Secure Group Key Management using Ciphertext in MANETs

P Swetha[1], Dr. P. Premchand[2], Dr. P. Naveen Kumar[3]

[1] Department of Computer Science and Engineering, JNTUH College of Engineering Jagtial, Nachupally (Kondagattu), Karimnagar, 505501, TS, India.
[2] Department of Computer Science and Engineering, University College of Engineering, Osmania University, Hyderabad, 500 007, TS, India
[3]Assistant Professor, Department of ECE, University College of Engineering, Osmania University, Hyderabad

***Abstract**: The modern developments in Mobile AdhocNETworks (MANETs) suffer from efficient and secure broadcasting of messages to a distant system. The main obstacles for the secured information transfer are restricted communication from the cluster/group nodes to the sender and unavailability of a trusted public/private key generation center. This paper proposes a technique that uses a combination of broadcast encryption technique and Cluster/Group key agreement to overcome the obstacles. Among the proposed techniques each member maintains one public/secret key pair, seeing that the remote sender broadcasts to the subgroup. Though the non-intended member conspire they will not extractinformation that boosts security of the message. The Cluster /Group size is independent of computation and communication overhead that decreases the delay. The proposed technique provides competent approach to add or delete members of a cluster and a flexible rekeying strategy. In case of conspiracy it provides security to the broadcasted data. Simulation results are given using NS2.*

***Keywords**: MANET, network lifetime, throughput,Network delay.*

I. Introduction

A mobile ad hoc networkMANET is an self-directed system of mobile nodes coupled by wireless links. Every node operates not only as an end system, however conjointly as a router to forward packets on behalf of others. The absence of centralized administration and also the infrastructure less nature build MANETs smart for emergencies, disaster relief efforts, military, and quick deployment communications. The security of most typical networks depends on the existence of a specialised network administration that defines the security framework and provides the infrastructure for implementing it. The lack of any centralized network management or certification authority makes MANET susceptible to infiltration, eavesdropping, interference, and so on. Security in MANET is an important element to provide the network with the essential functions like routing and packet forwarding. Efficient and strong key management services are importantto provide MANETs with security, due to the open and distributed nature of MANET, it is essential to provide access control to sensitive data to improvise security from droppers and malicious attackers. The restricted communication from group to sender is difficult in providing security. The lack of perfectly trusted third party key generation center plays a significant role in providing security. The major security significance within the group oriented communications with access control is that the key management. The prevailing key management system has two approaches brought up as cluster key agreement and key distribution system. In Cluster/Group key agreement a group of users talk over a typical secret key via an open insecure network. Then any member will encipher the confidential message with the shared secret key and will only group members can decode. A large range of Cluster/Group key agreement protocols are proposed [2-13].

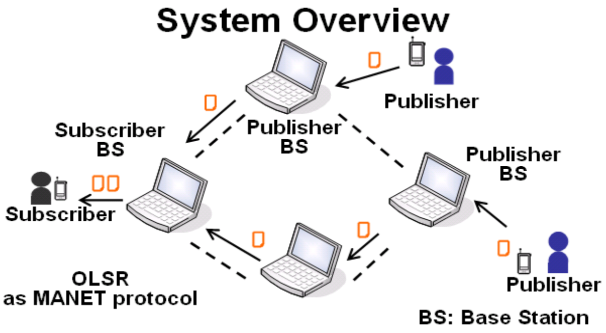


Figure 1. The structure of MANET

The major challenges in devising such systems are to overcome the obstacles of the possibly restricted communication from the cluster/group to the sender, the inaccessibility of a totally trustworthy key generation center, and therefore the dynamics of the sender. The predominant key management paradigms cannot alter these challenges effectively.

II. Literature Survey

The major security significance in group-oriented communications with access management is key management. Existing key management systems in these situations are primarily enforced with two approaches stated as cluster key agreement (or cluster key exchange) and key distribution systems are the additional powerful notion of broadcast encryption. Both are active analysis areas having generated massive various bodies of literature.

Cluster key agreement permits a group of users to negotiate a standard secret key via open insecure networks. Then any member will encipher any confidential message with the shared secret key and only the cluster members will decipher. In this method, a confidential internal broadcast channels are often established without looking forward to a centralized key server to come up with and circulate secret keys to the potential members. A large range of cluster key agreement protocols are projected [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

The earlier efforts [2], [3] targeted on efficient establishment of the initial cluster key. Later studies [4] modify efficient member joins, however the value for a member quit remains relatively high. A tree key structure has been further projected and improved to attain higher potency for member joins and member quits [5], [7], [11]. The theoretical analysis in [14] proves that, for any tree-based cluster key agreement method, the bound of the worst-case cost is O(log n) rounds of interaction for member join or quit, where n is the range of cluster members. This optimum round potency was recently achieved in [12]. By employing a ring-based key structure, the up-to-date proposal in [13] breaks this round barrier because only a continuing range of rounds is needed for member changes.

Design Aspects To Improve Security In Manets

This paper proposes an easy yet effective scheme for improvement of QoS in MANETs for information security. The projected technique is a blend of broadcast encryption and cluster key agreement strategy. Every cluster is given a public/shared key by the trusted third party key distribution system, once a sender is prepared to transmit message to an intended cluster the message is encrypted using broadcast encryption and public/shared key of the cluster and it is broadcasted within the network. The non-intended cannot decode the message with their key. Therefore in case of collusion, information will not be lost and can be decrypted solely by the supposed cluster's public/shared key. The members of a clusterare added or deleted as follows.

Basic system model proposed:

Consider a cluster composed of N users, indicated by {U1,
・・・, UN}. . A sender would like to send secret messages to a receiver subset S of the N users, where the dimensions of S is n ≤ N. The problem is how to modify the sender to efficiently and securely finish the transmission with the subsequent constraints:

1) it is hard to deploy a key generation authority absolutely trustworthy by all users and potential senders in open network settings.
2) The transmission of messages from the receivers to the sender is restricted, e.g. within the battleground communication setting.
3) N could be terribly large and up to millions, as an example, in vehicular ad hoc networks.
4) Both the sender and also the receiver sets are dynamic because of ad hoc communication. According to the application scenarios, there are also some mitigating options that will be exploited for finding the problem:

1) n is typically a small or medium value
2) The receivers are cooperative and communicated via efficient native (broadcast) channels.
3) A partially trusted authority, e.g. a public key infrastructure, is offered to authenticate the receivers (and the senders).

The above drawback is addressed by formalizing a new key management paradigm stated as Cluster key agreement based mostly broadcast encryption. The system design is illustrated in above figure. The potential receivers are connected along with efficient local connections. Through communication infrastructures, they will additionally connect with heterogeneous networks. Every receiver contains a public/secret key pair. The public key is certified by a certificate authority however the secret key is kept solely by the receiver. A distant sender will retrieve the receiver's public key from the certificate authority and confirm the authenticity of the public key by checking its certificate, which suggests that no direct communication from the receivers to the sender is required. Then the sender will send secret messages to any chosen subset of the receivers. Next formally it defines the model of Cluster key agreement primarily based broadcast encryption. The definition incorporates the up-to-date interpretation of cluster key agreement [19] and public-key broadcast encryption [32]. Since the core of key management is to securely transfer a session key to the supposed receivers, it is enough to outline the system as a session key encapsulation mechanism. Then the sender will simultaneously encipher any message under the session key and only the supposed receivers will decipher.

Algorithms

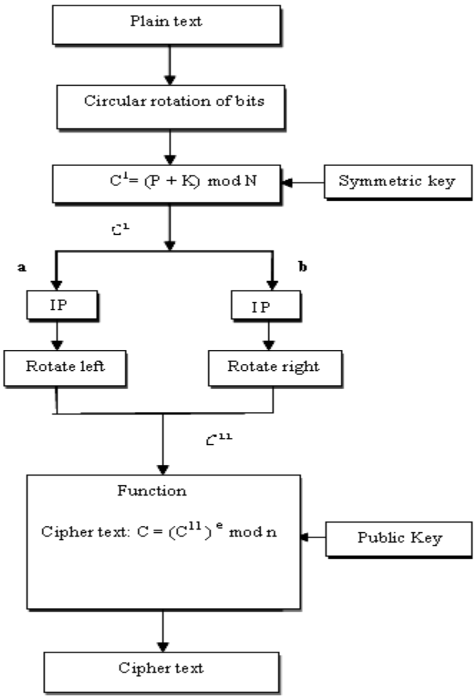


Fig 2. The flow diagram of generation of Encrypt data.

KeyGen(i, n,N): This key generation algorithm is run byeach user Ui €{U1 , ・ ・ ・ , UN} to generate the public/private key pair. A user takes as input the system parameters n,N and the index i €{1, ・ ・ ・ ,N}, and outputs _pki, ski_ as public/secret key pair. Denote {_pki, ski_|Ui € S €{U1, ・ ・ ・ , UN}} by _pki, ski_S and similarly, {_pki_|Ui € S €{U1, ・ ・ ・ , UN}} bypki_S. Here, implicitly omit the input security parameter λ. actually, n,N are polynomials in λ. Assume that every user's public key is certified by a publicly available certificate authority so that anyone can retrieve the public keys and verify their genuineness. This is often plausible as public key infrastructures are a standard part in several systems supporting security services. The key generation and therefore the registration to the certificate authority will be done offline before the online message transmission by the sender.

Encryption(S, _pki_S):

It is run by any sender who may ormay not be in {U1, ・ ・ ・ , UN}, provided that the sender knows the public keys of the potential receivers. It takes as input a recipient set S €{U1, ・ ・ ・ , UN} and the public key pki for Ui €S. If |S| = n, it outputs a pair _Hdr, k_ where Hdr is called the header and k is the message encryption key. (S,Hdr) is sent to the receivers. This algorithm incorporates the functionality of the encryption procedure in traditional broadcast encryption systems.

Decryption (Uj(skj)S,Hdr, _pki_S): This algorithm is conjointly run by the intended receivers to extract the secret session key k hidden within the header. Every receiver Uj in private inputs the secret key skj . The common inputs are the header Hdrand the public keys of receivers in the recipient set S. If |S| = n, every receiver in S outputs a similar session key k. This procedure incorporates a standard cluster key agreement protocol. It exploits the cooperation of the receivers with efficient local connections. It next justifies the assumptions on trustworthy authorities and restricted communication from the receivers to the sender during this key management paradigm. At a primary look, the new paradigm appears to need a trusted third party as its counterpart in traditional broadcast cryptography systems. A more in-depth look shows there is a distinction. In an exceedingly traditional broadcast cryptography system, the third party must be totally trusted, that is, the third party is aware of the secret keys of all cluster members and may browse any transmission to any subgroup of the members. This type of totally trusted third party is difficult to implement in open networks.

In contrast, the third party in this key management model is only partially trusted. In other words, the third party is aware and certifies the general public key of every member. This type of partially trusted third party has been enforced and is known as Public Key Infrastructure (PKI) in open networks. Second, the new key management paradigm apparently needs a sender to know the keys of the receivers, which may want communications from the receivers to the sender as in traditional cluster key agreement protocols. However, some subtleties should be detected here. In traditional cluster key agreement protocols, the sender must at the same time keep connected with the receivers and direct communications from the receivers to the sender is required. This is often troublesome for a distant sender. On the contrary, in this key management paradigm, the sender solely must acquire the receivers' public keys from a third party and no direct communication from the receivers to the sender is needed, which is implemented with precisely the existing PKIs in open networks. Hence, this is possible for a distant sender. Moreover, a sender does not need to frequently contact the third party or keep a large number of keys since a sender sometimes communicates to a relatively fixed cluster in practice. For instance, a department manager typically communicates with the subordinates, superiors and other department managers, however rarely must send secret messages to all staff members.

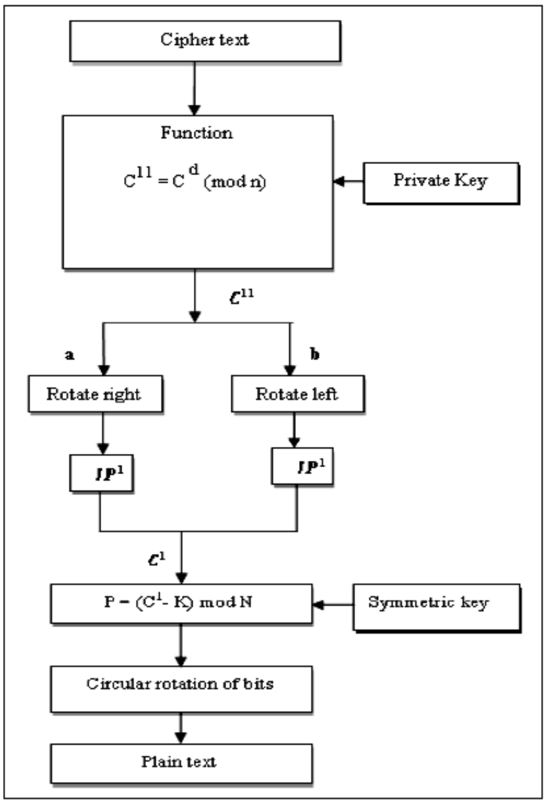
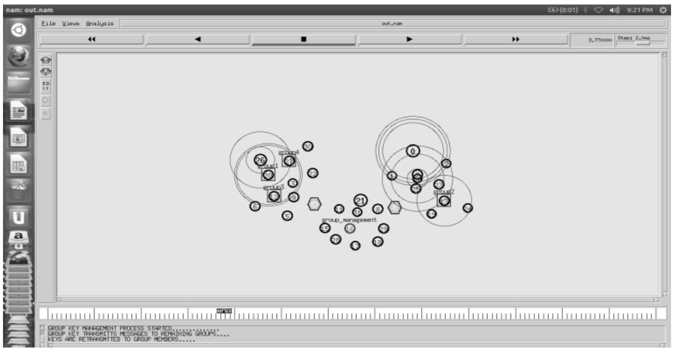


Fig 3. The flow diagram of generation of Decrypt data.

III. Results & Discussion

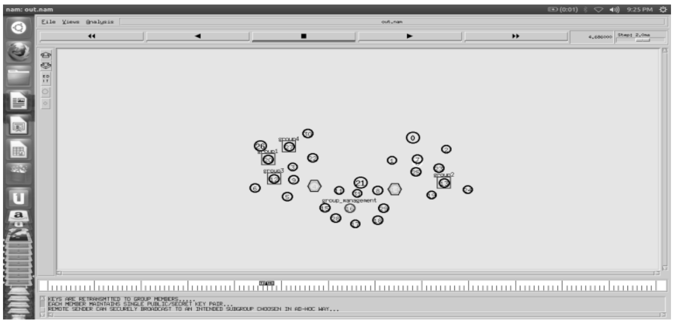
In this paper, each receiver incorporates a public/secret key pair. The general public key's certified by a certificate authority, however the secret key is kept only by the receiver. A distant sender will retrieve the receiver's public key from the certificate authority and validate the authenticity of the public key by checking its certificate, which suggests that no direct communication from the receivers to the sender is necessary. Then, the sender will send secret messages to any chosen set of the receivers. The results are extracted by executing NS2 code



```
GROUP KEY MANAGEMENT PROCESS STARTED..............
GROUP KEY TRANSMITTS MESSAGES TO REMAINING GROUPS....
KEYS ARE RETRANSMTTED TO GROUP MEMBERS.....
```

Fig 4. Keys are being transmitted among the groups

The fig4. gives the structure of MANET after encryption code generating key and is being transmitted among the group.



```
KEYS ARE RETRANSMTTED TO GROUP MEMBERS.....
EACH MEMBER MAINTAINS SINGLE PUBLIC/SECRET KEY PAIR...
REMOTE SENDER CAN SECURELY BROADCAST TO AN INTENDED SUBGROUP CHOOSEN IN AD-HOC WAY...
```

Fig 5. Remote sender sending messages only to the intended sub group among the network

From the fig 5,the senders information can be reached to the destination with decryption key the node can decrypt the encoded data which is transmitted by the sender.

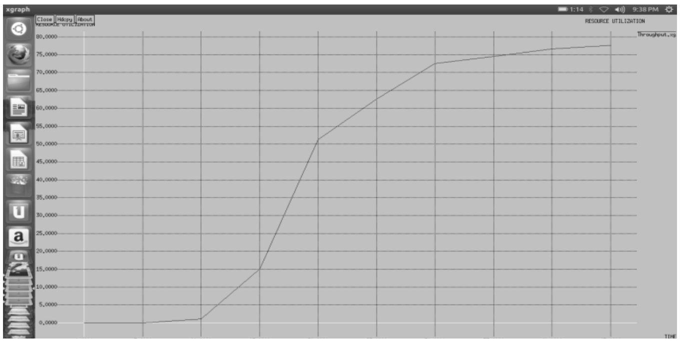


Fig 6. Resource utilization graph

The results obtained from the analysis of fig 6, indicates that as time progresses the network setup utilizes the maximum resources of the system

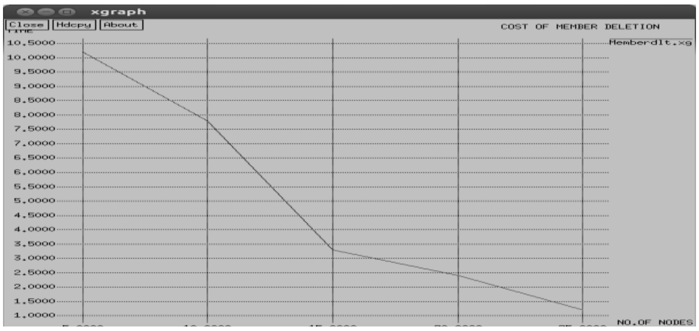


Fig 7. Cost of member deletion graph

The results obtained from the analysis of figure 7,indicates that as the number of nodes increases the members has to be deleted.

IV. Conclusion

The proposed new key management system permits the sender to partially depend on the third party key generation center to confirm safe and secure broadcasting of information between sender and intended sub cluster. Collusions within the network cannot extract the information which reinforces security. QoS is improved because the cluster size is independent of the computation and communication overhead that decreases the delay.

References

[1]. Qianhong Wu, Member, IEEE, Bo Qin, Lei Zhang, Josep Domingo-Ferrer, Fellow, IEEE, and JesúsA. Manjón "Fast Transmission to Remote Cooperative Groups: A New Key Management Paradigm"- IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 21, NO. 2, APRIL 2013.

[2]. M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," Adv. Cryptol., vol. 950, EUROCRYPT'94, LNCS, pp. 275–286, 1995.

[3]. 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, Sep. 1999.

[4]. 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. 2000.

[5]. A. Sherman and D. McGrew, "Key establishment in large dynamic groups using one-way function trees," IEEE Trans. Softw. Eng., vol. 29, no. 5, pp. 444–458, May 2003.

[6]. Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stanton, and G. Tsudik, "Secure group communication using robust contributory key agreement," IEEE Trans. Parallel Distrib. Syst., vol. 15, no. 5, pp. 468–480, May 2004.

[7]. Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," Trans. Inf. Syst. Security, vol. 7, no. 1, pp. 60–96, Feb. 2004.

[8]. Y. Sun,W. Trappe, and K. J. R. Liu, "A scalablemulticastkeymanagement scheme for heterogeneous wireless networks," IEEE/ACMTrans. Netw., vol. 12, no. 4, pp. 653–666, Aug. 2004.

[9]. W.Trappe, Y.Wang, andK. J. R.Liu, "Resource-aware conference key establishment for heterogeneous networks," IEEE/ACM Trans. Netw.,vol. 13, no. 1, pp. 134–146, Feb. 2005.

[10]. 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, Apr. 2006.

[11]. Y. Mao, Y. Sun, M. Wu, and K. J. R. Liu, "JET: Dynamic join-exittree amortization and scheduling for contributory key management," IEEE/ACM Trans. Netw., vol. 14, no. 5, pp. 1128–1140, Oct. 2006.

[12]. W. Yu, Y. Sun, and K. J. R. Liu, "Optimizing the rekeying cost for contributory group key agreement schemes," IEEE Trans. Depend. Secure Comput., vol. 4, no. 3, pp. 228–242, Jul.–Sep. 2007.

[13]. R. Dutta and R. Barua, "Provably secure constant round contributory group key agreement in dynamic setting," IEEE Trans. Inf. Theory, vol. 54, no. 5, pp. 2007–2025, May 2008

[14]. J. Snoeyink, S. Suri, and G. Varghese, "A lower bound for multicast key distribution," Proc. IEEE INFOCOM, pp. 422–431, 2001.

[15]. I. Ingemarsson, D. T. Tang, and C. K.Wong, "A conference on key distribution system," IEEE Trans. Inf. Theory, vol. 28, no. 5, pp. 714–720, Sep. 1982.

[16]. M. Abdalla, Y. Shavitt, and A. Wool, "Key management for restricted multicast using broadcast encryption," IEEE/ACM Trans. Netw., vol. 8, no. 4, pp. 443–454, Aug. 2000.

[17]. B. M. Macq and J.-J. Quisquater, "Cryptology for digital TV broadcasting," Proc. IEEE, vol. 83, no. 6, pp. 944–957, Jun. 1995.

[18]. J. Lotspiech, S. Nusser, and F. Pestoni, "Anonymous trust: Digital rights management using broadcast encryption," Proc. IEEE, vol. 92, no. 6, pp. 898–909, Jun. 2004.

[19]. A. Fiat and M. Naor, "Broadcast encryption," Adv. Cryptol., vol. 773, CRYPTO'93, LNCS, pp. 480–491, 1993.

[20]. C. K. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," IEEE/ACM Trans. Netw., vol. 8, no. 1, pp. 16–30, Feb. 2000.