Journal of Computer Science 8 (6): 951-956, 2012
ISSN 1549-3636

Efficient Star Topology based Multicast Key Management Algorithm

Saravanan, K. and T. Purusothaman
Department of CSE, Government College of Technology, Coimbatore, India

Abstract: Problem statement: Secure group communication is very important for many applications such as internet pay sites. It provides efficient delivery of identical data to only the customers in the group. In large and dynamic multicast groups, the group keys of members have to be changed frequently whenever the member leaves or joins. A common method is to apply a symmetric key that is used to encrypt the transmitted data. The rekeying cost scales linearly with the number of members in the group and cost of the rekeying process is the main issue. The tree-based architecture is commonly used to reduce the rekeying cost in terms of storage, transmission and computation. But it usually gives extra overhead to balance the tree which is in order to achieve logarithmic rekeying cost. **Approach:** The main aim was to use star topology based architecture to avoid the balancing and eliminate the rekeying processes and more over it was more secured by exchanging the secret key between only server and each group member. The features of proposed algorithm were that the private key was computed by individual member. **Results:** The burden of server was reduced and also there was no rekeying when a member leaves the group. The secret value of leaving member was not added in the encryption and so the private value could not be obtained after decryption. **Conclusion:** Proposed algorithm is simple and no rekeying when a member leaves and also reduces the computation and communication complexity.

Key words: Group communication, rekeying, secure multicast, key management

INTRODUCTION

In the modern technology world, network attacks have become more sophisticated and harder to identify the attack. When many applications like scalable chat services and streaming video, are expected to run over the Internet, the security is necessary in computing and communication became a necessity. The internet today provides less security for privacy and authentication of multicast packets. The number of applications using multicast increases day by day and so it need secure multicast services.

Multicast is an internetwork service which provides efficient delivery of data from a source to multiple receivers and also improve the bandwidth efficiency of the network. A common group key is necessary for individual members in the group for secure multicast communication. In general the group key management (Peyravian *et al*., 1999; Rafaeli and Hutchison, 2003; Zhu and Jajodia, 2003; Kim *et al*., 2005; Devi and Padmavathi, 2010; Sahar *et al*., 2010; Abdul-Rahman *et al*., 2011) can be divided into three categories (a) centralized key management (b) distributed key management (c) decentralized key management.

In all approaches (Harney and Muckenhirm, 1997; Waldvogel *et al*., 1999; Wong *et al*., 2000; Sherman and McGrew, 2003; Sahar *et al*., 2010) whenever a member joins or leaves the group or the members are static in nature, the group key has to be changed to achieve forward secrecy which assures that the newly joined members cannot decrypt the multicast data sent earlier before joining the group and assures that the former members cannot decrypt the communication after leaving the group.

In most of the key management protocol (Peyravian *et al*., 1999; Tu *et al*., 1999; Wong *et al*., 2000; Selcuk and Sidhu, 2002; Zhu and Jajodia, 2003; Kreishan, 2011; Mansouri *et al*., 2011) tree topology is used. Tree balancing is another issue when a member joins or leaves. The main drawback of tree topology is that number of overhead and cost for rekeying proportionately increase if the number of member increases. A huge database is necessary for storage and complexity also increases. Scalability is an issue in connection with the dynamic multicast members.

MATERIALS AND METHODS

The drawback of tree based architecture was overcome in SBMK (Lin *et al*., 2010) which uses star-based architecture in which the server computes a secret key and unicast to every user separately. But the drawbacks of these kinds of protocols are as follows:

Corresponding Author: Saravanan, K., Department of CSE, Government College of Technology, Coimbatore, India

- It increases the load on the server
- Computational and communication complexities are increased
- If private key is computed and sent by a server to all the members then the private component of members may not be used for authentication

Our star topology based proposed algorithm has overcome the above problem:

- The total load on the server reduces because the private key is computed and sent by each user to server. So it reduces the load on the server
- The private component of members may be used for authentication
- It also reduces the computation complexity of server
- It gives a better rekeying performance than that of the key tree and there is no need to balance the tree
- Moreover our proposed scheme takes care of the important security requirements for secure group communication such as group secrecy, forward secrecy and backward secrecy

In this study, we propose an efficient star based key management algorithm for internet pay sites, which is relatively simple to implement.

The rest of the study describes the proposed scheme, derives the result with suitable illustration of proposed algorithm, discuss and compare the proposed algorithm with the existing algorithms and finally concludes the study and future work.

Proposed scheme: In the proposed star topology based algorithm, the individual member joining the group is allowed to choose prime numbers and compute their private key and the secret value of N computed is sent to the server by a secure unicast message. Thus the burden of server is reduced and also rekeying is totally reduced and also scalable for a large multicast group.

Key assignment phase: The steps of key assignment are as listed below:

Step 1: First the server authenticates the user who want to join the multicast group and also announce public value as e. It is common for the server as well as users.

Step 2: The individual user Mi randomly select two prime numbers m and n and calculate the product Xi = mi x ni and ϕ (Xi) = (mi-1) x (ni-1).

Step 3: The private of key of individual member will be calculated (Rivest *et al*., 1978; Menezes *et al*., 1997; Sharma *et al*., 2011) by the user using the extended Euclidean algorithm to calculate a unique integer d_i such that Eq. 1:

$$e \times d_i \equiv 1 \bmod (\phi (X_i)) \quad (1)$$

where, $d_i > \phi (X_i)$

Step 4: The authenticated individual members send their X value to the server.

Step 5: The server verifies and accepts the X_i value only if it is unique value from other members and hold the value of X_i as secret.

Message encryption:

Step 1: When the server wants to send a secret message P to selected users in the multicast group M_1, M_2, M_4, then the server uses e as well as the secrets of Members X_1, X_2 and X_4

The encryption of the secret message is computed using the general formulae Eq. 2 (Lin *et al*., 2010):

$$C = (P)^e \bmod \prod Xi \quad (2)$$

where, Xi includes the X value of members to whom the secret plain text has to be sent.

Step2: The server computes cipher text and sends a broadcast message to all the members of the group.

Message decryption:

Step 1: The individual member Mi receiving the cipher text C can use its private key di and his/her public parameter Ni, to decrypt the plain text and obtain the secret and confidential message P using (Lin *et al*., 2010) the following formulae Eq. 3:

$$Mi \rightarrow P = (C \bmod Xi)^{di} \bmod X_i \quad (3)$$

Member joining: When a new member Mn+1 want to join the group, the key server repeats the procedures similar to key assignment.

Step 1: First the server authenticates the user who want to join the multicast group and also announce public value as e .It is common for the server as well as users.

Step 2: The individual user M_{i+1} select two prime numbers m and n and calculate the product $X_{i+1} = m_{i+1} \times n_{i+1}$ and $\phi(X_{i+1}) = (m_{i+1}-1) \times (n_{i+1}-1)$.

Step 3: The private of key (Rivest *et al*., 1978; Menezes *et al*., 1997; Sharma *et al*., 2011) of individual member will be calculated by the user using the extended Euclidean algorithm to calculate a unique integer d0 such that:

$$e \times d_{i+1} = 1 \bmod (\phi(X_{i+1}))$$

Step 4: The newly joined member send their X_{i+1} value to the server

Step 5: The server verify the X_{i+1} and accept the X_{i+1} only if it is unique value from other members and hold the value of X as secret

Members leaving: When a member Xi leaves the group, the key server just deletes the secret information X_i. Therefore, in the cipher text computation (Rivest *et al*., 1978; Menezes *et al*., 1997; Sharma *et al*., 2011) in Formula (2) removes the modulus operations with respect to X_i ($m_i \times n_i$). Member Mi, cannot decrypt the secrete message because Xi is not added in cipher text calculation. Hence both forward and backward secrecy is maintained. The pair of prime numbers of a leaving member cannot be reassigned to new user joining the group. So there is no need for rekeying even if the members of multicast group change.

RESULTS

Illustration of the proposed algorithm with suitable examples and the result obtained is discussed in this section.

Key assignment phase: The steps of key assignment are as listed below:

Step 1: First the server authenticate the user who want to join the multicast group and also announce public value as e = 103. It is common for the server as well as users.

Step 2: The individual user Mi selects two prime numbers mi and ni randomly and also compute their Xi = mi x ni and $\Phi(X_i) = (m_i-1) \times (n_i-1)$:

M_1 selects m_1=163 and n1=227
computes X_1=37001 and $\phi(X_1)$ =36612
M_2 selects m_2=181 and n2=233
computes X_2=42173 and $\phi(X_2)$ =41760
M_3 selects m_3=163 and n3=199
computes X_3=32437 and $\phi(X_3)$ =32076
M_4 selects m_4=137 and n4=173
computes X_4=23701 and $\phi(X_4)$ =23392
M_5 selects m_5=223 and n5=211
computes X_5=47053 and $\phi(X_5)$ =46620
M_6 selects m6=251 and n6=191
computes X_1=47941 and $\phi(X_6)$ =47500

Step 3: The private of key of individual member is calculated by each user using the extended Euclidean algorithm used in RSA algorithm:

$103 \times d_1 \equiv 1 \bmod (\phi(X_1) \equiv 1 \bmod 36612$ and d1 =16351
$103 \times d_2 \equiv 1 \bmod (\phi(X_2) \equiv 1 \bmod 41760$ and d2 =6487
$103 \times d_3 \equiv 1 \bmod (\phi(X_3) \equiv 1 \bmod 32076$ and d3 =28339
$103 \times d_4 \equiv 1 \bmod (\phi(X_4) \equiv 1 \bmod 23392$ and d4 =6359
$103 \times d_5 \equiv 1 \bmod (\phi(X_5) \equiv 1 \bmod 46620$ and d5 =16747
$103 \times d_6 \equiv 1 \bmod (\phi(X_6) \equiv 1 \bmod 47500$ and d6 =2767

Step 4: The individual members communicate their X value to the server:

X1=37001, X2=42173, X3=32437 X4=23701 X5=47053, X6=47941

Step 5: Server verifies the secret value Xi of individual users and accepts the X only if it is unique value from other members and hold the value of Xi as secret.

Multicast communication:

Step 1: Assume there are 6 members X1, X2, X3, X4, X5, X6 in the multicast group. When the server wants to send a secret message P= 5 to all members in the multicast group, then the server uses its public value e0 as well as the secrets of Members X1, X2, X3, X4, X5,X6 in the encryption formulae

The encryption of the secret message is computed using the general formulae:

$$C = 5^{103} \bmod (X_1 \times X_2 \times X_3 \times X_4 \times X_5 \times X_6)$$
$$= 5^{103} \bmod (37001 \times 42173 \times 32437 \times 23701 \times 47053 \times 47941)$$
$$= 17496300349800947092273130 40$$

Step2: The server computes cipher text C and sends a broadcast message to all the members of the group

Step3: All six members in the multicast group can decrypt the secret message P from the cipher text received. The secret message is decoded as follows:

$M1 \rightarrow P = (CmodX_1)^{d1} \bmod X_1$
$= (17496300349800947092273130 40 \bmod 37001)^{16351} \bmod 37001 = 5$
$M2 \rightarrow P = (CmodX_2)^{d2} \bmod X_2$
$= (17496300349800947092273130 40 \bmod 42173)^{6487} \bmod 42173 = 5$
$M3 \rightarrow P = (CmodX_3)^{d3} \bmod X_3 = (17496300349800947092273130 40 \bmod 32437)^{28339} \bmod 32437 = 5$
$M4 \rightarrow P = (CmodX_4)^{d4} \bmod X_4 = (17496300349800947092273130 40 \bmod 37001)^{16351} \bmod 37001 = 5$
$M5 \rightarrow P = (CmodX_5)^{d5} \bmod X_5 = (17496300349800947092273130 40 \bmod 47053)^{16747} \bmod 47053 = 5$
$M6 \rightarrow P = (CmodX_6)^{d6} \bmod X_6 = (17496300349800947092273130 40 \bmod 47941)^{2767} \bmod 47941 = 5$

Members joining: When two members M7 and M8 want to join the multicast group, the key server repeats the formalities similar to key assignment.

Step 1: First the server authenticates the users M7 and M8 who want to join the multicast group and also inform the public value as e = 103.It is common for the server as well as users.

Step 2: Member M7 selects two prime numbers (m7=149, n7=191) and M8 selects two prime numbers (m8=199, n8 = 179) randomly and also compute their.

$X_i = m_i \times n_i$ and $\phi(X_i) = (m_i-1) \times (n_i-1)$
M_7 selects m_1 =149 and n1=191 computes $X_7 = 28459$ and $\phi(X_7) = 28120$
M_8 selects m_2 =199 and n2 = 179 computes $X_8 = 35621$ and $\phi(X_8) = 35244$

Step 3: The private of key of individual member M7 and M8 is calculated by each user using the extended Euclidean algorithm used in RSA algorithm:

$103 \times d7 \equiv 1 \bmod (\phi(X7) \equiv 1 \bmod 28120$ and d7 =27847
$103 \times d8 \equiv 1 \bmod (\phi(X8) \equiv 1 \bmod 35244$ and d8 =13687

Step 4: Member M7 and M8 inform their X values to the server

Step 5: Now the server will add M7 and M8 in the database and also when it is sending a new secret value, it will add M7 and M8 in the cipher text formulae

Step 6: Suppose if the server wants to send a new secret message P = 342 to all the members X_1, X_2, X_3, X_4, X_5, X_6, X_7 and X_8, it will compute new cipher text using the formulae in Eq. 2:

$C = 342^{103} \bmod (X_1 \times X_2 \times X_3 \times X_4 \times X_5 \times X_6 \times X_7 \times X_8)$
$= 342^{103} \bmod (37001 \times 42173 \times 32437 \times 23701 \times 47053 \times 47941 \times 28459 \times 35621)$
$= 353976104622099213964209472644959996 3$

Step7: The existing group members X_1, X_2, X_3, X_4, X_5, X_6 use their exiting private key to decrypt the cipher text to get the new secret message and the newly added two members X_7 and X_8 use their private key to decrypt the secret message as given below:

$M1 \rightarrow P = (CmodX1)^{d1} \bmod X_1$
$= (35397610462209921396420947264495999 63 \bmod 37001)^{16351} \bmod 37001 = 342$
$M2 \rightarrow P = (CmodX2)^{d2} \bmod X_2$
$= (35397610462209921396420947264495999 63 \bmod 42173)^{6487} \bmod 42173 = 342$
$M3 \rightarrow P = (CmodX3)^{d3} \bmod X_3$
$= (35397610462209921396420947264495999 63 \bmod 32437)^{28339} \bmod 32437 = 342$
$M4 \rightarrow P = (CmodX4)^{d4} \bmod X_4$
$= (35397610462209921396420947264495999 63 \bmod 23701)^{16351} \bmod 23701 = 342$
$M5 \rightarrow P = (CmodX5)^{d5} \bmod X_5$
$= (35397610462209921396420947264495999 63 \bmod 47053)^{16747} \bmod 47053 = 342$
$M6 \rightarrow P = (CmodX6)^{d6} \bmod X_6$
$= (35397610462209921396420947264495999 63 \bmod 47941)^{2767} \bmod 47941 = 342$
$M7 \rightarrow P = (CmodX7)^{d7} \bmod X_7$
$= (35397610462209921396420947264495999 63 \bmod 28459)^{27847} \bmod 28459 = 342$
$M8 \rightarrow P = (CmodX8)^{d8} \bmod X_8$
$= (35397610462209921396420947264495999 63 \bmod 35621)^{13687} \bmod 35621 = 342$

Members leaving: When two members M_5 and M_6 leaves the group, the key server just deletes the secret information of X_5 and X_6 correspond to M_5 and M_6.

Step1: In the cipher text computation in Formula (2) removes the modulus operations with respect to

X_5= 47053 and X_5= 47941 if the server wants to send a new secret message P=25 to members X_1, X_2, X_3, X_4, X_7 and X_8:

$C = 342^{103} \bmod (X_1 \times X_2 \times X_3 \times X_4 \times X_7 \times X_8)$ =55083698761217809158180342 95

Step 2: Using the private keys, the members M1, M2, M3, M4, M7, M8 can decrypt the secret message P =25

Members M5 and M6 will get different secret message as 25057 and 11127 respectively, which is different from the actual one.

The members M_5 and M_6 cannot decrypt the secret message because X_5 and X_6 is not added in the cipher text computation. Hence both forward and backward secrecy is maintained. When a member leaves the group, the server would not allow a new member to select the same pair of prime number. So there is no need for rekeying even if the members of multicast group change.

DISCUSSION

Complexity analysis: The secret Xi generated by the individual authenticated user is hold by server as a confidential one. It is known only to the corresponding users and server.

Difficulty for unauthorised member try to deduce the private key and secret value: The security of our proposed algorithm depends on the secret value and private key of individual users. It is not possible for the unauthorized person to derive the private key di from the public parameter e. It is extremely difficult for the adversary to derive the private key from the public parameter e alone. Moreover the secret value X is unique and the number of digit may also vary for every user.

Preventing the unauthorized access: If a member is not authenticated by the server, the server will not add the secret value of X in the cipher text calculation and unauthorized member will get different value when he tries to decrypt the encrypted message. So it is more secured.

Performance analysis based on complexity comparison of various key management schemes: Table 1-4 provides the comparative analysis of the various protocols. It shows that every protocol achieves unique results when applying different techniques. Some protocols achieve exceptionally better results than others do. By comparing the table, we can clearly understand that the bottleneck of server is avoided by reducing total no of keys managed by server in our proposed algorithm. It is also smaller when compared with LKH, OFT and SBMK algorithm (Kim *et al*., 2005; Lin *et al*., 2010; Abdul-Rahman *et al*., 2011).

Only one multicast message will be send to the group when a member joins and no message is send when a member leaves the group. So there is no need for rekeying when a member leaves also the rekeying overhead is less compared with LKH and OFT (Kim *et al*., 2005; Abdul-Rahman *et al*., 2011).

The proposed algorithm achieves better results for storage, communication, computation and processing on both server and user. The computation cost of server is greatly reduced by allowing the users to calculate their private key and secret values compared with other techniques.

The cost of encryption when a member joins the group is 1 and the cost of encryption when a member leaves the group.

From the tables we can easily understand that proposed protocol is more suitable for a dynamic users and storage cost of server is reduced (Lin *et al*., 2010) and distributed to the users.

Table 1: Communication cost

Protocols	Join multicast	Leave unicast
LKN 2 LOG -1	Log n	
One way function tree	Log n+1	Log n+1
SBMK	1	0
Proposed protocol	1	0

Table 2: Computation cost

Protocols	Join	Leave
LKH	2 log n -1	2 log n
One way function tree	log n +1	log n +1
SBMK	1	0
Proposed protocol	1	0

Table 3: No of rekey messages

	No of rekey messages needed	
Protocols	Join	Leave
LKH	d+1	2d
One way function tree	d+1	d+1
SBMK	1	0
Proposed Protocol	1	0

Table 4: Key storage during join and leave operations

Protocols	Server	User
LKH	2n	log n +1
One way function tree	2n	2log n +
SBMK	2n + 2	3
Proposed protocol	n +2	3

CONCLUSION

In this study, an efficient Star Topology based Multicast Key Management algorithm is proposed and implemented which produces better results than the existing protocols in terms of less computational, communication and storage costs. The proposed star based architecture reduces the rekeying overhead. The private key of the users are computed by the individual and so it can be used for authentication also.

The computation complexity of the server is totally reduced in the new protocol. It is also scalable and easy to implement when the number of users are very high and dynamic in nature .As future scope of work, it may be extended for bulk member join and leaves.

REFERENCES

Abdul-Rahman, H., A.M. Alashwal and Z.H. Jamaludin, 2011. Implementation and methods of project learning in quantity surveying firms: Barriers, enablers and success factors. Am. J. Econ. Bus. Admin., 3: 430-438. DOI: 10.3844/ajebasp.2011.430.438

Devi D.S. and G. Padmavathi, 2010. Secure multicast key distribution for mobile adhoc networks. Int. J. Comput. Sci. Inform. Security, 7: 218-222.

Harney, H. and C. Muckenhirm, 1997. Group Key Management Protocol (GKMP) architecture. SPARTA, Inc.

Kim, H., S.M. Hong, H. Yoon and J.W. Cho, 2005. Secure group communication with multiplicative one-way functions. Proceedings of the IEEE International Conference on Information Technology: Coding and Computing, Apr. 4-6, IEEE Xplore Press, Washington, DC, USA., pp: 685-690. DOI: 10.1109/ITCC.2005.252

Kreishan, F.M., 2011. Economic growth and unemployment: An empirical analysis. J. Soc. Sci., 7: 228-231. DOI: 10.3844/jssp.2011.228.231

Lin, I.C., S.S. Tang and C.M. Wang, 2010. Multicast key management without rekeying processes. Comput. J., 53: 940-950. DOI: 10.1093/comjnl/bxp060

Mansouri, M., A. Ganguly and A. Mostashari, 2011. Evaluating agility in extended enterprise systems: A transportation network case. Am. J. Eng. Applied Sci., 4: 142-152. DOI: 10.3844/ajeassp.2011.142.152

Menezes, A.J., P.C.V. Oorschot and S.A. Vanstone, 1997. Handbook of Applied Cryptography. 1st Edn., CRC Press, Boca Raton, Fla., ISBN: 0849385237, pp: 780.

Peyravian, M., S.M. Matyas and N. Zunic, 1999. Decentralized group key management for secure multicast communications. Comput. Commun., 22: 1183-1187. DOI: 10.1016/S0140-3664(99)00121-8

Rafaeli, S. and D. Hutchison, 2003. A survey of key management for secure group communication. J. ACM Comput. Surveys, 35: 309-329. DOI: 10.1145/937503.937506

Rivest, R.L., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Mag. Commun.. ACM., 21: 120-126. DOI: 10.1145/359340.359342

Sahar, N.B.M., S. Ardi, S. Kazuhiko, M. Yoshiomi and M. Hirotsugu, 2010. HAZOP analysis management system with dynamic visual model aid. Am. J. Applied Sci., 7: 943-948. DOI: 10.3844/ajassp.2010.943.948

Selcuk, A.A. and D. Sidhu, 2002. Probabilistic optimization techniques for multicast key management. Comput. Netw., 40: 219-234. DOI: 10.1016/S1389-1286(02)00252-9

Sharma, S., P. Sharma and R.S. Dhakar, 2011. RSA algorithm using modified subset sum cryptosystem. Proceedings of the 2nd International Conference on Computer and Communication Technology (ICCCT), Sep. 15-17, IEEE Xplore Press, Allahabad, pp: 457-461. DOI: 10.1109/ICCCT.2011.6075138

Sherman, A.T. and D.A. McGrew, 2003. Key establishment in large dynamic groups using one-way function trees. IEEE Trans. Software Eng., 29: 444-458. DOI: 10.1109/TSE.2003.1199073

Tu, F.K., C.S. Laih and H.H. Tung, 1999. On key distribution management for conditional access system on pay-TV system. IEEE Trans. Consumer Elect., 45: 151-158. DOI: 10.1109/30.754430

Waldvogel, M., G. Caronni, D. Sun, N. Weiler and B. Plattner, 1999. The versakey framework: Versatile group key management. IEEE J. Selected Areas Commun., 17: 1614-1631. DOI: 10.1109/49.790485

Wong, C.K., M. Gouda and S.S. Lam, 2000. Secure group communications using key graphs. IEEE/ACM Trans. Netwo., 8: 16-30. DOI: 10.1109/90.836475

Zhu, S. and S. Jajodia, 2003. Scalable Group rekeying for secure multicast: A survey. Lecture Notes Comput. Sci., 2918: 833-833. DOI: 10.1007/978-3-540-24604-6_1