

AGPM: An Authenticated Secure Group Communication Protocol for MANETs

B. Gopalakrishnan¹, T. V. P. Sundararajan² and Dr. A. Shanmugam²

¹Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

Email: bgopal1977@gmail.com

²Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

Email { tvp_zen@yahoo.co.in , dras_bit@yahoo.com }

Abstract-Secure group communication is a challenging task with respect to MANET's, authentication of mobile nodes, group key establishment and rekeying for secure information exchange and QoS in data transfer. In this paper we authenticate the mobile nodes through transitive signature scheme in the routing phase of AODV protocol. For a secure group communication we establish a collaborative group key with the members participating in the route path to the destination. The nodes are dynamic in nature, in which any new node can join in the group or leave the group. Instead of performing individual Rekeying operations, it is performed at a particular time interval. Performance of the group communication is compared with the existing protocols. The analysis is made with respect to the throughput, rekeying time, delay, overhead and communication cost. The simulation result shows that our protocol enjoys greater advantage over other protocols in the literature.

Index Terms: Ad-hoc Networks, Secure Group communications, Routing, Key Generation and Rekeying

I. INTRODUCTION

A Mobile Ad-hoc NETWORK (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies. In mobile ad hoc network, nodes can directly communicate with all the other nodes within their radio frequency range; whereas nodes that are not in the direct communication range use intermediate node(s) to communicate with each other. In these two situations, all the nodes that have participated in the communication automatically form a wireless network, therefore this kind of wireless network can be viewed as mobile ad hoc network. These properties make MANET very suitable for group communications.

The routing is a process of identifying the path between the two nodes, which is accomplished through different protocols such as AODV, DSR and DSDV for mobile ad-hoc networks. These protocols do not provide the authentication scheme for the mobile nodes that participates in the route discovery in a group.

In this section, we introduce the basics of the AODV routing protocol. AODV is a simple and

efficient on-demand ad hoc routing protocol. Basically, it uses RREQ (route request), RREP (route reply) and RRER (route error) messages to accomplish route discovery and maintenance operations. It also utilizes sequence numbers to prevent routing loops. Routing decision making is based on sequence numbers and routes maintained in each node's routing table. The routing operations of AODV generally consists of two phases namely route discovery and route maintenance. Route discovery is performed through broadcasting RREQ message. Whenever a node needs to send data packets to a destination, it first checks if it has an existing route in the routing table. If not, the source node will initiate a RREQ and broadcast this request to all the neighbours. Then neighboring nodes will update their routing table according to the received message. When RREQ reaches the destination, a RREP will be generated by the destination node as a response to RREQ. The RREP will be transmitted back to the originator of RREQ in order to inform the route. If an intermediate node has an active route towards destination, it can reply the RREQ with a RREP, which is called Gratuitous Route Reply. The intermediate node will also send an RREP to destination node. The RREP will be sent in reverse route of RREQ if a bidirectional link exists. This is another way of disrupting topology by creating route loops.

II. RELATED WORKS

A Burmester and Desmedt Protocol [BD]

Burmester and Desmedt Protocol [1] is an extension of the Diffie-Hellman key distribution system.

$$K_i = (z_{i-1})^{n_i} \cdot X_i^{n-1} \cdot X_{i-1}^{n-2} \cdot \dots \cdot X_{i-2} \text{ mod } p.$$

That is each group user will come up with the same secret key $k = g^{r_1r_2+r_2r_3+\dots+r_{m-1}r_m} \text{ mod } p$, which is the group key shared by all group members.

In BD scheme, each group member needs to perform $n+1$ exponentiations. It also requires a total number of $2n$ broadcast messages.

B Group Diffie–Hellman Key Exchange

Group Diffie–Hellman key exchange [2] is an extension of the DH key agreement protocol that supports group operations. The DH protocol is used for two parties to agree on a common key.

In this protocol, instead of two entities, the group may have n members. The group agrees on a pair of primes (q and α) and starts calculating in a distributive fashion for the intermediate values.

The first member calculates the first value (α_{x1}) and passes it to the next member. Each subsequent member receives the set of intermediary values and raises them using its own secret number generating a new set.

A set generated by the i^{th} member will have i intermediate values. For example, the fourth member receives the set:

$$\{\alpha^{x2x3}, \alpha^{x1x3}, \alpha^{x1x2}, \alpha^{x1x2x3}\}$$

and generates the set

$$\{\alpha^{x2x3x4}, \alpha^{x1x3x4}, \alpha^{x1x2x4}, \alpha^{x1x2x3}, \alpha^{x1x2x3x4}\}$$

The setup time is linear (in terms of n) since all members must contribute in generating the group key. Therefore, the size of the message increases as the sequence reaches the last member and more intermediate values are necessary. With that, the number of exponential operations also increases.

C Logical Key Hierarchy [LKH]

Perrig and Kim et al. [3] also use a logical key hierarchy to minimize the number of key held by group members. The difference here is that group members generate the keys in the upper levels using the Diffie–Hellman algorithm rather than using a one-way function. The key of each node is generated from its two children ($k = \alpha^{klkr} \text{ mod } p$).

D Tree Based Group Diffie- Hellman

Y. Kim, A. Perrig, and G. Tsudik,[4] [TGDH] proposed a novel approach to group key agreement by blending binary key trees with Diffie–Hellman key exchange. The resultant protocol suite is very simple, fault-tolerant and secure. We unify the following two important trends in group key management:

1) The use of so-called *key trees* to compute efficiently and update group keys.

2) The use of Diffie–Hellman key exchange hybrids to achieve provably secure and fully distributed protocols.

III. PROPOSED SYSTEM MODEL

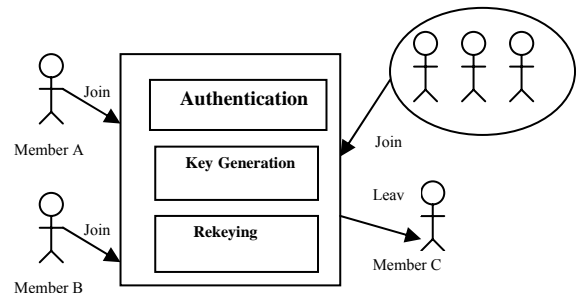


Figure. 1: Model of AGPM

A Authentication through Transitive scheme

In this section, we introduce transitive signature scheme to authenticate the nodes at route discovery phase. It was originally used to dynamically build an authenticated graph, edge by edge. The signer, having secret key sk and public key pk , can at any time pick a pair i, j of nodes and create a signature of $\{i, j\}$, thereby adding edge $\{i, j\}$ to the graph.

In addition, given a signature of an edge $\{i, j\}$ and a signature of an edge $\{j, k\}$, anyone in possession of the public key can create a signature of the edge $\{i, k\}$.

Setup

Each node in the network agrees with the following parameters:

- Large prime p and q such that q divides $p-1$
- Two generates g and h of subgroup Gq of order $q \in \mathbb{Z}_p^*$ such that the base- g logarithm of h modular p is infeasible for others to compute.

Then each node n_i does the following:

1. Randomly choose two values x_i and y_j from \mathbb{Z}_p^* ;
2. Compute $\alpha_i = x_i \text{ mod } q$ and $\beta_i = y_i \text{ mod } q$;
3. Compute $v_i = g_i^x h_i^y \text{ mod } p$;
4. Broadcast α_i and β_i to node's neighbors.
5. Upon the receipt of α_j and β_j from each neighbor, node i compute:

$$\alpha_{ij} = x_i - x_j \text{ mod } q$$

and $\beta_{ij} = y_i - y_j \text{ mod } q$
 6. Node i records in its memory the quadruple:
 $(v_i, v_j, \alpha_{ij}, \beta_{ij})$

Sign

To sign the path between node A and node B, node B must have received α_A, β_A , and v_A from node A. Then node B computes the signature as:

$$\alpha_{AB} = x_A - x_B \text{ mod } q \text{ and}$$

$$\beta_{AB} = y_A - y_B \text{ mod } q$$

Node B publishes the quadruple as the signature:

$$(v_A, v_B, \alpha_{AB}, \beta_{AB})$$

Verify

Any node can verify the previous signature by checking:

$$V_A = v_B g^{\alpha_{AB}} h^{\beta_{AB}} \text{ mod } q$$

Path Composing

When the next hop node C receives the signature between node A and node B, it first verifies the validity of the signature in order to ensure that node B does have an active route towards node A. Then node C generates a transitive signature over the received one so as to incorporate itself into the path.

Given signature $(v_A, v_B, \alpha_{AB}, \beta_{AB})$, node C retrieves the quadruple $(v_B, v_C, \alpha_{BC}, \beta_{BC})$ and computes the new transitive signature $(v_A, v_C, \alpha_{AC}, \beta_{AC})$ as:

$$\alpha_{AC} = \alpha_{AB} - \alpha_{BC} \text{ mod } q \text{ and}$$

$$= x_A - x_C \text{ mod } q$$

$$\beta_{AC} = \beta_{AB} - \beta_{BC} \text{ mod } q \text{ and}$$

$$= y_A - y_C \text{ mod } q$$

The signature for the path from node A to node C is:
 $(v_A, v_C, \alpha_{AC}, \beta_{AC})$

The use of the transitive signature scheme to enable the route aggregation has one big benefit. It enables the authentication of both originator and gratuitous replier in one signature. In delegation by warrant, the token is signed with the routing packet by the gratuitous replier. Thus, the authentication of the gratuitous replier has to be done by verifying the conventional signature, and the token which is signed using conventional signature scheme has to be verified at the same cost. By using transitive signatures, the originator and replier can be authenticated at the same time. However, the use of

the transitive signature scheme to enable gratuitous reply authentication requires the cost of exchanging public key quadruples and computing the path signatures between neighboring nodes. It is considered to be the major drawback of this application.

B Interval-Based Distributed Rekeying Algorithms

Interval-based rekeying algorithm proposes that rekeying is done at equal intervals instead of each leave and join operation.

Notations

Let T denote the existing key tree. Assume that $L \geq 0$ existing members $M^l = (M_1^l, \dots, M_L^l)$ wish to leave and $J \geq 0$ new members $M^j = (M_1^j, \dots, M_J^j)$ wish to join the group within a rekeying interval.

Rebuild Algorithm

At the beginning of every rekeying interval, we reconstruct the whole key tree with all existing members that remain in the communication group, together with the newly joining members. The resulting tree is a *left-complete* tree. The pseudo-code of the Rebuild algorithm to be performed by every member in the group.

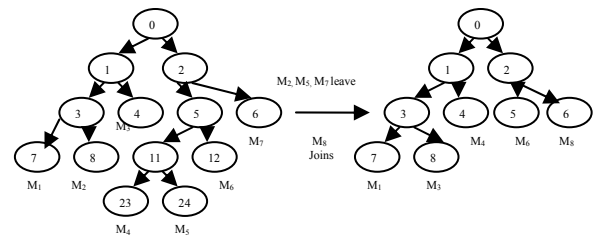


Figure. 2: Example for Rebuild

Rebuild (T, M^l , J, M^j , L)

1. Get all members from T and store them in M'
2. Remove the L leaving members in M'
From M'
3. Add the J new members in M^j to M'
4. Create a new binary tree T' based on members in M' and set $T = T'$
5. Select all members to be sponsors
6. Rekey renewed nodes and broadcast new blinded keys in T

Figure. 2 shows the scenario where members M2, M5, and M7 wish to leave and a new member M8 wishes to join the communication group. Based on the algorithm, the resulting key tree consists of five members and has all non-leaf nodes renewed. Besides, the sponsors include all the five members.

C Performance evaluation

The above protocol (AGPM) is implemented in ns2 simulator. We evaluate the performance of the interval based algorithm in simulation based experiment. We study their performance in more general setting and also compare the performance of our protocols with other approach specified in the related works.

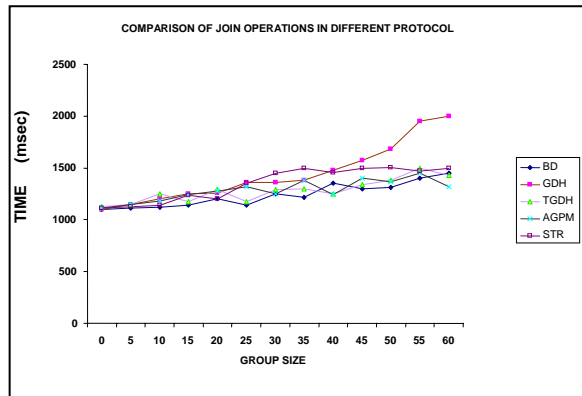


Figure.3: Comparison of join operations with other protocols

The above figure shows the performance of protocol AGPM with other protocols. The y axis shows the time taken to generate the group key and x axis shows the number of members participated in the group key generation. The time taken is normal when the size of the group size increases.

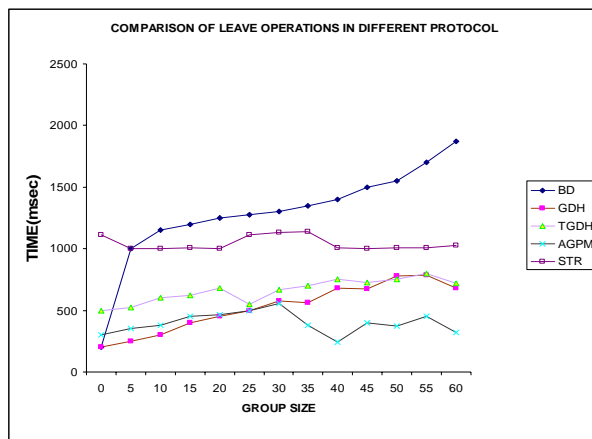


Figure. 4: Comparison of leave operation with other protocols

other protocols

The above figure shows the time taken to reconstruct the group key when the node leaves the group is directly proportional to the size of the group members.

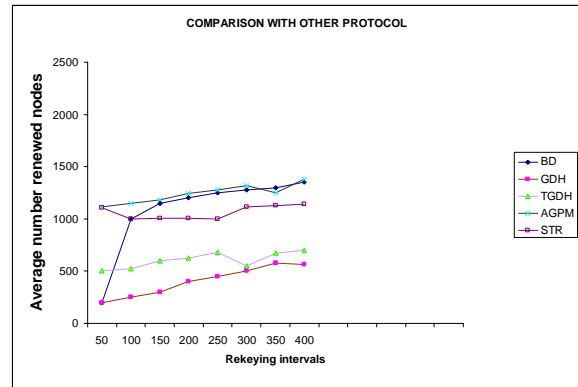


Figure. 5: Comparison with rekeying operation

The above figure shows the no. of nodes changed during the rekeying operation, ie, the time for rekeying increases gradually with respect to the size of members in the group.

IV. CONCLUSION

This paper presents a novel scheme to implement AODV routing protocol for secure group communication. It uses an efficient way of authenticating the nodes in the route discovery process using transitive signature scheme. The collaborative group key is generated by TDGH and the rekeying is done in an interval based scheme. The performance of the above protocol shows that the data transferred between the group members is secured.

REFERENCES

- [1] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. *Advances in Cryptology – EUROCRYPT '94*, 1995.
- [2] STEINER, M., TSUDIK, G., AND WAIDNER, M. 1996. Diffie-Hellman key distribution extended to group Communication. In *SIGSAC Proceedings of the 3rd ACM Conference on Computer and Communications Security*. (New Delhi, India, Mar.), ACM, New York, pp. 31–37.
- [3] PERRIG, A.. Efficient collaborative key management protocols for secure autonomous group communication. In *Proceedings of the International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*. (Hong Kong, China, July), pp. 192–202, 1999.
- [4] Perrig, A and G. Tsudik. Tree-Based Group Key Agreement. *ACM Trans. on Information and System Security*, 7(1):60–96, Feb 2004.
- [5] Y. Kim, A. Perrig, and G. Tsudik. Communication-

- Efficient Group Key Agreement. In *Proceedings of the 17th International Information Security Conference IFIP SEC'01*, Nov 2005.
- [6] Y. Kim, G. S. Setia, S. Koussih, and S. Jajodia. Kronos: A Scalable Group Re-Keying Approach for Secure Multicast. In *Proc. of IEEE Symposium on Security and Privacy*, May 2000.
- [7] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam. Reliable Group Rekeying: A Performance Analysis. *Proc. of ACM SIGCOMM*, August 2001
- [8] C. E. Perkins, E. M. Royer, and S. R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETFINTERNET DRAFT, MANET working group. Feb. 2003.
- [9] S. Micali and R. Rivest. Transitive Signature Schemes. In B. Prneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 236-243 Springer-Verlag, 2002.
- [10] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam. Batch Rekeying for Secure Group Communications. In *Proc. of Tenth International World Wide Web Conference (WWW10)*, May 2001.
- [11] Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6): PP 644–654, 1996.