Trust Metrics for Group Key Management in Malicious Wireless Networks

[1]V.Bhuvaneswari, [2] Dr. M. Chandrasekaran

[1]Government Polytechnic College, Department of Computer Engineering, Dharmapuri, India
[2]Government College of Engineering,
Department of Electronics and Communication Engineering, Bargur, India

Abstract: Group communication is accomplished with the aid of group key management by preventing non-group members from accessing data exchanged. For improving security in wireless networks trust information has been widely used. In this work trust is used as a criterion for cluster formation.Direct trust and indirect trust is computed to identify Cluster Heads (CH) and the concept of backup cluster head is introduced for effective key management. Simulation results show the proposed method performs better in group key management than other techniques found in literature.

Keywords: Mobile ad hoc networks (MANETs), Dynamic Source Routing (DSR), Malicious Nodes, Clustering and Key Management.

I. Introduction

In Adhoc network, each node acts like a router and forwards the packets from one peer node to other peer nodes. The wireless channel is accessible in both legitimate network users and for malicious attackers. As a result, there is a blurry boundary which separates the inside network from the outside world [1]. Also in MANET, all networking functions including routing and data transmission, are performed by the nodes without the need for a central point to control and organizes the resource management process. Therefore security is a very challenging task. Security vulnerabilities for a network includes of the following aspects: Confidentiality, integrity, authentication, non-repudiation [2].

Encryption is the process of converting a plain text “unhidden”, into a cryptic text “hidden” to secure it against data thieves. This process also consists of another part where cryptic text needs to be decrypted on the other end to be understood.

Many encryption algorithms are available and used in information security widely. They can be categorized into Symmetric (private) and Asymmetric (public) keys encryption algorithms. In Symmetric keys encryption or secret key encryption, only one key is used for encrypting and decrypting the data. In Asymmetric keys, two keys are used such as private and public keys. Public key is used for an encryption and private key is used for the decryption technique (for e.g. RSA and ECC) [4].

Key management is the most important issues in security protocol design. In a secure group communication, key management techniques are used to provide a correct distribution and easy maintenance of cryptographic keys. The cryptographic keys, which can be used to encrypt Group Key (GK), are called as Key Encryption Key (KEK). As a result, key management problem can be considered as the secure and efficient distribution of KEKs and GK to only valid members [5]. The KEK is derived directly from the Authentication Key (AK), and it is 128 bits long. The KEK is not used for encrypting traffic data. Traffic Encryption Key (TEK) is generated as a random number generating in the Base Station (BS) using the TEK encryption algorithm where KEK is used as the encryption key. TEK is then used for encrypting the data traffic [6]. The TEK distribution mode is used to securely distribute TEKs only. The TEK distribution mode uses asymmetric-key based proxy re-encryption schemes, and the data transfer mode uses symmetric-key based proxy re-encryption schemes [7].

The study is organized as follows: Section 2 reviews some of the related works available in the literature, section 3 details the methodology used in this investigation, section 4 is proposed method, section 5 gives the results and section 6concludes the paper.

II. Literature Survey

Cryptography plays an integral role in secure communication and is usually the strongest link in the chain of security. Multilanguage cryptography, an advancement of classical cryptography, may evolve as a choice of classical cryptography lovers seeking a better security. Srivastava, et al., [8] proposed an algorithm in Multilanguage approach, which generated different cipher texts at different time for the same plaintext over a range of languages supported by Unicode. It has a better frequency distribution of characters in the cipher text than previous work on this approach. Bouassida, et al., [9] showed the specific challenges towards key management protocols for securing multicast communications in ad hoc networks, and provides taxonomy of these protocols in MANETs. A new approach, called BALADE, was also presented. It was based on a sequential multi-sources model, and taken into account both localization and mobility of nodes, while optimizing energy and bandwidth consumptions.

Chen, et al., [10] proposed a scheme for secure group key management using uni-directional proxy re-encryption in which each group member holds just one secret auxiliary key and logN public auxiliary keys. This scheme was immune to the collusion attack of other members. Rahman, et al., [11] proposed a new key management protocol which provides a support for both pair-wise and group-wise key management with identity pre-distributed secret. This protocol was efficient in terms of communication and storage overhead.

Gomathi and Parvathavarthini [12] proposed new Cluster Based Tree (CBT) for the secure multicast key distribution. DSDV routing protocol was used for collecting its one hop neighbours to form a cluster. John and Samuel [13] proposed a hierarchical key management scheme using a stable and power efficient cluster management technique. The overhead on centralized server has been reduced with these techniques.

Niu [14] proposed a scheme using soft encryption combined with multipath routing to provide security of data transmission over MANETs. This approach substantially reduces the computational overhead of using cryptographic method to encrypt entire message while security has been ensured.

Wu, et al., [15] introduced a MANET setting adapted, simple group key management scheme in which a multicast tree is formed for efficiency. To achieve fault tolerance, two multicast trees are constructed and maintained parallels. When one tree links is broken, it is substituted by the other. One tree is named blue and the other red. Group members act as group coordinators in rotation to compute/distribute intermediate keying materials to members through active tree links. This work is undertaken in rounds with the coordinator being selected in a distributed way. The latter is also responsible to maintain multicast group connections. Group coordinators compute/distribute intermediate keying materials through the underlying tree links to all members.

An authenticated key transfer protocol based on secret sharing scheme that KGC can broadcast group key information to all group members was proposed by Harnand Lin [16]. Here group key recovery is only through authorized group members. Information is theoretically secure due to the confidentiality of this transformation. Group key transportation authentication is provided.

Lim and Lim [17] suggested two group key management schemes for hierarchical self-organizing wireless sensor network architecture, designed so that the forwarding node has more computational and communication burden with a similar load being kept very low with other sensor nodes. This also ensures multilevel security to sensor groups at various levels. Sensor network implements these encryption primitives efficiently without sacrificing strength.

A cluster-based group key management scheme for wireless sensor networks aimed at reducing communication overhead and sensor nodes storage cost was proposed by Zhang, et al., [18]. The procedure includes group key generation through cluster head collaboration with cluster nodes. Cluster heads are responsible to reconstruct and delivery group key. Performance evaluations reveal that the scheme has good security while simultaneously reducing communication overhead when compared to existing schemes like large scale WSN.

Drira, et al., [19] proposed a group key management framework based on a trust oriented clustering scheme. It was demonstrated that trust is a relevant clustering criterion for group key management in MANETs. Trust information enforce authentication and is disseminated by the mobility of nodes. Furthermore, it helps to evict malicious nodes from the multicast session even if they are authorized members of the group. Simulation results show that the solution is efficient and typically adapted to mobility of nodes.

III. Methodology

The Dynamic Source Routing (DSR) is the routing protocol which uses the source routing approach (i.e., every data packet carries the whole path information in its header) to forward packets. Before a source node sends the data packets, it must know the total path going to be taken for transmitting packets to the destination. Otherwise, it will initiate a route discovery phase by flooding a Route Request (RREQ) packet message. DSR is a simple and loop-free protocol. However, it may waste bandwidth if every data packet carries the entire path information along with it. The response time may be large since the source node must wait for a successful RREP if no routing information's to the intended destination are available. Additionally, if the destination is unreachable from the source node due to a network partition, the source node will continue to send RREQ messages, possibly congesting the network [20]. In DSR, the response time may be large if the source node's routing table has no entry to the destination and thus it discovers a path before the message transmission. Advantages of DSR are that it does not use any periodic routing messages (e.g. no router advertisements and no link-level neighbor status messages). Hence, DSR reduces network bandwidth overhead, conserves battery power, and avoids the propagation of potentially large routing updates throughout the ad hoc network [21].

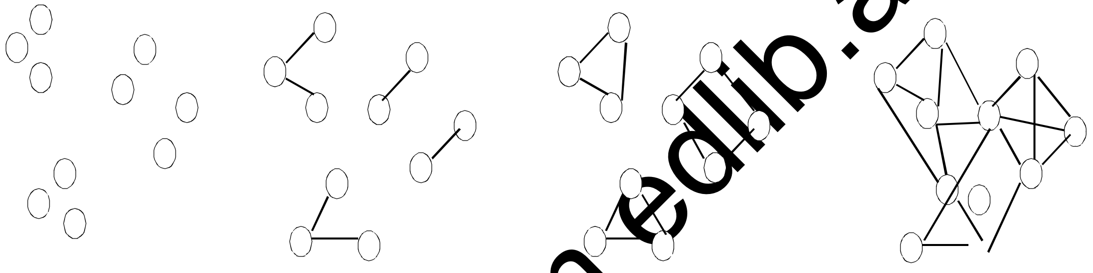
Ad-hoc On-demand Distance Vector (AODV) is a routing protocol which is designed for MANETs and it employs the on-demand routing method to establish the routes between nodes. The main benefit of this protocol is establishment of desired route to destination when the source code requires, and it keeps the routes as long as they are needed. Also, AODV has proper quality to support broadcast, multicast and unicast routing with a scalable characteristic and self-starting. AODV allows mobile nodes for forwarding the packets through their neighbors which may not have a direct communication to the destination until the destination node receives the data packets. This protocol is able to find the shortest and loop free routes to transmit data packets. Also, AODV creates a new route in case of link downs or changes in route [22]. Some advantages of AODV are that the routes are established on demand and destination sequence numbers are used to find the latest route to the destination. Then the connection setup delay is lower. Also, it responds very quickly to the topological changes that affects the active routes. The Time-To-Live (TTL) field in the IP header of the RREQ packets controls the search. If a route to a previously known destination is needed, the prior hop-wise distance is used to optimize the search. This enables computing the TTL value dynamically.

In 1976, Whitfield Diffie and Martin Hellman were influenced by the work of Ralph Merkle on a public key distribution, and proposed an algorithm for key exchange which uses exponentiation in a finite field. Today, Diffie Hellman (DH) algorithm is used in a variety of protocols and services. It is used in interactive transactions, than compared with use in a batch transfer from a sender to a receiver. The algorithm is used when data is encrypted on the Web by using either SSL or TLS and in VPN. Therefore its security is of utmost importance [23].A shared secret is important between two parties who may not have ever communicated previously, so that they can encrypt their communications. As such, it is used by several protocols, including Secure Sockets Layer (SSL), Secure Shell (SSH), and Internet Protocol Security (IPSec). These protocols will be discussed in terms of the technical use of the DH algorithm and the status of the protocol standards established or still being defined. The mathematics behind this algorithm is conceptually simple. The fundamental math includes the algebra of exponents and modulus arithmetic.

IV. Proposed Method

A rekeying process restores the group key after change of each group membership, i.e. join or leave operation. So rekeying may encourage communication overhead during change of frequent group membership. Rekeying mechanism includes property as 1-affects-n scalability which measures how well it scales to large and dynamic groups [19]. To enhance 1-affects-n scalability, some GKM solutions propose to organize the secure group based on logical topology (cluster). Using clusters with different local TEK, the impact of the key updating process (1-affects-n) gets reduces, but needs decryption and re-encryption operations between clusters.

The estimated distance between nodes is graphically represented in figure 1. The cluster head on the formed clusters is selected based on the energy availability.



a) Original data (b) distance=1 (c) distance=2 (d) distance=3

Figure 1. Cluster formation based on distance

In the proposed work, trust is used as the clustering similarity. The cluster formation is adapted from [24]. Also the technique determines the similarity between each pair of clusters named as C_i and C_j with their relative inter-connectivity $RI\ C_i ; C_j$ and their relative closeness $RC\,C_i ;\ C_j$. The hierarchical clustering algorithm selects to merge the pair of clusters where both $RI.\ C_i ;\ C_j$ and $RC\,C_i ;\ C_j$ are high. Through this selection procedure, [24] overcomes the limitations of existing algorithms.

The inter cluster connectivity between a pair of clusters C_i and C_j is defined as the absolute inter cluster connectivity between C_i and C_j is normalized with the internal inter cluster connectivity of the two clusters C_i and C_j. The absolute inter cluster connectivity between a pair of clusters C_i and C_j is defined as the sum of weight of edges that connects vertices in C_i to vertices in C_j. This is the Edge Cut (EC) of the cluster containing two clusters mentioned above. The cluster connectivity of a cluster C_i is captured by the size of its min-cut bisector [26, 27]. Thus the relative inter-connectivity (RI) between a pair of clusters C_i and C_j is given by

$$RI(C_i,C_j)=\frac{|EC_{\{C_i,C_j\}}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}} \qquad (1)$$

which normalizes the absolute inter cluster connectivity with the average internal inter-connectivity of the two clusters. By focusing on the relative inter cluster connectivity between clusters, [25] overcomes the limitations of existing algorithms that use static inter cluster connectivity models. For instance, Figure 1 shows that how the clusters are merged (a) and (b) over clusters (c) and (d), because the relative inter cluster connectivity between clusters (a) and (b) is higher than the relative inter cluster connectivity between clusters (c) and (d), even though the later pair of clusters have a higher absolute inter-connectivity. Hence, the relative inter cluster connectivity is taken into account differences in shapes of the clusters as well as differences in degree of connectivity of different clusters.

The absolute similarity between a pair of clusters is captured in different ways [27]. A drawback of these schemes is that by relying only on a single pair of points, they are less tolerant to outliers and noise. So, the closeness of two clusters is measures by computing the average similarity between the points in C_i that are connected to points in C_j. Since these connections are determined by distance between nodes, their average strength provides a good measure of the affinity between the data items along the interface layer of the two sub-clusters. The internal similarity of each cluster C_i is measured in different ways. The average weights of the edges on the internal bisection of C_i and C_j is smaller than the average weight of all the edges in these clusters. But the average weight of these edges is a better indicator of the internal similarity of these clusters. Hence the relative closeness between a pair of clusters C_i and C_j is computed as,

$$RC(C_i,C_j)=\frac{\bar{S}_{EC_{\{C_i,C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}}+\frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}} \qquad (2)$$

where $\bar{S}_{EC_{C_i}}$ and $\bar{S}_{EC_{C_j}}$ are the average weights of the edges that belong in the min-cut bisector of clusters C_i and C_j ,respectively, and *SEC*f C_i ; C_j gives the average weight of the edges that connect vertices in C_i to vertices in C_j. Also a weighted average of the internal closeness of clusters C_i and C_j is used to normalize the absolute similarity of the two clusters, that favors the absolute similarity of cluster that contains the larger number of vertices.

A. Cluster Head Selection Using Trust

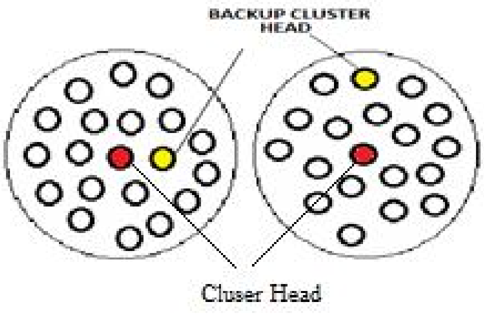


Figure 2. General Architecture of the proposed work

Figure.2. Illustrate the main features and elements of the proposed architecture. Each cluster composed of cluster head, backup node and members of the cluster. The cluster head is the node that identifies the cluster. It is responsible for communication within the members of the cluster and between the clusters. The backup node is responsible for ensuring the redundancy. In case of failure of the cluster head, the backup node will act as the cluster head. Remaining nodes in the cluster are known as the members of the clusters that are not the cluster head and the backup nodes. The cluster head, backup nodes and the members are forming the key agreement zone and generate the group key for cluster.

Trust is one of the basic levels of security. Trust is calculated by each node and the values are stored locally, and regular updating is performed based on new interactions. The trust values are expressed between 0 and 1. 0 indicates a complete mistrust and 1 indicates complete trust. When a new or unknown node y enters the neighbouring hood of node x, the trust agent of node x calculates the trust value of node y.

Direct Trust Direct trust value is evaluated basing on the direct experience that one node may have on another node. Such direct experience can be either full or nil. Full experience increases credential and nil experience decreases credential accordingly. The number of experiences may be unlimited. But the computation trust value is within the range between 0 and 1.

Indirect Trust When node x doesn't have enough direct experience on node y, the node x may enquire to a third node for recommendation.

A cluster head is chosen and it checks the required trust in the network. The algorithm compares the node's trust value by combining direct and indirect trusts to achieve whole trust. Trust value ($T_{theroshold}$) is associated with each job that is processed till all the Cluster Heads (CH) is selected. Trust (T) is then tested against trust sources with direct trust value (D_t), indirect trust value (I_t), and total trust value (T_t). If the T_t is higher than or equal to required trust value then the node is selected as the CH provided none of the two hop nodes that have higher Trust value than the current node. The next highest trust value within the two hop node is named as backup node.

The CH is elected i.e. if a node (X) becomes a cluster head, then check whether it had any earlier experience with its neighbourhood nodes and if so, the direct trust value (D_t) is represented as shown in equation

$$D_t = \mp\sum_{i=1}^{n}\frac{w_i T_{y_i}(x)}{\sum w_i} \tag{3}$$

where, $T_{yi}(x)$ is the sum of its trust value with its two hop neighbors and described later in this section.

If $D_t \geq T_{max}$, then the associated risk is lower than risk threshold and the node (X) becomes CH where there is no node that has higher T value than current node (X). So the indirect trust value (I_t) is represented as in equation

$$I_t = \frac{\sum_{y=1}^{m} T_y(x)}{m} \tag{4}$$

Where $T_y(x)$ trust value of node X based on recommendations from its two hop neighbors.

If $I_t \geq T_{max}$ then associated risk is lower than risk threshold so that node(X) becomes CH provide that there are no neighbour nodes with higher T values. If node (X) value T is lower than T_{max} then total trust value (T_t) is computed as

$$\mathrm{T_t} = D_t * W_A + I_t * W_B \tag{5}$$

where W_A and W_B are weights assigned.

If (T_t) is greater than/equal to ($T_{threshod}$) then the process is continued as above.

In case if all CH is not discovered $T_{threshold}$ is decreased.

Once CH is selected, the trust value certificates can be used by the nodes when it moves to adjacent clusters and this count is used to compute indirect trust. The indirect trust uses communication data rate (R_c) is the rate of successful communication with evaluated nodes with values between 0 and 1 and initial value is 1. The data delivery rate (R_d) is the rate of successful packet delivery by the evaluated node. The indirect trust is the weighted sum of Trust value certificate and communication data rate.

The CH and the backup node are termed the “control set”. The CH, backup node and all the members of the cluster are generating the TEK agreement using A-GDH.2 from the clique’s protocol [22]. It is based on Diffie-Hellman (DH) [23] key agreement method that is responsible for key authentication. The backup node is responsible to maintain the redundant details of CH and it will be the CH if CH is left from the cluster. The pseudo code of A-GDH.2 protocol algorithm is shown below.

```
Let $M = \{M_1, .., M_n\}$ be set of users wanting to share key $S_n$.
$A - GDH.2$ executes n rounds
Initialize:
Let p be a prime and q a prime divisor of p-1
Let G be unique cyclic subgroup of $\mathbf{Z}_p^*$ of order q
Let $\alpha$ be a generator of G
Round i $(0 < i < n)$
1. $M_i$ selects $r_i \in R\mathbf{Z}_p^*$
2. $M_i \rightarrow M_{i+1} : \left\{ \alpha^{\frac{r_1 \ldots r_i}{r_j}} \middle| j \in [1, i] \right\}, \alpha^{r_1 .. r_i}$
Round n
1. $M_n$ selects $r_n \in R\mathbf{Z}_p^*$
2. $M_n \rightarrow ALL\ M_i : \left\{ \alpha^{\frac{r_1 \ldots r_i}{r_j}.K_{in}} \middle| i \in [1, n] \right\}.$
Upon receipt of the above, every $M_i$ computes:
$\alpha^{\left(\frac{r_1 \ldots r_i}{r_j}.K_{in}\right).K_{in}^{-1}.r_i} = \alpha^{r_1 .. r_n} = S_n$
```

Figure 3. A-GDH.2 Protocol

The concept of number of Data Transfer Communication (DTC) is represented as:

$$DTC_{m,n} = \frac{\sum_{t=0}^{T} n_{m,n}}{T} \qquad (6)$$

Where T is the time period, m and n is the nodes through which data are transferred. If two nodes enter each other’s wireless transmission range then $n_{m,n}$ is 1 else 0.

Number of Successful Delivery (SD) can be represented as:

$$SD_{m,n} = \sum_{t=0}^{T} S_{m,n} + S_{n,m} \qquad (7)$$

Duration within Communication Range (CR) can be represented as:

$$CR_{m,n} = \frac{\sum_{t=0}^{T} \mathit{No\ of}\ \text{broad cast ack recieved}}{\mathit{Total}\ \text{number of broadcasts}} \qquad (8)$$

The direct trust can be calculated as:

$$\mathit{Direct}\ \text{Trust} = \frac{\alpha_1 DTC_{m,n} + \alpha_2 SD_{m,n} + \alpha_3 CR_{m,n}}{\sum_{k=1}^{3} \alpha_k} \qquad (9)$$

V. Results and Discussion

Simulations were run using 150 nodes over an area of 1500 sq m. Experiments were conducted for different computed trust and mobility with DSR as the underlying routing protocol. The impact of Diffie Hellman (DH) and GDH for key management was studied. The number of clusters formed, route discovery time, end to end delay and packet delivery ratio respectively was measured.

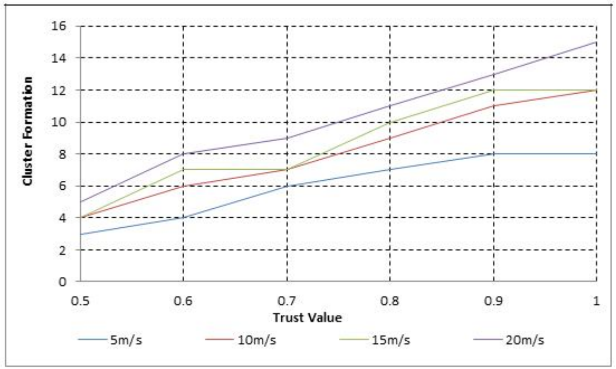


Figure 4. Formation of Number of Clusters

From figure 4 it can be seen that higher trust values increase the number of clusters formed and thus provides better intra cluster communication with very low energy cost. At trust value of 0.7 the inter cluster connectivity and the intra cluster connectivity is balanced for all node motilities.

The cluster head formation over time shows improvement and stability of the proposed technique compared to [19].

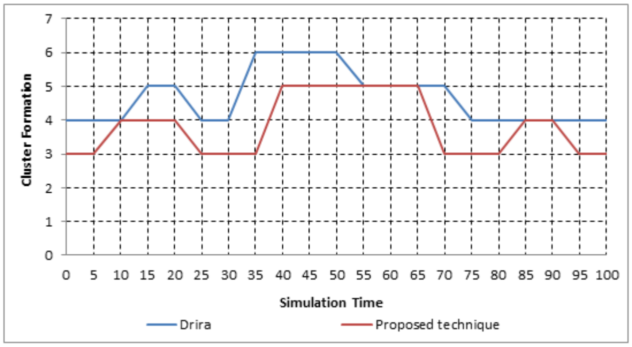


Figure 5. Cluster Formations over Time

In [19], the security was enhanced by the clustering criterion that monitors the trust relations continuously and detects the malicious nodes. Two steps discussed to enhance the efficiency of cluster method and to have accurate trust values are

a) Special traffic and interactions were generated to measure trustworthiness of neighbors and
b) Recommendation is sent to initiate the trust vales for the unknown neighbors.

Proposed method reduces the time for cluster formation when compared to method proposed in [27]. Table 1 shows the average route discovery time and End to End Delay in seconds for different techniques.

Table 1 Route discovery time in seconds and end to end delay in seconds

Techniques	Route Discovery Time in Seconds	End to End Delay in Seconds
DSR	0.962	0.0104
AODV	1.04	0.011
Trust model proposed by Drira et al.,	1.114	0.00928
Proposed DSR with GDH	1.06	0.00817
Proposed DSR with DH Key Management	1.03	0.00836

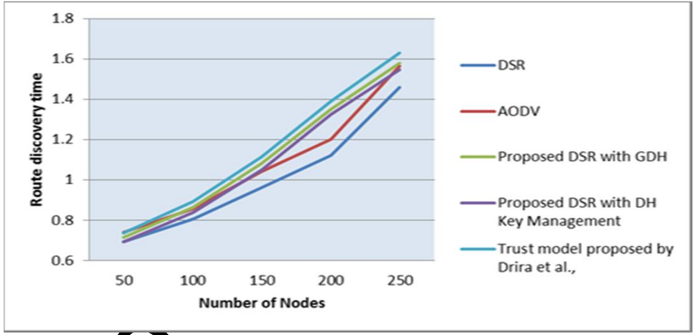


Figure 6. Route Discovery Time

Results show that route discovery time of proposed DSR with Diffie-Hellman (DH) key management is increased as 7.07% than DSR, but reduced as 0.96% than AODV, as7.54% than trust model proposed by Drira and as2.83% than proposed DSR with GDH. From table 1 it is observed that the End to End Delay is achieved by comparing with different methods. Results show that End to End Delay of proposed DSR with DH key management is decreased as 19.62% than DSR, as 24% than AODV, as 9.91% than trust model proposed by Drira but increased as 2.33% than proposed DSR with GDH.

Table 2 Packet delivery ratios

DSR	0 .904
AODV	0 .86
Trust Model Proposed by Drira et al.,	0 .914
Proposed DSR with GDH	0

	.932
Proposed DSR with DH Key Management	0 .917

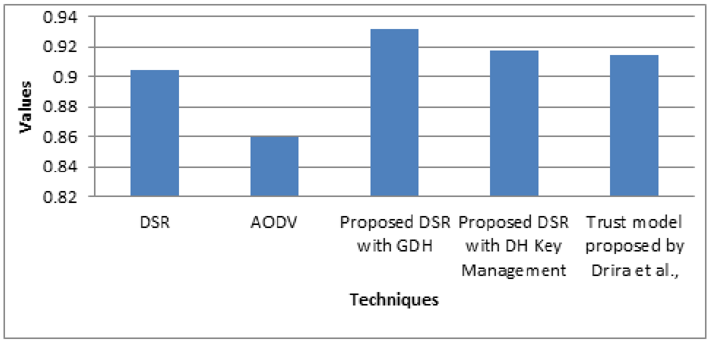


Figure 7. Packet Delivery Ratios

From table 2 and figure 7 it is observed that the Packet Delivery Ratio is achieved by comparing with different methods. Results show that Packet Delivery Ratio of proposed DSR with DH key management is increased with1.44% when compared to DSR, with6.63% than AODV, with0.33% than trust model proposed by Drira but decreased as 1.61% than proposed DSR with GDH.

VI. Conclusion

Key management is crucial for MANET security. In MANET, all networking functions including routing and data transmission are done by the nodes without the need for a central point to control. In a secure group communication, key management techniques are used to provide a correct distribution and easy maintenance of cryptographic keys. This study investigates network performance degradation due to such attacks when trust is used. Trust based clusters are formed based on intermediate nodes trust values. A control group generating the group key is proposed as a new technique in group key management. This includes construction of a group with total users N being divided into many clusters. Secure key management is performed by malicious nodes being avoided due to cluster heads exchanging keys based on trust. Simulation shows the effectiveness of the proposed routing. End to end delay is considerably reduced and packet delivery ratio increased with the proposed method. It is also observed that the performance of proposed routing is considerably better in larger networks.

References

1. Raj, P. N., & Swadas, P. B. (2009). Dpraodv: A dyanamic learning system against blackhole attack in aodv based manet. *arXiv preprint arXiv:0909.2371*.
2. Ramesh, P. S. T. (2013). Secure Routing using Reverse Method and Malicious Node Detection using Global Trust Scheme in MANET.
3. Singh, S. P., & Maini, R. (2011). Comparison of data encryption algorithms. *International Journal of Computer Science and Communication*, *2*(1), 125-127.
4. Elminaam, D. S. A., Abdual-Kader, H. M., & Hadhoud, M. M. (2010). Evaluating The Performance of Symmetric Encryption Algorithms. *IJ Network Security*, *10*(3), 216-222.
5. Yavuz, A. A., AlagOz, F., & Anarim, E. (2010). A new multi-tier adaptive military MANET security protocol using hybrid cryptography and signcryption. *Turkish Journal of Electrical Engineering & Computer Sciences*, *18*(1), 1-21.
6. Kahya, N., Ghoualmi, N., & Lafourcade, P. (2012). Secure Key Management Protocol in WIMAX. *International Journal*, *4*.

7. Chiu, Y. P., Huang, C. Y., & Lei, C. L. (2012). SEMPRE: Secure Multicast Architecture Using Proxy Re-Encryption. *International Journal Of Innovative Computing Information And Control*, *8*(7 A), 4719-4748.
8. Srivastava, A. K., Sharma, S., &Sahu, S. (2012). Msmet: A Modified & Secure Multilanguage Encryption Technique. International Journal on Computer Science and Engineering, 4(3).
9. Bouassida, M. S., Chrisment, I., &Festor, O. (2008). Group Key Management in MANETs. IJ Network Security, 6(1), 67-79
10. Chen, Y. R., Tygar, J. D., &Tzeng, W. G. (2011, April). Secure group key management using uni-directional proxy re-encryption schemes. In *INFOCOM, 2011 Proceedings IEEE* (pp. 1952-1960). IEEE.
11. Rahman, M., Sampalli, S., &Hussain, S. (2010, December). A robust pair-wise and group key management protocol for wireless sensor network. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* (pp. 1528-1532). IEEE.
12. Gomathi, K., & Parvathavarthini, B. (2010, December). An efficient cluster based key management scheme for MANET with authentication. In *Trendz in Information Sciences & Computing (TISC), 2010* (pp. 202-205). IEEE.
13. John, S. P., & Samuel, P. (2010, October). A distributed hierarchical key management scheme for mobile ad hoc networks. In *Information Networking and Automation (ICINA), 2010 International Conference on* (Vol. 1, pp. V1-308). IEEE.
14. Niu, Q. (2009, October). A Trust-Based Message Encryption Scheme for Mobile Ad Hoc Networks. In *Computer Science and Engineering, 2009. WCSE'09. Second International Workshop on* (Vol. 1, pp. 172-176). IEEE.
15. Wu, B., Wu, J., & Dong, Y. (2009). An efficient group key management scheme for mobile ad hoc networks. International Journal of Security and Networks, 4(1), 125-134.
16. Harn, L., & Lin, C. (2010). Authenticated group key transfer protocol based on secret sharing. Computers, IEEE Transactions on, 59(6), 842-846.
17. Lim, S. Y., & Lim, M. H. (2011). Energy-Efficient and Scalable Group Key Management for Hierarchical Sensor Network. Journal of Ubiquitous Systems & Pervasive Networks, 2(1), 39-47.
18. Zhang, Y., Shen, Y., & Lee, S. (2010, April). A cluster-based group key management scheme for wireless sensor networks. In Web Conference (APWEB), 2010 12th International Asia-Pacific (pp. 386-388). IEEE.
19. Drira, K., Seba, H., & Kheddouci, H. (2010). ECGK: An efficient clustering scheme for group key management in MANETs. Computer Communications, 33(9), 1094-1107.
20. Fotino, M., Gozzi, A., De Rango, F., Marano, S., Cano, J. C., Calafate, C., & Manzoni, P. (2007, July). Evaluating Energy-aware behaviour of proactive and reactive routing protocols for mobile ad hoc networks. In 10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'07) (pp. 16-18).
21. Tamilarasi, M., Palanivelu, T. G., Rajan, B., & Das, S. K. (2005). Node Optimization in MANETs for Maximum Throughput Using On-Demand Routing Protocols. In Proceedings of the Eleventh National Conference on Communications: NCC-2005, 28-30 January, 2005 (p. 66). Allied Publishers.
22. Carts, D. A. (2001). A review of the Diffie-Hellman algorithm and its use in secure internet protocols. SANS institute, 1-7.
23. Pereira, O., & Quisquater, J. J. (2002). Security analysis of the cliques protocols suites: first results. In Trusted Information (pp. 151-166). Springer US.\Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68-75.
24. Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68-75.
25. Karypis, G., & Kumar, V. (1995). Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0.
26. Karypis, G., & Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, *20*(1), 359-392.
27. Guha, S., Rastogi, R., & Shim, K. (1998, June). CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record* (Vol. 27, No. 2, pp. 73-84).