Research Journal of Applied Sciences, Engineering and Technology 12(3): 320-327, 2016
DOI: 10.19026/rjaset.12.2339
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: August 8, 2015 Accepted: September 3, 2015 Published: February 05, 2016

Research Article

Optimized Minimum Spanning Tree for Secure Routing in MANET

[1]V. Bhuvaneswari and [2]M. Chandrasekaran
[1]Department of Computer Science and Engineering, Government Polytechnic College, Dharmapuri,
[2]Department of Electronics and Communication Engineering, Government College of Engineering, Bargur, India

Abstract: Secure group communication transfers message from one member to another confidentially. Key management for secure communication in wireless networks is primitive based on cryptographic techniques. Inter-cluster routing was used to improve wireless networks security. In the new scheme, Minimum Spanning Tree (MST) and GBest-BAT algorithm are computed to identify Cluster Heads (CH) with a concept of backup nodes being introduced for effective key management. This study proposes MST formation for inter-cluster routing and it is optimized with GBEST- BAT algorithm.

Keywords: Cluster Head (CH), GBest-BAT algorithm, inter-cluster routing, Minimum Spanning Tree (MST), secure group communication, security

INTRODUCTION

Mobile Ad-hoc Networks (MANET) (Dalal *et al.*, 2012a) are structure less, dynamic networks of mobile nodes without physical links. A MANET has many mobile wireless nodes and communication is carried out without any centralized control. MANET is a self-organized, self-configurable network sans infrastructure, where nodes move arbitrarily. MANET nodes are mobile due to which topology changes dynamically (Singh and Rathore, 2013). MANET is at risk to security attacks due to its topology. So, a secure key management scheme is a prime need for MANETs.

Security is critical for ad hoc networks and it is a largely unexplored area. As nodes use open, shared radio medium in an insecure environment, they are prone to malicious attacks like Denial of Service (DoS). Lack of centralized network management or certification authority ensures that dynamically changing wireless structure are vulnerable to infiltration, eavesdropping and interference. Security is considered to be major “roadblock” in ad hoc network technology’s commercial applications (Jain *et al.*, 2005). Conventional data protection methods with cryptography face the task of key distribution and refreshing. Accordingly, research on security concentrated on secure data forwarding. But, security risks are related to ad hoc networks peculiar features, the most serious being the risk of a node being seized and compromised. This node would have access to the network’s structural information, relayed data and it can send false routing information, which can paralyze the network quickly. A current approach to security issues is building a self-organized public-key infrastructure for adhoc networks cryptography. Key exchange raises scalability issues.

MANET security requirements are (Djenouri *et al.*, 2005):

Availability: Ensuring that desired network services are available when expected, despite attacks. Systems that ensure availability combat DoS and energy starvation attacks to be seen later.

Authenticity: Ensuring genuine inter-node communication. It ensures that a malicious node cannot act as a trusted node.

Data confidentiality: A core security primitive for ad hoc networks, ensuring that a given message can be understood by the recipient(s) only. Data confidentiality is enabled through cryptography.

Integrity: Denotes data authenticity when sent from one node to another i.e., ensuring that a message from node A to node B is not modified by malicious node, C, in transmission.

Non-repudiation: Ensures that message origin is legitimate i.e., when a node receives a false message non-repudiation allows former to accuse latter of

Corresponding Author: V. Bhuvaneswari, Department of Computer Science and Engineering, Government Polytechnic College, Dharmapuri, India

sending false message helping other nodes to learn about it.

Key management manages cryptographic keys in a cryptosystem including handling generation, storage, exchange, use and replacement of keys. It also incorporates cryptographic protocol design, key servers, user procedures and relevant protocols (Singh and Rathore, 2013). In MANETs, key management is classified into 2 kinds; the first is based on a centralized/distributed Trusted Third Party (TTP) responsible for renewing, issuing, revoking and providing keying material to nodes participating in situations where key management process is done with threshold cryptography (Rafsanjani and Shojaiemehr, 2012). The second key management type are self-organized key management schemes that allow nodes to generate own keying material, issue public-key certificates to other network nodes based on their knowledge. Nodes store and distribute certificates.

MANETs key management schemes are classified Symmetric Key Cryptography, Asymmetric Key Cryptography and Group Key Management (Xiong and Gong, 2011). Symmetric Key Cryptography applied to MANETs are based on keys deployed in advance, including a single key used by nodes. A node shares a single key with single/multi-nodes. A deployed node possesses the following key. Such schemes are divided into determinate key management scheme and stochastic key management scheme. MANETs secure communication demands Group Key Management. The donated group key security protocol solves problems like group key update regularity, group key update, group key produce, when a node member joins or leaves. Group key management schemes are divided into Centralized Group Key Agreement Protocols (CGKAP), Distributed Group Key Agreement Protocols (DiGKAP) and Decentralized Group Key Management Protocols with relaying (DeGKMP). Asymmetric Key Cryptography schemes are based on Certificate Based Cryptography (CBC), where public key certificates authenticate public key. The CBC-based scheme needs certificate based public-key generation and distribution. Such schemes do not fit MANETs as they cause unfavorable communication latency and huge communication overheads.

To ensure MANET security, different Key Management schemes are used. Using and managing keys for security is crucial in MANETs due to its energy constrained operations, limited security, variable capacity links and dynamic topology. MANET speed depend on applications, for example, in commercial applications (short range networks) speed is high whereas in military applications (long range network) speed is low, i.e., speed is inversely prepositional to network range. MANET have special features like network working in a standalone intranet and also can be connected to a large internet. It can cover an area bigger than a transmission range and is quickly deployable due to using internal routing (Dalal *et al*., 2012b).

Clustering divides a network into different virtual groups, based on rules to discriminate nodes allocated to different sub-networks. The goal is to achieve scalability in large networks and high mobility (Anupama and Sathyanarayana, 2011). Cluster-based routing solves node heterogeneity and limits routing information propagating inside a network. It increases routes life and decreases routing control overhead (Bakht, 2011; Narayanan *et al*., 2013). There are 3 types of nodes: cluster heads, cluster members and cluster gateways. Cluster Heads (CHs) coordinate nodes in their clusters (intra-cluster communication) and also communicate with other cluster heads (inter-cluster communication). Cluster Members (CMs) are ordinary nodes that transmit information to their cluster heads, which aggregate received information and forward it to a sink. Cluster gateways are non-cluster heads having inter-cluster links to contact neighboring clusters to forward information.

Tree-Based Multi-Channel Protocol (TMCP) is a greedy, tree-based, multi-channel data collection applications protocol which partitions a network into multiple sub-trees reducing intra-tree interference by assigning different channels to nodes on different branches starting from top to the bottom of the tree scheduled according, to TMCP for aggregated data collection. Here, nodes on left most branch are assigned frequency F1, second branch frequency F2 and last branch frequency F3. After channel assignments, time slots are assigned to nodes with the BFS Time Slot Assignment algorithm. TMCP's advantage is that it is designed to support convergence of cast traffic and needs no channel switching. But, contention inside branches is not resolved as all nodes on same branch communicate on the same channel. This study proposed GBest with BAT to optimize inter-cluster routing using Minimum Spanning Tree (MST).

LITERATURE REVIEW

Noisy versions of Minimum Spanning Tree (MST) problem was investigated by Gronskiy and Buhmann (2014) who compared MST algorithms generalization properties. An information-theoretic analysis of MST algorithms measures information on spanning trees extracted from an input graph. Early stopping of MST algorithm yields approximate spanning trees set with increased stability compared to minimum spanning tree. The framework provides insights for algorithm design when noise is unavoidable in combinatorial optimization.

A Modified Shuffled Frog-Leaping Algorithm (MSFLA) with Genetic Algorithm (GA) cross-over to solve MST problem was proposed by Roy (2011). SFLA is a natural memetics inspired meta-heuristic search method combining benefits of meme-based Memetic Algorithm (MA) and social behavior based Particle Swarm Optimization (PSO). SFLA was modified for MST problem. Results reveal that the algorithm ensures accurate results with minimum iterations.

The selection process of Scattered settlements composed of individual buildings seen as point cluster was performed by Zheng *et al*. (2011). The selection was performed with properties like selectable, disposable and selectable-or-disposable. The point cluster selection was transformed into a simplification of the linear cluster, with Ant Colony Optimization (ACO) algorithm being applied to simplify linear objects. The experiment showed that the new method ensured feasible and effective results.

Bees algorithm based approach to handle degree constrained problem was proposed by Malik (2012). Travel Salesman Problem (TSP) was considered and a set of 2-degree spanning trees extracted from a graph and supplied to the new algorithm. A bees algorithm based approach optimized spanning trees based on cost values. Fitness function points that cost effective degree constrained spanning trees. Experiments with TSP show that the new approach produces cost/time effective results.

An MST-based and new GA algorithm for distribution network optimal planning was presented by Li and Chang (2011). Two new operators were introduced to reduce computational time and avoid infeasible solution and to ensure that individuals are feasible solutions. An electricity distribution network and feeder cross-sectional area selection simultaneous optimization model dealt with the weight of minimal-cost system tree. This combinatorial coding guarantees solution validity to a global optimum.

Minimum Energy Network Connectivity (MENC) problem, that reduces sensors transmission power in wireless networks and lowers its energy consumption while simultaneously keeping global connectivity was addressed by Abreu and Arroyo (2011). MENC problem is NP-hard and its hardness motivates the development of a PSO based heuristic algorithm to get near-optimal solutions. The new heuristic was tested on a 50 instances problem set. Computational results show that the new approach performs better than classical MST heuristic.

An improved Discrete PSO (DPSO) approach for mcd-MST that compromises between key objectives in WSNs like energy consumption, reliability and QoS provisioning was presented by Guo *et al*. (2009). GA's mutation and crossover operator principles were incorporated in the new PSO algorithm to achieve better diversity and break from local optima. The new algorithm was compared to an enumeration method. The simulation shows that the new algorithm provides efficient/high-quality solutions for mcd-MST.

A study on PSO applying an instance of Multi-Level Capacitated Minimum Spanning Tree Problem was presented by Papagianni *et al*. (2009). A diversity preservation global variant of PSO meta-heuristic was specifically presented. The specific PSO variant includes Gaussian mutation to avoid premature convergence and alternative selection of flight guide per particle. Results were compared to corresponding evolutionary approaches. Network Random Keys decoded/encoded Potential tree solutions.

An ant-based algorithm to find low cost Degree-Constrained Spanning Trees (DCST) presented by Bui *et al*. (2012) uses a set of ants which traverse the graph and identify candidate edges from which DCST was constructed. Local optimization algorithms improved DCST. Experiments using 612 problem instances show improvements over current algorithms.

METHODOLOGY

BAT optimization algorithm and hybrid GBEST-BAT are explained in detail in the following sections.

A Minimum Spanning Tree (MST): Minimum Spanning Tree (MST) (Upadhyayula and Gupta, 2006) is a sub-graph that spans over vertices of a graph without any cycle and with minimum sum of weights over all edges. Weight for every edge is considered in MST-based clustering as the Euclidean distance between end points forming that edge. So, an edge that connects 2 sub-trees in MST must be shortest. In such clustering, unusually longer inconsistent edges are removed from MST. MT's connected components obtained by removing edges are treated as clusters. Elimination of longest edge results in 2-group clustering. Removal of next longest edge leads to 3-group clustering.

A packet is transmitted by a node that does not exist after one hop. To spend least energy in packet transmission, a node transmits to its closest (weight) neighbor (towards sink node). Energy consumed is given by Eq. (1):

$$E_{total} = K \times \sum_{\forall(u,v)\in T} w(u,v) \qquad (1)$$

where, K is a constant packet traveling along a graph, $w(u, v)$ is the weight of the link between nodes u and v and T is a tree. In Eq. (1), E_{total} is minimized only when $\sum_{\forall(u,v)\in T} w(u,v)$ is minimized.

Cluster head selection: Cluster formation is adapted from (Karypis *et al*., 1999). The technique also determines similarity between each cluster pair named C_i and C_j with their relative inter-connectivity *RI*. C_i; C_j/ and their relative closeness *RC*. C_i; C_j/. A hierarchical clustering algorithm merges a pair of clusters where both *RI C$_i$; C$_j$/* and *RC. C$_i$; C$_j$/* are high. By this selection, (Karypis *et al*., 1999) overcomes limitations of current algorithms.

Inter-cluster connectivity between a pair of clusters C_i and C_j is defined as absolute inter-cluster connectivity between C_i and C_j and is normalized with internal inter-cluster connectivity of 2 clusters C_i and C_j. Absolute inter-cluster connectivity between a pair of clusters C_i and C_j is defined as the sum of the weight of edges connecting vertices in C_i to vertices in C_j. This is Edge Cut (EC) of the cluster having the 2 clusters mentioned above. Cluster connectivity of cluster C_i is captured by the size of its min-cut bisector (Karypis and Kumar, 1995, 1998). Thus Relative Inter-connectivity (RI) between a pair of clusters C_i and C_j is given by Eq. (2):

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

Which normalizes absolute inter-cluster connectivity with average internal inter-connectivity of the two clusters. By focusing on the relative inter-cluster connectivity between clusters, overcomes limitations of present algorithms that use static inter-cluster connectivity models.

Trust for cluster head selection: Trust is a basic level of security. It is calculated by a node and values are stored locally. Regular updating based on new interactions is performed. Trust values expressed between 0 and 1. 0 indicate complete mistrust and 1 indicates complete trust. When a new or unknown node y enters the neighbourhood of node x, trust agent of node x calculates trust value of node y.

A chosen cluster head checks required network trust. The algorithm compares the node's trust value by combining direct/indirect trusts to achieve total trust. Trust value ($T_{\text{threshhold}}$) is associated with a job processed till all Cluster Heads (CH) are chosen. Trust (T) is tested against trust sources with direct trust value (D_t), indirect trust value (I_t) and total trust value (T_t). When T_t is higher than or equal to required trust value, then a node is selected as CH provided no 2 hop nodes have a higher trust value than the current node. The next highest trust value in a 2 hop node is named backup node.

CH is elected i.e., when a node (X) becomes a cluster head, then checks whether it had earlier experience with neighborhood nodes. If so, direct trust value (D_t) is represented as in Eq. (3):

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

where, $T_{yi}(x)$ is the sum of its trust value with its 2 hop neighbors.

If $D_t \geq T_{\max}$, then associated risk is lower than risk threshold and node (X) becomes CH where there is no node with higher T value than current node (X). So indirect trust value (I_t) is represented as in Eq. (4):

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

where, $T_y(x)$ trust value of node X based on recommendations from its 2 hop neighbors.

If $I_t \geq T_{\max}$ then associated risk is lower than risk threshold so that node (X) becomes CH provided there are no neighbor nodes with higher T values. If node (X) value T is lower than $T_{\max}$ then total trust value (T_t) is computed as in Eq. (5):

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

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.

If CH is not discovered $T_{\text{threshold}}$ is decreased.

When CH is selected, trust value certificates are used by nodes when moving to adjacent clusters. This count computes indirect trust. The indirect trust uses communication data rate (R_c) which is a rate of successful communication with evaluated nodes with values between 0 and 1 and whose initial value is 1. Data delivery rate (R_d) is the rate of successful packet delivery by evaluated node. Indirect trust is a weighted sum of Trust value certificate and communication data rate.

CH and the backup node are termed "control set". CH, backup node and all cluster members generate TEK agreement using A-GDH.2 from cliques protocol (Gomathi and Parvathavarthini, 2010) based on Diffie-Hellman (DH) (Zhang *et al*., 2010) key agreement method responsible for key authentication. A Backup node maintains the CH's redundant details and it becomes CH when the real CH leaves the cluster.

Proposed Gbest BAT algorithm: Yang (2010) proposed Bat Algorithm was inspired by bats echolocation characteristic. Echolocation is sonar which bats use to detect prey and avoid obstacles. Bats emit a very loud sound and listen for an echo to bounce back from objects. Thus, a bat computes how far it is from an object. Also, bats distinguish the difference between obstacle and prey in total darkness (Nakamura *et al*., 2012). To transform such bat behavior to an algorithm, Yang idealized some rules (Komarasamy and Wahi, 2012):

- All bats use echolocation to sense distance and to know difference between food and background barriers; Bats fly randomly with velocity v_i at position x_i with a frequency f_{min}, varying wavelength and loudness A_0 to search for prey. They automatically adjust wavelength (frequency) of emitted pulses and adjust pulse emission rate [0, 1], based on the target's proximity.
- Though loudness varies in many ways, it is assumed that the variance is from a large (positive) A_0 to a minimum constant value $A_{\min}$.

Initialization of bat population: Random generation of initial population is done from real-valued vectors with dimension d and number of bats n, by considering lower and upper boundaries as in Eq. (6):

$$x_{ij} = x_{\min j} + rand(0,1)(x_{\max j} - x_{\min j}) \tag{6}$$

where, i = 1, 2,…n, j = 1, 2,….d, $x_{\min j}$ and $x_{\max j}$ are lower and upper boundaries for dimension j respectively.

Update process of frequency, velocity and solution: A frequency factor controls solution step size in BA. This factor is assigned a random value for every bat (solution) between upper and lower boundaries [$f_{\min}$, $f_{\max}$]. Solution velocity is proportional to frequency and a new solution depends on new velocity (in Eq. (7)):

$$\begin{aligned} f_i &= f_{\min} + (f_{\max} - f_{\min})\beta \\ v_i^t &= v_i^{t-1} + (\mathrm{x}_i^t - x^*) f_i \\ x_i^t &= x_i^{t-1} + v_i^t \end{aligned} \tag{7}$$

where, β∈[0, 1] indicates randomly generated a number, x* represents current global best solutions.

Update process of loudness and pulse emission rate: Loudness/pulse emission rate must be updated as iterations proceed. As a bat (Yilmaz and Kucuksille, 2013) gets closer to prey then loudness A decreases and pulse emission rate increases. Loudness A and pulse emission rate r are updated by Eq. (8):

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t \\ r_i^{t+1} &= r_i^0[1 - e^{(-\gamma t)}] \end{aligned} \tag{8}$$

where, α and γ are constants. r^0_i and A_i are factors consisting of random values and A^0_t can be [1, 2], while r^0_i can typically be [0,1].

Initially, all bats fly randomly in search space producing random pulses. After each fly, each bat's position is updated as in Eq. (9) (Baziar *et al*., 2013):

$$\begin{aligned} V_i^{new} &= V_i^{old} + f_i(Gbest - X_i); i = 1,..., \mathrm{N}_{Bat} \\ X_i^{new} &= X_i^{old} + V_i^{new}; i = 1,..., \mathrm{N}_{Bat} \\ f_i &= f_i^{\min} + \varphi_1(f_i^{\max} - f_i^{\min}); i = 1,..., \mathrm{N}_{Bat} \end{aligned} \tag{9}$$

where, Gbest is best bat from an objective function point of view; NBat is number of bats in a population; to reach a better random walking, another random fly is simulated where a random number β is generated randomly. In each iteration, if random value β is larger than ri then a new solution around Xi is generated as in Eq. (10):

$$X_i^{new} = X_i^{old} + \varepsilon A_{mean}^{old}; i = 1,..., N_{Bat} \tag{10}$$

where, ε is a random value in a range of [−1, 1] and A_{mean}^{old} is the mean value of all bats loudness. If random value β is less than ri then a new position $\mathrm{A_i^{new}}$ is generated randomly. New position $\mathrm{A_i^{new}}$ is accepted when Eq. (11) is satisfied:

$$[\beta < A_i]\&[f(X_i) < f(\mathrm{Gbest})] \tag{11}$$

Also, values of loudness and rate are updated as in Eq. (12):

$$\begin{aligned} A_i^{new} &= \alpha A_i^{old} \\ r_i^{Iter+1} &= r_i^0[1 - \exp(-\gamma \times Iter)] \end{aligned} \tag{12}$$

where, α and γ are constant values and Iter is number of the iterations during optimization.

RESULTS AND DISCUSSION

Table 1 to 4 and Fig. 1 to 4 shows the result values and graph for average packet delivery ratio, average end to end delay, average number of hops and jitter respectively.

Table 1 and Fig. 1 shows that the average packet delivery ratio for Trust Cluster GBEST BAT MST with GDH performs better by 5% than DSR with GDH and by 3.04% than Trust Cluster BAT MST with GDH at number of nodes are 75. Similarly, the average packet delivery ratio for Trust Cluster GBEST BAT MST with GDH performs better by 15.9% than DSR with GDH and by 9.92% than Trust Cluster BAT MST with GDH at number of nodes are 450.

Table 2 and Fig. 2 shows that the average end to end delay for Trust Cluster GBEST BAT MST with GDH performs better by reducing delay as 16.92% than DSR with GDH and by 6.5% than Trust Cluster BAT MST with GDH at number of nodes are 75. Similarly,

Table 1: Average packet delivery ratio

Number of nodes	DSR with GDH	Trust cluster BAT MST with GDH	Trust cluster GBEST BAT MST with GDH
75	0.858	0.875	0.902
150	0.770	0.817	0.829
225	0.784	0.801	0.811
300	0.750	0.767	0.785
375	0.696	0.714	0.735
450	0.582	0.613	0.677

Table 2: Average end to end delay

Number of nodes	DSR with GDH	Trust cluster BAT MST with GDH	Trust cluster GBEST BAT MST with GDH
75	0.001077	0.000970	0.000909
150	0.001276	0.001245	0.001150
225	0.002588	0.002804	0.001279
300	0.004293	0.003975	0.001426
375	0.012270	0.013116	0.008401
450	0.087699	0.085501	0.012729

Table 3: Average number of hops to destination

Number of nodes	DSR with GDH	Trust cluster BAT MST with GDH	Trust cluster GBEST BAT MST with GDH
75	3.79	3.41	3.67
150	4.92	4.57	4.62
225	4.67	5.37	5.49
300	5.91	5.75	5.35
375	6.44	5.72	5.82
450	6.5	5.83	6.04

Table 4: Jitter

Number of nodes	DSR with GDH	Trust cluster BAT MST with GDH	Trust cluster GBEST BAT MST with GDH
75	0.000249	0.000249	0.000232
150	0.000566	0.000581	0.000534
225	0.000752	0.000823	0.000723
300	0.000787	0.000775	0.000708
375	0.001103	0.001061	0.000974
450	0.001178	0.001155	0.001023

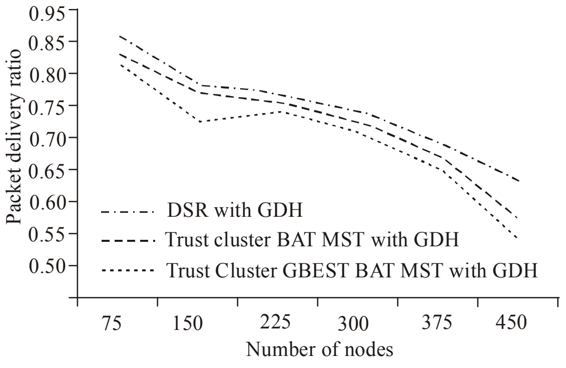

Fig. 1: Average packet delivery ratio

the average end to end delay for Trust Cluster GBEST BAT MST with GDH performs better by 149.3% than DSR with GDH and by 148.2% than Trust Cluster BAT MST with GDH at number of nodes are 450.

Table 3 and Fig. 3 shows that the average number of hops to destination for Trust Cluster GBEST BAT MST with GDH performs better by 3.22% than DSR with GDH and by 7.34% than Trust Cluster BAT MST with GDH at number of nodes are 75. Similarly, the average number of hops to destination for Trust Cluster GBEST BAT MST with GDH performs better by 7.34% than DSR with GDH and by 3.54% than Trust Cluster BAT MST with GDH at number of nodes are 450.

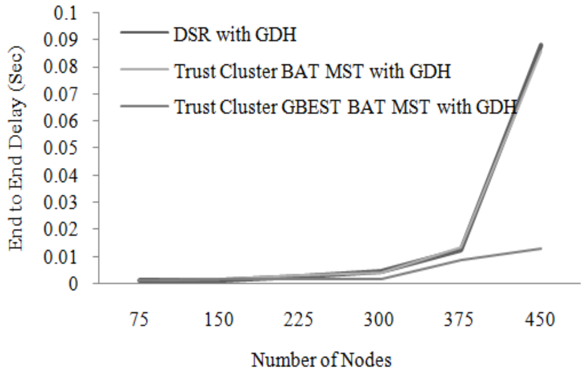

Fig. 2: Average end to end delay

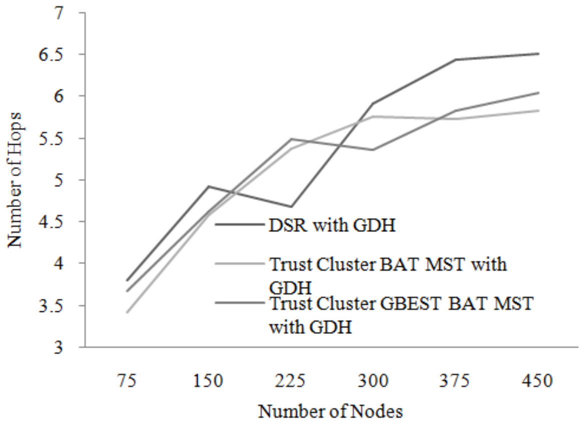

Fig. 3: Average number of hops to destination

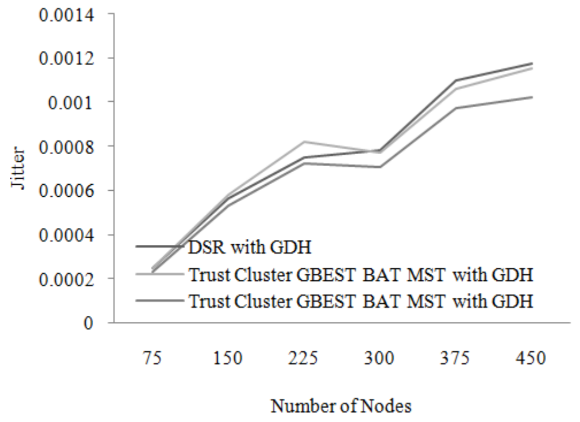

Fig. 4: Jitter

Table 4 and Fig. 4 shows that the jitter for Trust Cluster GBEST BAT MST with GDH performs better by 7.07% than DSR with GDH and by 7.07% than Trust Cluster BAT MST with GDH at number of nodes are 75. Similarly, the jitter for Trust Cluster GBEST BAT MST with GDH performs better by 14.08% than DSR with GDH and by 12.12% than Trust Cluster BAT MST with GDH at number of nodes are 450.

CONCLUSION

MANETs are susceptible to attacks by malicious nodes resulting in packets being dropped. Key management is crucial in MANET security issues as it is the basis for security services. This study uses inter-cluster routing to mitigate network performance degradation due to malicious nodes. Inter-cluster routing is a clustering criterion for MANETs group key management. GBEST-BAT algorithm is disseminated by meta-heuristic population based optimization algorithm inspired from bats search for food. Then node mobility detects malicious group members. Experiments show that the new Trust Cluster GBEST BAT MST with GDH ensures improved average packet delivery ratio, average end to end delay, average hops and jitter than Trust Cluster BAT MST with GDH and DSR with GDH. Average packet delivery ratio for Trust Cluster GBEST BAT MST with GDH is 5% than DSR with GDH and by 3.04% than Trust Cluster BAT MST with GDH when there 75 nodes. Similarly, average packet delivery ratio for Trust Cluster GBEST BAT MST with GDH improves by 15.9% than DSR with GDH and by 9.92% than Trust Cluster BAT MST with GDH when there are 450 nodes. Similarly, average end to end delay for Trust Cluster GBEST BAT MST with GDH reduces delay by 16.92% than DSR with GDH and by 6.5% than Trust Cluster BAT MST with GDH when there are 75 nodes. Average end to end delay for Trust Cluster GBEST BAT MST with GDH improves by 149.3% than DSR with GDH and by 148.2% than Trust Cluster BAT MST with GDH when there are 450 nodes.

REFERENCES

Abreu, R.C. and J.E.C. Arroyo, 2011. A particle swarm optimization algorithm for topology control in wireless sensor networks. Proceeding of the 30th IEEE International Conference of the Chilean Computer Science Society, pp: 8-13.

Anupama, M. and B. Sathyanarayana, 2011. Survey of cluster based routing protocols in mobile ad hoc networks. Int. J. Comput. Theor. Eng., 3(6): 806-815.

Bakht, H., 2011. Survey of routing protocols for mobile ad-hoc network. Int. J. Inform. Commun. Technol. Res., 1(6): 258-270.

Baziar, A., A. Kavoosi-Fard and J. Zare, 2013. A novel self- adaptive modification approach based on bat algorithm for optimal management of renewable MG. J. Intell. Learn. Syst. Appl., 5(01): 11.

Bui, T.N., X. Deng and C.M. Zrncic, 2012. An improved ant-based algorithm for the degree-constrained minimum spanning tree problem. IEEE T. Evolut. Comput., 16(2): 266-278.

Dalal, R., M. Khari and Y. Singh, 2012a. Different ways to achieve trust in MANET. Int. J. AdHoc Netw. Syst., 2(2): 53-64.

Dalal, R., Y. Singh and M. Khari, 2012b. A review on key management schemes in MANET. Int. J. Distrib. Parall. Syst., 3(4): 165-172.

Djenouri, D., L. Khelladi and N. Badache, 2005. A survey of security issues in mobile ad hoc networks. IEEE Commun. Surv., 7(4): 2-28.

Gomathi, K. and B. Parvathavarthini, 2010. An efficient cluster based key management scheme for MANET with authentication. Proceedings of the Trendz in Information Sciences and Computing (TISC, 2010). Chennai, pp: 202-205.

Gronskiy, A. and J.M. Buhmann, 2014. How informative are minimum spanning tree algorithms? Proceeding of the IEEE International Symposium on Information Theory (ISIT, 2014). Honolulu, HI, pp: 2277-2281.

Guo, W.Z., H.L. Gao, G.L. Chen and L. Yu, 2009. Particle swarm optimization for the degree-constrained MST problem in WSN topology control. Proceeding of the IEEE International Conference on Machine Learning and Cybernetics, 3: 1793-1798.

Jain, J., M. Fatima, R. Gupta and K. Bandhopadhyay, 2005. Overview and challenges of routing protocol and mac layer in mobile ad-hoc network. J. Theor. Appl. Inform. Technol., 8(1): 6-12.

Karypis, G. and V. Kumar, 1995. Metis-unstructured graph partitioning and sparse matrix ordering system. Version 2.0, Technical report, Department of Computer Science, University of Minnesota, Minneapolis, MN.

Karypis, G. and V. Kumar, 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput., 20(1): 359-392.

Karypis, G., E.H. Han and V. Kumar, 1999. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8): 68-75.

Komarasamy, G. and A. Wahi, 2012. An optimized K-means clustering technique using bat algorithm. Eur. J. Sci. Res., 84(2): 263-273.

Li, Y. and X. Chang, 2011. A MST-based and new GA supported distribution network planning. Proceeding of the International Conference on Mechatronic Science, Electric Engineering and Computer (MEC, 2011), pp: 2534-2538.

Malik, M., 2012. Bees algorithm for degree-constrained minimum spanning tree problem. Proceeding of the National Conference on Computing and Communication Systems (NCCCS, 2012). Durgapur, pp: 1-8.

Nakamura, R.Y.M., L.A.M. Pereira, K.A. Costa, D. Rodrigues, J.P. Papa and X.S. Yang, 2012. BBA: A binary bat algorithm for feature selection. Proceeding of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), August 22-25, pp: 291-297.

Narayanan, A.E., R. Devi and A.V. Jayakumar, 2013. An energy efficient cluster head selection for fault-tolerant routing in MANET. Int. J. Eng. Technol., 5(2): 0975-4024.

Papagianni, C., C. Pappas, N. Lefkaditis and I.S. Venieris, 2009. Particle swarm optimization for the multi level capacitated minimum spanning tree. Proceeding of the IEEE International Multiconference on Computer Science and Information Technology (IMCSIT), pp: 765-770.

Rafsanjani, M.K. and B. Shojaiemehr, 2012. Improvement of self-organized public key management for MANET. J. Am. Sci., 8(1): 197-202.

Roy, P., 2011. A new technique to solve Minimum Spanning Tree (MST) problem using Modified Shuffled Frog-Leaping Algorithm (MSFLA) with GA cross-over. Proceeding of the 3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom). Bangalore, pp: 32-35.

Singh, U.P. and R.S. Rathore, 2013. Distributed hierarchical group key management using elliptic curve and hash function. Int. J. Comput. Appl., 61(19).

Upadhyayula, S. and S.K.S. Gupta, 2006. Spanning tree based algorithms for latency and energy efficient data aggregation enhanced convergecast (DAC) wireless sensor networks. Ad Hoc Netw., 5(5): 626-648.

Xiong, W.A. and Y.H. Gong, 2011. Secure and highly efficient three level key management scheme for MANET. WSEAS T. Comput., 10(1): 6-15.

Yang, X.S., 2010. A New Metaheuristic Bat-Inspired Algorithm. In: Gonzalez, J.R., D.A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO, 2010). Springer-Verlag, Berlin, Heidelberg.

Yilmaz, S. and E.U. Kucuksille, 2013. Improved Bat Algorithm (IBA) on continuous optimization problems. Lect. Notes Software Eng., 1(3): 279-283.

Zhang, Y., Y. Shen and S. Lee, 2010. A cluster-based group key management scheme for wireless sensor networks. Proceeding of the 12th IEEE International Asia-Pacific Web Conference (APWEB), pp: 386-388.

Zheng, C., H. Hu, Y. Hu and H. Xia, 2011. The selection of scattered settlements based on ant colony optimization algorithm. Proceeding of the 19th International Conference on Geoinformatics, pp: 1-6.