May 2004.

[17] Y. Kim, A. Perrig and G. Tsudik, "Tree-Based Group Key Agreement," ACM Trans. Inf. Syst. Security, vol. 7, no. 1, pp. 60-96, Feb. 2004.

[18] Y. Sun, W. Trappe and K.J.R. Liu, "A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks," IEEE/ACM Trans. Netw., vol. 12, no. 4, pp. 653-666, Aug. 2004.

[19] W. Trappe, Y. Wang and K.J.R. Liu, "Resource-Aware Conference Key Establishment for Heterogeneous Networks", IEEE/ACM Trans. Netw., vol 13, no 1, pp.134-146, Feb. 2008.

[20] P. P. C. Lee, J. C. S. Lui and D. K. Y. Yau, "Distributed Collaborative Key Agreement and Authentication Protocols for Dynamic Peer Groups," IEEE/ACM Trans. Netw., vol. 14, no. 2, pp. 263-276, April 2006.

[21] Y. Mao, Y. Sun, M. Wu and K. J. R. Liu, "JET: Dynamic Join-Exit Tree Amortization and Scheduling for Contributory Key Management," IEEE/ACM Trans. Netw., vol 14, no 5, pp.1128-1140, Oct. 2006.

[22] W. Yu, Y. Sun and K. J. R. Liu, "Optimizing the Rekeying Cost for Contributory Group Key Agreement Schemes," IEEE Trans Dependable and Secure Computing, vol. 4, no. 3, pp. 228 - 242, July-Sep. 2007.