ISSN 2278-3091
Volume 8, No.1, January – February 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse07812019.pdf
https://doi.org/10.30534/ijatcse/2019/07812019

A Secure and Power-Aware Protocol for Wireless Ad Hoc Networks

Pooja Singh[1], Nasib Singh Gill[2]
[1]Research Scholar, DCSA, Maharshi Dayanand University, Rohtak, India, poojasinghpooja77@gmail.com.
[2] Professor, DCSA, Maharshi Dayanand University, Rohtak, India, nasibsgill@gmail.com.

ABSTRACT

Wireless Ad hoc networks are more prone to attacks then the wired network. Cryptographic keys provide valuable ways to intact the security of such networks. Public key cryptography is one of the most applicable ways to manage the security of wireless networks. The mobile wireless ad hoc networks that operate on resource-limited devices require computationally efficient security system. The elliptic curve cryptography (ECC) is public key cryptography based on scalar multiplication and has a small length of keys. Therefore, the ECC has comparatively lowered computational and communication cost as compare to Robin's scheme RSA. In this paper, we propose a security scheme (ECCDGKM) based on ECC accompanied by a trust-oriented weight clustering algorithm (TWCA). The TWCA forms clusters on the bases of computational power, the number of neighbors, the proximity with the neighbors and their trust value. The cluster size is proportional to the weight of cluster head (CH) i.e. the CH with higher weight will manage a larger cluster. The key management workload is proportionally distributed among the CH's according to their weight. The CH's are selected in such a manner that each CH is a neighbor of one or two CHs. This will eliminate the need for gateway nodes. The process is repeated after a threshold time and thus, reduces the frequent drowning of CH. The ECC enabled digital signature scheme is used to generate and share the local group key for intra-cluster communication and global group key for inter-cluster communication. We compare our scheme with two recent and similar schemes. The ECCDGKM shows better results than these schemes in terms of the number of cluster forms, keys generated, operations performed and messages exchanged.

Key words: Cluster, Decentralized group key management, Elliptic curve cryptography, Intra-cluster communication, Inter-cluster communication, Trust-oriented weight clustering algorithm.

1. INTRODUCTION

Wireless ad hoc networks have increased its applicability in various fields. The setup time of these networks is considerably less than the wired one. No need for fixed infrastructure is the prime factor that increases its applicability. Quick network setup, in the battlefield, during the natural disaster rescue operation, while establishing business or personal communication become possible with these networks [1]. The wireless communication medium is highly prone to various active and passive attacks. The wireless medium is an open system and is more vulnerable than a closed or wired medium. Previous research studies have revealed that cryptography is a promising approach to maintain security in the wireless ad hoc network [2].

Cryptography key management techniques describe the method of key generation and its distribution. The key management process can be broadly classified into three categories- centralized, distributed and decentralized key management. The centralized key management system where a single node handles all the key management process is not suitable for wireless ad hoc network as all nodes are peer nodes. So the excess workload on a single node may rapidly exhaust the root node. The distributed key management process is in which the group key is partially distributed among all nodes or a threshold number of nodes. This approach can be applied on wireless ad hoc network but the involvement of a large number of nodes for all key management activities increase the time and number of message exchange. The decentralized key management process is where the activities of the root node are decentralized among the subgroup/cluster heads and is the most appropriate key management approach for wireless ad hoc networks [3].

The Key management is based on crypto-security mechanism. Here also two different approaches of symmetric and asymmetric cryptographic keys provide benefit in different scenarios. The symmetric cryptographic key management has a single key for encryption and decryption. This system requires a secure channel for sharing of keys. The asymmetric cryptographic keys have two keys namely private and public key. These schemes do not require a secure channel for sharing of secure keys. The major drawback of some of its well-known asymmetric cryptographic schemes is the high computational cost and large size of its keys. The asymmetric cryptography is appropriate for wireless ad hoc network because there is no need of secure channel for sharing of group secret key but the large size of keys increase the computational and communication complexity and this is not suitable for battery operated devices. The elliptic curve cryptography ECC is a good choice for wireless ad hoc network. The asymmetric elliptic curve cryptographic key management schemes provide the same secrecy level but with a much smaller key size. The most popular and accepted asymmetric cryptography public key infrastructure RSA

provide an acceptable level of security with 1024 bit key size, whereas ECC gives the same secrecy level with 160-bit key size.

In this paper, we proposed a secure and power-aware protocol based on an elliptic curve cryptographic enable security scheme and a trust-oriented weight clustering algorithm for decentralized group key management. The paper is further divided into 4 sections. Section 2 summarized a few popular and recent research paper related to decentralized group key management, asymmetric cryptography and clustering. Section 3 elaborates our proposed key management scheme. In section 4, the experimental results are explained and in section 5, we conclude our findings.

2. RELATED WORK

Group Key Management Protocol using huddle hierarchy and a gray code is proposed by R. Varalakshmi and V. R. Uthariaraj[4]. This protocol is designed to reduce the storage overhead of both the group head and group member by using huddle hierarchy and gray code. This gray code is used for data error correction. Cho *et al.* [5] proposed a region based group key management protocol where the group is divided into subgroups on the basis of region. This algorithm reduces network traffic by balancing the inter-regional and intra-regional communication. In [6] Zhang *et al.* proposed a group key agreement protocol using k-bilinear Diffie-Hellman (DH) exponent over identity cryptosystems. This is an asymmetric dynamic group key management protocol where a single round provides the complete authentication for the secured users.

Sun *et al.* [7] proposed a group key management protocol where repeated one-way function tree is used to reduce the malicious attacks. In [8] a tree-based group key management protocol is proposed. This scheme utilizes the security pattern in hierarchical NoC based group communication. Interactive Diffie-Hellman algorithm is used to distribute group key among the members in a specific zone. Gomathi *et al.* [9] proposed a hierarchical distributed group key management protocol with integrated fuzzy trust clustering so that repeated refreshments of group key and re-keying in clusters can be avoided. In [10] the subgroup heads posse's two keys. One key for inter-group and another key for intra-group communication so that node join or leave within a subgroup can be managed locally. Although this algorithm is scalable but has the drawback of affecting the data path. Hydra [11] is proposed by Rafaeli *et al.*, has the same group key for inter and intra-group communication. The subgroup head initiates the re-key process on the membership change.

LEACH [12] the subgroup or the cluster heads (CH) are elected on the basis of battery power and it incorporates randomized rotation of the high energy cluster head position to avoid the drowning of the battery of any one node in the network. In [13] a weighted clustering algorithm is proposed where the cluster heads are selected by considering the weight of the node. The weight of each node is calculated on the basis of node degree, node energy and relative speed of a node. The minimum weight of a node promises its position in cluster head selection. This algorithm reduces the number of the clusters but the inter-cluster communication among the cluster heads that increase the network traffic is not considered properly. V. S. Anitha and M. P. Sebastian [14] proposed a weighted and adaptive algorithm where the energy level of node acts as the base for cluster head selection. The high energy level guarantees the CH position. This procedure executes extra computational steps that create a burden on network traffic. J. Sathiamoorthy and B. Ramakrishnan [15] proposed a hybrid scheme for dynamic cluster formation. Two algorithms are combined to attain the goal of stability of the cluster. This algorithm consumes more energy for cluster member as well as cluster head selection. In [16] a weight clustering technique is implemented to reduce the risk of forwarding hello packets in the cluster formation process. This algorithm considers the mobility pattern, degree and the time for which it is active to detect its battery power for weight calculation. In [17] clusters are formed according to region-clustering based on a hyper-sphere. This algorithm shows a positive result only when the node distribution is normal. In [18] a self-organization clustering algorithm based on the zone is proposed. The formation and maintenance of clusters in this scheme are inspired by the birds flocking behavior.

M. Gharib *et al.* [19] proposed a fully distributed ID-based key management scheme using ECC. The computational overhead is less and it also eliminates the central control. In [20] a cluster based and partially distributed key management approach is proposed. CH authenticates each cluster member based on its ID. The identity of nodes and their trust value is used to update keys periodically. In [21] communication within and between the clusters are proposed. Diffie-Hellman key exchange protocol is used for authentication and sharing of keys. The criterion of cluster formation is not mentioned and the computational cost using Diffie Hellman for key exchange is high. In [22] a pairing free certificate-less authentication scheme with batch verification for VANETs is proposed. This scheme overcomes the high cost of bilinear pairing through ECC and batch processing. In [23-24], ECHCKM a hierarchical cluster key management scheme based on the elliptic curve is proposed for the sensor network. This scheme implements elliptic curve encryption-decryption and ECDSA for key generation. The secure key exchange between root cluster head, cluster head and cluster members is deeply explained in this paper.

Above mentioned research papers consider the limitations of ad hoc network while designing their proposals but still cannot provide a comprehensive scheme. Therefore, we propose a secure and power-aware security scheme to provide a complete solution for security management.

3. PROPOSED SECURE AND POWER-AWARE PROTOCOL FOR WIRELESS AD HOC NETWORK (ECCDGKM)

3.1. ECCDGKM Structure

We consider a wireless ad hoc network of N nodes. The proposed secure and power-aware protocol ECCDGKM for decentralized group key management is implemented. This

scheme is divided into two phases. In the first phase, the whole network is divided into smaller group or cluster by a clustering algorithm TWCA. The TWCA is a trusted weight clustering algorithm. The weight of a node is expressed in terms of CPU power, battery power, memory capacity, trust value, number of neighbor nodes and nodes approachability to their neighbor nodes. The highest weighted node forms the first cluster and its neighbor nodes became its cluster members. The first selected cluster head (CH) elect the next CH from its members that have the highest weight among them. Therefore, all CHs are in communication range of its previous and next selected CH. This will eliminate the need for gateway nodes for inter-cluster communications. The weight of the cluster head is directly proportional to the number of neighbor nodes. Therefore, the size of the cluster will be according to the weight of CH that is the CH with high weight will manage a large number of nodes than other CHs. This scheme distributes key management load proportionally on all CH according to their weights.

In the second phase, CHs will generate and share local group key LGK for intra-cluster communications and global group key GGK for inter-cluster communications by implementing elliptic curve cryptography and digital signature.

3.2. Network setup: Offline initialization

A central trust party TP initializes the network. The TP ensure that all nodes should have a private-public key pair (x_i, y_i). The TP assign a trust value T according to the history of the nodes. The TP provide the revocation list to all nodes. The TP also publish the public bulletin Q, having network public parameters. After the initialization phase, TP leaves the network. In our proposal, we consider the elliptic curve $y^2=x^3+ax+b$, $46a^2+b\neq0$ defined over a prime field. The parameters of ECC in the prime case are (q, a, b, P, n, h) where q is the prime number defining the field of the elliptic curve, a and b are coefficients of the elliptic curve. P is the generator point on the curve, n is a non-negative number that defines the order of P and h denotes cofactor. A random number (r_i) is chosen by all nodes between 0 and n-1 as their private key. All nodes calculate their public key Q_i by multiplying generator point with a private key. The two keys will be (r_i, $Q_i = r_i*P$). The nodes share their public key with TP and are the part of public bulletin Q published by TP.

3.3. Online network management

3.3.1. Cluster formation: Trusted Weight Clustering Algorithm (TWCA)

All nodes participate in the cluster formation process. The nodes determine their computational value. The computational value (CV) of a node is the average of its CPU power, memory capacity and battery power at that instance of time. Every node sends a hello packet to know their neighbor nodes. Each node calculates the average of its CV, NC and T as its weight (W).

W= (CV + NC + T)/3, where CV is the computational power. It is the average of CPU power, Memory capacity and Battery power at time t. NV is the neighbor vector. It is the ratio of the number of neighbors to the total number of nodes. T is the trust value of node provided by TP.

Nodes send and forward packets containing W to neighboring nodes. The TP select the node with the highest weight (W) as the first cluster head and leaves the network. Each node also maintains a record of its neighbor, their Weight, Distance vector (DV) and weight index (WIN). Distance vector (DV) is the proximity between node and its neighbor i.e. rations of the distance between the node and its neighbor divided by the communication range of the node and WIN is the average of W and DV.

The TP initiate the process of cluster formation and then leaves the network. The first selected Cluster Head CH form a cluster and declare its neighbors as cluster members. The selected CH check the condition that if the number of its cluster members is equal to N-1 then the process is complete otherwise the following steps are performed

step 1. The CH selects the node with the highest WIN among its members. Declares it the next CH and exclude it from its cluster.

step 2. The selected CH forms a new cluster and declares its neighbors as its cluster members. The CH will not include any node that is previously selected either as CH or CM.

step 3. The process is completed when the union of all CH's and CM's is equal to the total number of nodes in the network otherwise step 1 to step 3 are repeated.

3.3.2. A new node joins the network

A new node broadcast joining request packet containing its ID and location. The receiving nodes forward the request to their CH. Three cases may arise.

Case 1: The new node is inside the communication range of only one CH then it will be added as its cluster member.

Case 2: The new node is inside the communication range of two or more CH then the CH with high weight will add the new node.

Case 3: The new node is not in the communication range of any CH then the CH selection process will be restarted.

In case 1 and 2, the local key is changed for backward secrecy. The trust value of the new node is assigned zero by the respective CH.

3.3.3. Node leaves either cluster or the network

If the node is mobile and leaves either the cluster or the network, the following cases may arise.

Case 1: Node is the member node. It leaves one cluster and joins another. In this case, the CH from where the node leaves will rekey the local key for forward secrecy. The CH where the node joins will add the node as a member and rekey the local key for backward secrecy.

Case 2: Node is the member node and leaves the network. In this case, the respective CH will rekey the local key for forward secrecy.

Case 3: Node is a CH. It either leaves cluster or network in both cases the process of selection of CH is restarted.

3.3.4. Reselection process

The key management activities like key generation, distribution and re-distribution of secure keys create an extra computational burden on CH. The reselection process circulates this burden. After even interval of time, the reselection process restarts the process of cluster formation so that the cluster maintenance load is shifted to more capable nodes at that particular time. This eliminates the risk of frequent drowning of CH and increases the life span of the network.

3.4. Intra-Cluster Communication

The local group key LGK is generated by CH. The CH share LGK with its members for intra-cluster communications. Let m be the number of CH and p is the number of their cluster members.

3.4.1. CH generate LGK

- A random number c lies between 0 and n-1 is chosen by CH.
- CH calculate LGK by multiplying generator point P by c i.e. LGK = c * P. The LGK will be a point on the elliptic curve defined above.

3.4.2. CH sends LGK to cluster members by performing the following steps.

- A random number x and generates a message M = x * P, where M will be a point on the elliptic curve defined above.
- The message M is encrypted with the public key Q_j of its cluster member CM_j, where j ϵ (1,2,……,p).
  a) A random number s is chosen by CH.
  b) Two points on elliptic curve E1 = s * P and E2 = Q_j + Ms generated.
- CH generate ECDSA signature of the message sign(M) by using its private key.
- CH sends the encrypted points E1, E2 and sign(M) to CM_j.
- CM_j decrypts E1 and E2 using r as its private key. Find M by M = E2 – r * E1.
- CM_j accepts M as the shared key if the sign(M) is verified otherwise reject the message. CM_j sends an acknowledgement to CH and in case of rejection, CH resends the shared key.
- The shared key Mx is the x coordinates of M.
- CH calculates the multiplicative inverse of Mx, multiply the LGK with Mx^{-1} and sends it to CM_j
  Ktemp = LGK * Mx^{-1}
- CM_j calculates the local group key by multiplying Ktemp with shared key Mx.
  Ktemp * Mx = LGK * Mx^{-1} * Mx
  = LKG
- CH repeats these steps with every cluster member.

3.5. Inter-Cluster Communication

CHs generate and shared Global group key for inter-cluster communication. The first selected CH will start the process and generate a global group key GGK. The CH shared GGK with next elected CH in the same manner as discussed in section 3.4.2. The CH share GGK with next CH and the process continues till the last CH gets GGK.

4. PERFORMANCE ANALYSIS

4.1. Key Storage

- All CH possess one private key, one public key, one LGK, one GGK, (1+p) shared keys (one for next CH and one each for p number of CM in the cluster) and (1+p) signatures (one for next CH and one each for p number of CM in the cluster). The number of keys a CH holds is 2p+6. Therefore the total number of keys held by all CHs is (2p+6) * m = $2\sum_{i=1}^{m} pi + 6m$.
- All CM possess one private key, one public key, one signature, one shared key, one LGK. The number of keys each CM holds is 5. Therefore, the total number of keys held by all CMs is $5\sum_{i=1}^{m} pi$.
- The total number of keys generated in the ECCDGKM, ECHCKM and Guo. *et al.* Scheme is shown in Table 1.

4.2. Number of Operations

- Each CH generates one pair of private/public keys. For distribution of LGK, p signatures are generated. 2p encryptions of messages and LGK are performed. For sharing of GGK to next CH, one signature generation, one message encryption and one GGK encryption are performed. The CH that receives GGK performs one signature verification, one message decryption and one GGK decryption. Thus, 3p+6 operations are performed by each CH.
- Each CM generates one pair of private/public keys, one signature verification, one message decryption and one LGK decryption.
- The total number of operation performed in ECHCKM, Guo. *et al.* Scheme and our scheme ECCDGKM is shown in Table 2.

4.3. Number of message exchange

- For local group key generation, four messages are exchanged. First, CH sends the signature of the message. Second, CH sends an encrypted message. Third, CM acknowledges CH for signature verification or requests another message and signature if the signature is not matched and fourth, CH sends the encrypted LGK.

Table 1: The total number of keys generated in ECHCKM, Guo. *et al.* Scheme and ECCDGKM.

	ECHCKM	**Guo. *et al.* Scheme**	**ECCDGKM**
Cluster Head	RCH: 2m+4 CH: 2p+7	4p+4	**2p+6**
Cluster member	6	7	**5**
Total keys	$8\sum_{i=1}^{m} pi+9m+4$	$11\sum_{i=1}^{m} pi+4m$	$\mathbf{7\sum_{i=1}^{m} pi+6m-2}$

Note: RCH: Root Cluster Head, CH: Cluster Head

Table 2: The total number of keys generated in ECHCKM, Guo. *et al.* Scheme and ECCDGKM

	ECHCKM	**Guo. *et al.* scheme**	**ECCDGKM**
Cluster Head	RCH: 1+4m CH: 4p+5	7p+2	**3p+6**
Cluster Member	5	7	**4**
Total number of operations	$9\sum_{i=1}^{m} pi+9m+1$	$14\sum_{i=1}^{m} pi+2m$	$\mathbf{7\sum_{i=1}^{m} pi+6m}$

Table 3: The total number of keys generated in ECHCKM, Guo. *et al.* Scheme and ECCDGKM

	ECHCKM [23]	**Guo. *et al.* [21]**	**ECCDGKM**
Cluster Head	4p + 2	7p+2	**4p**
Cluster Member	4	7	**4**
Total number of message exchange	$4\sum_{i=1}^{m} pi+ 4m +2$	$14\sum_{i=1}^{m} pi+2m$	$\mathbf{4\sum_{i=1}^{m} pi+4m}$

- For global group key generation, in the same way, four messages are exchanged. First, CH sends the signature of the message to the next CH. Second, CH sends an encrypted message. Third, the receiving CH sends the acknowledgement of signature verification or request for another message and signature if the signature is not matched. Fourth, CH sends the encrypted global group key. Hence, the total number of the message exchanged is $4\sum_{i=1}^{m} pi + 4m$.
- The total number of the message exchange for key management in ECHCKM, Guo. *et al.* Scheme and ECCDGKM is shown in Table 3.

4.4. Key Size

The length of keys is also an important factor while considering a reduction in computational and communication cost for key management activities. Our scheme is based on elliptic curve cryptography (ECC). The ECC is public key cryptography and previous studies reveal that public key cryptography is a better approach for wireless ad hoc networks rather than private key cryptography because of wireless communication medium.

One of the most important public key cryptographic algorithms includes Robin's scheme RSA. The computational complexity of encryption-decryption algorithms in RSA is high and time-consuming as compared to ECC algorithms [25]. The key size of ECC is much less than RSA key size with the same level of security. RSA with 1,024-bit key provides an acceptable secrecy level whereas ECC with 160-bit key provides the same secrecy level.

Guo. *et al.* [21] implement RSA with Diffie-Hellman key exchange in its scheme whereas ECHCKM [23] and our scheme ECCDGKM uses ECC with Diffie-Hellman key exchange.

5. EXPERIMENTAL RESULTS

The experiment is performed with a different number of nodes. The simulation parameters are mentioned in Table 4. The experiment is repeated number of times to observe various parameters like the number of the cluster formed, the number of keys generated, the number of operations performed for key generation and its sharing and lastly the number of message exchange during key management activities. The results of our scheme ECCDGKM are compared with ECHCKM [23] and Guo. *et al.* scheme [21].

Table 4: The simulation parameters

Simulation	**Values**
Simulation Area	1000x1000 sq. mts
Number of Nodes	50,100,150,200
Transmission range	50m
Mobility model	Random walk model
Simulation time	300 seconds

The simulation is performed for a large network area with network size of 50, 100, 150, 200 nodes. The experiments are performed a number of times and the average is taken while comparing with other schemes. The result in Fig. 1 shows that the number of clusters formed in our scheme is much less than the [21] and [23]. The consideration of computational power constituting of CPU power, memory capacity and battery power along with the number of neighbor nodes in the formation of clusters reduces the total number of clusters in the system. For intra-cluster communication LGK and for inter-cluster communication GGK is generated in our scheme and similarly, various keys are generated for secure communication in schemes [21] and [23].

The results shown in Fig. 2 depicts that the number of keys generated in our scheme is less than these schemes. A large number of operations are performed for the generation and sharing of keys. We perform the simulation with different network size and calculate the number of operations performed in our scheme and compare it with other schemes [21] and [23].

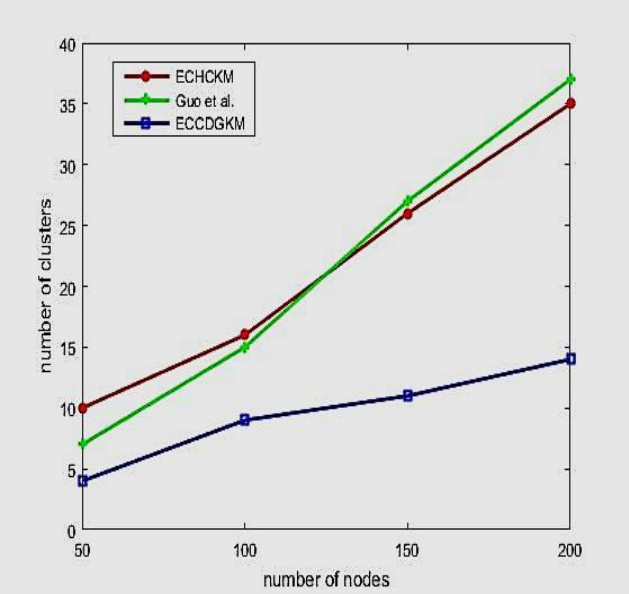

Fig. 1: Numbers of clusters formed verses network sizes in [21], [23] and our scheme.

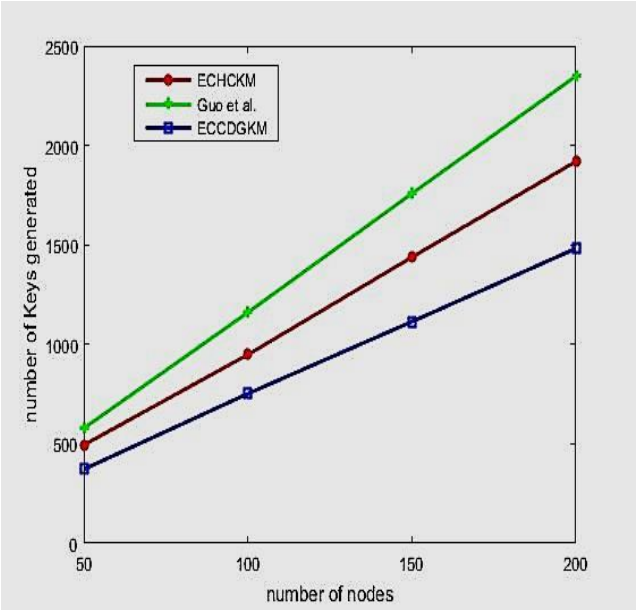

Fig. 2: Total number of keys generated verses different network sizes in [21], [23] and our scheme.

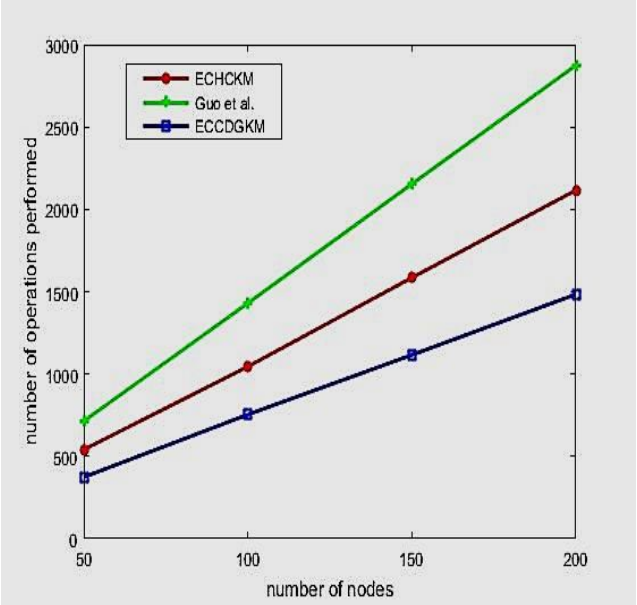

Fig. 3: Total number of operations performed for inter and intra-cluster communications versus different network size in [21], [23] and our scheme.

The results in Fig. 3 shows that the total number of operations performed in our scheme is much less than other considered schemes. Similarly, we calculate the number of message exchange during these activities and found that here also our scheme performs little better than ECHCKM [23] and much better than Guo. *et al.* scheme [21] as shown in Fig. 4.

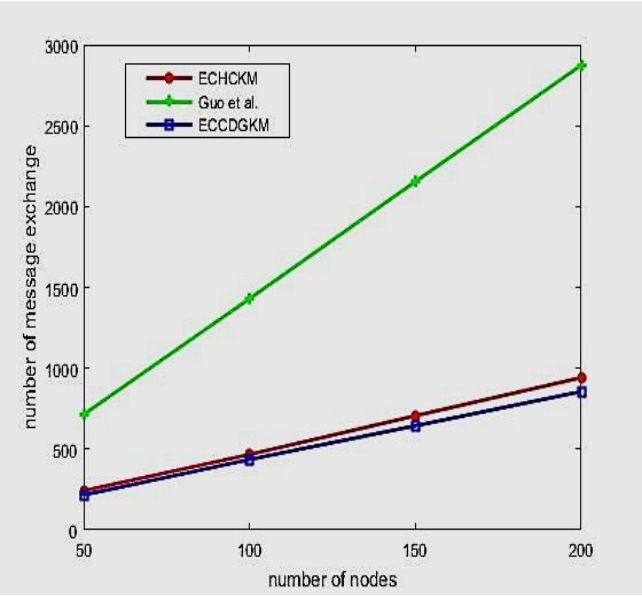

Fig. 4: Total message exchanged verses different network sizes in [21], [23] and our scheme.

6. CONCLUSION

The secure and power-aware protocol ECCDGKM is a decentralized key management scheme based on elliptic curve cryptography. The scheme is divided into two phases. In the first phase, the trust-oriented weight clustering algorithm TWCA select cluster heads and form clusters by considering nodes computational power, the number of neighbors, the proximity with the neighbors and their trust value. The node with high weight will manage a large cluster and therefore, reduce the number of clusters in the network. TWCA eliminate the need of gateway node and also distribute a load of key management activities on CH according to their weights. Therefore, reduce the frequent drowning of CHs. The second phase establishes the intra-cluster and inter-cluster communication with the help of ECC enabled Diffie-Hellman and Digital signature schemes. The experimental results clearly depict the advantage of our scheme over ECHCKM and Guo. *et al*. schemes. Our scheme shows better results in terms of the number of clusters formed, the number of keys generated, the number of operations performed and the number of the message exchanged than ECHCKM and Guo. *et al*. scheme.

REFERENCES

1. Basagni S, Conti M, Giordano S, Stojmenovic I. **Mobile Ad Hoc Networking: Cutting Edge Directions**, Second Edition, Chap-1. John Wiley and Sons; 2013.
2. Zhang Y, Lee W. **Security in Mobile Ad-Hoc Networks. Ad Hoc Networks Technologies and Protocols**. Springer; 2005.
3. Rafaeli S, Hutchison D. **A survey of Key Management for Secure Group Communication**. ACM Computing Surveys.2003(pp. 309-329, Vol. 35, No. 3)
4. Varalakshmi R, Uthariaraj V R. **Huddle hierarchy based group key management protocol using gray code**. Wireless Network. 2014(vol. 20, pp. 695-704)
5. Cho J H, Chen R. **Performance analysis of hierarchal group key management integrated with adaptive intrusion detection in mobile ad hoc networks**. Performance Evaluation. 2011(vol. 68(1), pp.58-75)
6. Zhang L, Wu Q, Domingo-Ferrer J, Qin B, Dong Z. **Round-efficient and sender-unrestricted dynamic group key management protocol for secure group communications**. IEEE Trans. Inf. Forensics Security. 2015(vol- 10(11), pp. 2352-2364)
7. Sun Y, Chen M, Bacchus A, Lin X. **Towards collusion-attack-resilient group key management using one-way function tree**. Computer Networks. 2016(104, pp. 16-26)
8. Sepulveda J, Flórez D, Immler V, Gogniat G, Sigl G. **Efficient security zones implementation through hierarchical group key management at NoC-based MPSoCs**. Microprocessor. Microsyst. 2017(50, pp. 164-174)
9. Gomathi K, Parvathavarthini B, Saravanakumar C. **An efficient secure group communication in MANET using fuzzy trust based clustering and hierarchical distributed group key management**. Wirel. Pers. Commun.2017 (https://doi.org/10.1007/s11277-016-3366-x)
10. Mittra S. **Iolus-A framework for scalable secure multicasting**. ACM SIGCOMM. 1997(Vol. 27, no. 4, pp. 277-288)
11. Rafaeli S, Hutchison D. **Hydra: A decentralized group key management**. 11th IEEE International WETICE: Enterprise Security Workshop, June (2002)
12. Heinzelman W, Chandrakasan A, Balakrishnan H. **Energy-Efficient Communication Protocols for Wireless Microsensor Networks**. In Proc. 33rd Hawaaian Int. Conf. on Systems Science. 2000
13. Tao Y, Wang J, Wang Y L, Sun T. **An enhanced maximum stability weighted clustering algorithm in ad hoc network**. In Proc. 4th Int. Conf. Wireless Communication Networks. Mobile Comput. 2008 (pp. 1-4)
14. Anitha V S, Seastian M P. **(k,r)-dominating set-based, weighted and adaptive clustering algorithms for mobile ad hoc networks**, IET Commun., 2011(vol. 5, no. 13, pp. 1836-1853)
15. Sathiamoorthy J, Ramakrishnan B. **Energy and delay efficient dynamic cluster formation using hybrid AGA with FACO in EAACK MANETs**. Wireless Network. 2017(vol. 23, no. 2, pp. 371-385)
16. Maragatham T, Karthik S, Bhavadharini R M. **TCACWCA: transmission and collusion aware clustering with enhanced weight clustering algorithm for mobile ad hoc networks**. Cluster Computing. 2018(https://doi.org/10.1007/s10586-017-1574-0)
17. Salma B U, Lawrence A A. **Improved group key management region based cluster protocol in cloud**. Cluster Computing. 2017(https://doi.org/10.1007/s10586-017-1455-6)

18. Aftab F, Zhang Z, Ahmad A. **Self-Organization Based Clustering in MANETs Using Zone Based Group Mobility**. IEEE Access. 2017(https://doi.org/10.1109/ACCESS.2017.2778019)
19. Gharib M *et al.* **Fully distributed ECC-based key Management mobile ad hoc networks**. Computer Networks. volume 113, issue C, pages 269-283, doi: 10.1016/j.comnet.2016.12.017, 2017.
20. Lakshmi R P, Kumar A V A. **Parallel key management scheme for mobile ad hoc network based on traffic mining**. IET Information Security 9(1): 2015(pp. 14-23)
21. Guo M H, Liaw H T, Deng D J, Chao H C. **A cluster based secure communication mechanism in wireless ad hoc networks**. IET Information Security. 2010(4(4): pp. 352–360)
22. Gayathri N B *et al.* **Efficient Pairing-Free Certificateless Authentication Scheme with Batch Verification for Vehicular Ad-hoc Networks**. IEE ACCESS. 2018 (Digital Object Identifier 10.1109/ACCESS.2018.2845464)
23. Sahoo S K, Sahoo M N. **An Elliptic-Curve-Based Hierarchical Cluster Key Management in Wireless Sensor Network.** Proceedings of the International Conference on Advanced Computing, Networking, and Informatics, India. June 2013(pp-397)
24. Harshita Chaurasiya, Dr. Shivnath Ghosh. **Performance Evaluation of Energy-Efficient Cluster based Algorithms in Wireless Sensor Network,** Published in International Journal of Advanced Trends in Computer Science and Engineering, pp. 77-81, Volume 7, No. 5, 2018.
doi.org/10.30534/ijatcse/2018/03752018.
25. Bafandehkar M, Yasin S M, Mahmod R, Hanapi Z M. **Comparison of ECC and RSA Algorithm in Resource Constrained Devices**. **Published in** International Conference on IT Convergence and Security (ICITCS). 2013 (**DOI:** 10.1109/ICITCS.2013.6717816)