Exclusion of malicious nodes using Secure Proactive Secret Sharing (SPSS) for Threshold Cryptography in MANETs

P. Swetha,
Department of CSE, JNTUH College of Engineering Jagityal, Nachupally, Karimnagar, TS

ABSTRACT
The malicious nodes in MANETs can cause network malfunction, resulting in the damage of other nodes. Utmost significance is given to establish a secure communication within the group members of a network. The primary objective is to develop an appropriate model for improved security in MANETs. A new model called Secure Proactive Secret Sharing (SPSS) is proposed which resolves the various challenges that are identified in Threshold Cryptography. The SPSS model, is developed for the secret key transmission, where the secret key is divided into shares and these shares are distributed to the nodes taking part in communication. The nodes collect the shares, and a threshold number of shares are needed to reconstruct the secret key. The challenge in SPSS model is to identify the honest group head and honest shareholders. The model monitors and eliminates any malicious node that get identified, from taking part in communication. The Secret key is refreshed at regular intervals, to abstain any exposure to intruders if the same key is used for prolonged duration. Further, within the honest nodes, if any node misbehaves and acts selfishly to preserve its resources or drops packet without forwarding to the next hop or modifies the packet, a new tree of honest nodes called Directed Acyclic Graph (DAG) is created to eliminate that selfish node from the group. Hence, this SPSS model is secure enough to protect against malicious nodes and selfish nodes. The simulation of SPSS with packet droppers and modifiers is evaluated for various performance metrics.

Keywords: *Threshold Cryptography, Proactive Secret Sharing, Commitments, Homomorphic encryption*

I. INTRODUCTION

The communication in a mobile network can be called secure and protected, if the availability of secret key is restricted to only the two communicating entities. A Cryptographic technique called Threshold Cryptography (TC)[2], helps us in distributing the secret key. In TC, a secret key [12] is divided into multiple shares, and distributed to the nodes participating in the communication in an infrastructure less network. In (n, t) TC scheme [3], a secret key is divided into 'n' shares and shared among 'n' nodes using some cryptographic operation. Any node can collect 't' threshold number of shares and can reconstruct the original key 'K' [7]. On contrary, it is not possible for any '$t-1$' nodes to construct the key 'K' even by collusion. A '$t-1$' degree polynomial is constructed with the constant secret key 'K' and random elements

$$y = f(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + a_{t-3}x^{t-3} + \cdots + a_1 x + K$$

However it is possible for a malicious node to generate the secret key by stealing 't' or more shares from the participating nodes, within long span of time. Proactive Secret Sharing (PSS) can be introduced to escape from the threats of exposing a secret key. PSS plays an important role of a key management protocol using threshold cryptography. In PSS, each share is periodically updated, with an intention to block a malicious node from constructing the secret key [4] within the short time span. In PSS scheme, all shares are refreshed with a new set of shares that are generated from the old shares for the same secret, and the old shares are discarded after each share is refreshed. For protocol consistency, all shareholders must cooperate with the PSS.

II. CHALLENGES IN THRESHOLD CRYPTOGRAPHY

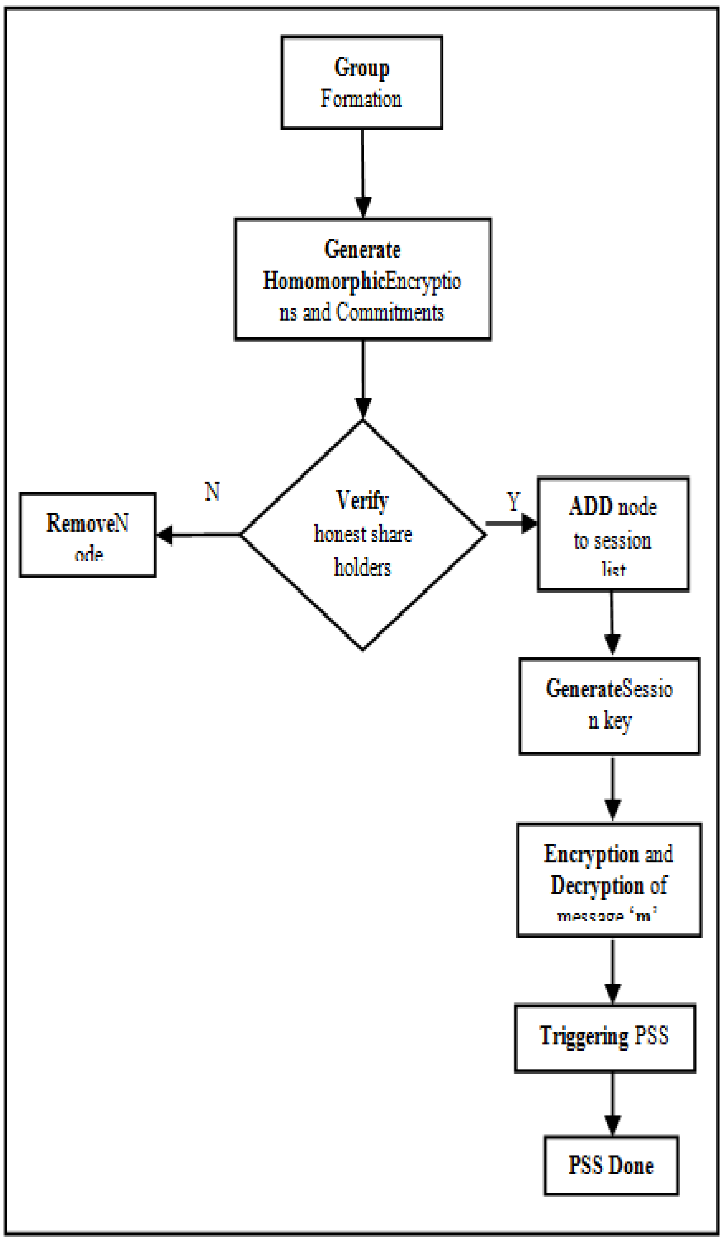
Threshold Cryptography is only considered in situation where all the participating users are legitimate shareholders. In TC, traditionally the dealer (master node) is trusted, but, this does not always hold. The following are the issues in the traditional threshold cryptography which are left unanswered,

The initial group formation
Sharing of group key among group members
Selection of the dealer
Verification on proper distribution of shares
Verification of adversaries in the group

Verification of legitimacy of the participating shareholders or attackers
All share holder nodes must synchronize the PSS procedure in a well-managed fashion to keep the protocol consistency.
These issues are to be solved for the improvement of the protocol.

III. PROPOSED MODEL – SECURE PROACTIVE SECRET SHARING (SPSS) MODEL

In this work, a model known as Secure Proactive Secret Sharing (SPSS) is proposed. This model resolves the issues present with the traditional Threshold Cryptography. This model defines how the nodes in the network form the group and how the Group Head (Dealer) [1] is selected, which constructively distributes the shares of the secret key among the group members. The process flow of SPSS model is shown in the Fig.1.


Group Formation
Generate HomomorphicEncryptions and Commitments
Verify honest share holders
N
Y
RemoveNode
ADD node to session list
GenerateSession key
Encryption and Decryption of message 'm'
Triggering PSS
PSS Done

Process Flow of the SPSS Model

The first module is the group formation and selecting a Group Head (master node/ Dealer), where the responsibility of group head is sharing the secret key among the group members.
The next module is to identify the legitimate nodes for the secret reconstruction using Homomorphic commitments [5], and further the encryption and decryption of messages is performed. All shareholders must

cooperate with the PSS procedure for the protocol consistency. In addition to the above, one more module is developed for PSS synchronization.
In a MANET, all mobile nodes that are willing to participate in communication has to form a group initially. The group formation module shows the steps for the group formation. The next step is for the group member nodes to elect one node as Group Head (Dealer). The responsibility of a dealer is to select a secret key which is divided into sub shares (subkeys) and distribute these shares to the group members by generating encryptions and commitments. Each shareholder node receives the commitments that further verify whether sub shares are distributed correctly by the dealer or not. Both the shareholder node and the dealer are subjected to this verification. This verification process validates if the commitment of the secret key is same as the cumulative commitments of the individual sub shares. If the commitments are same, the verification for the shareholders is successful, then it adds the node to the session list otherwise remove node from the session list. Later, the group head generates a session key and then the encryption and decryption of the message, for the refreshment of the shares is done, after which PSS module is triggered for PSS synchronization.

i. Group Formation

Assume a network of *N nodes* where N= *{* $N_1, N_2, \dots N_n$ *}*. Each node is certified by the Certificate Authority (CA) and the Certificates issued to these nodes are generated using ECDSA approach. Fig.2. shows the detailed steps for group formation

1. Initially each node in the network generates its own random binary value [0,1].
2. A Certificate authority (CA) periodically broadcasts its announcement to all the nodes in the networks.
3. The nodes which are willing to participate in the communication responds to the broadcast message by sending a SEND_CERTIFICATE_REQUEST message to CA along with its key and node_id.
4. CA generates the certificate which consists of node_id, public key, hash value along with the timestamp and sends to the requested nodes.
   CA (Ni) = {node_id, Pub. Key, Hash value, timestamp}
5. Nodes store the received certificate.

All the nodes with their respective CAs are formed in the group for the secure communication between the two communicating entities.

ii. Selecting Group Head (Dealer)

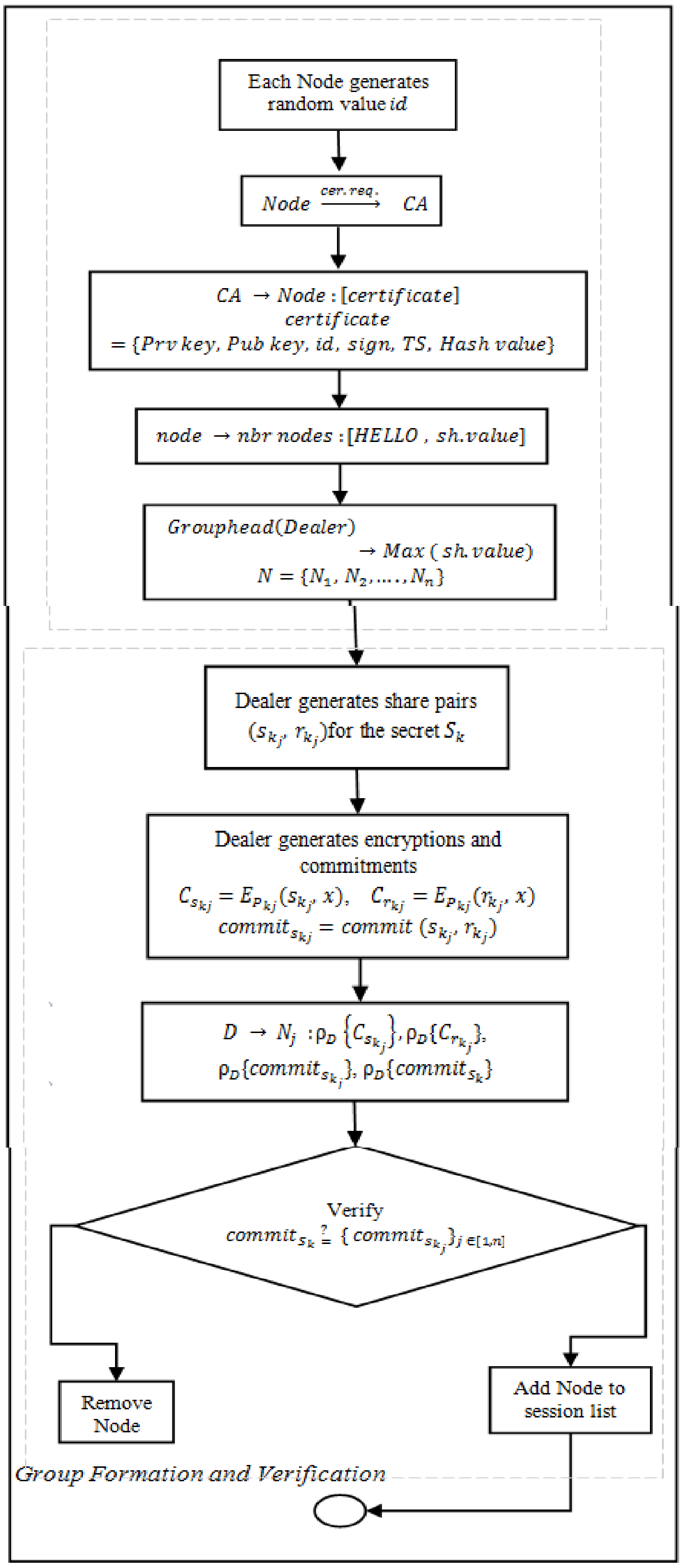
After group formation, all the nodes have to select one node as the Group Head /Manager (Dealer), which is responsible for Key Generation and Key Distribution

1. All nodes in the group broadcast HELLO message with its own random value.
2. Neighbour nodes receives the broadcast messages of other nodes and stores them with their respective random values
3. A node which has the maximum random value among the neighbouring nodes is selected as the Group Manager (Dealer)
4. The GM node announces its random value and its status as Dealer to all other nodes on the group.
5. Now the Dealer's node_id becomes its random value and the rest of the nodes in the group are called share holder nodes.
6. Dealer (GM) then sends JOIN message to its shareholder nodes where all nodes are joined
7. Once all the nodes joins in a group it starts request timer to request for shareholder.
8.

iii. Share Computation, Distribution and Verification

This model works in two different variants:

- Iterative Subshare Calculation: In Iterative sub-share calculation, the time of sub-share calculation is given by step-by-step manner for share transmission and share reception.


Each Node generates random value id
Node $\xrightarrow{cer.req.}$ CA
CA → Node : [certificate] certificate = {Prv key, Pub key, id, sign, TS, Hash value}
node → nbr nodes : [HELLO , sh.value]
Grouphead(Dealer) → Max (sh.value) N = {N₁, N₂, ….., Nₙ}
Dealer generates share pairs (s_{k_j}, r_{k_j})for the secret S_k
Dealer generates encryptions and commitments
$C_{s_{kj}} = E_{P_{kj}}(s_{k_j}, x)$, $C_{r_{kj}} = E_{P_{kj}}(r_{k_j}, x)$
$commit_{s_{kj}} = commit\ (s_{k_j}, r_{k_j})$
$D \rightarrow N_j : \rho_D\{C_{s_{k_j}}\}, \rho_D\{C_{r_{k_j}}\}, \rho_D\{commit_{s_{k_j}}\}, \rho_D\{commit_{s_k}\}$
Verify
$commit_{s_k} \overset{?}{=} \{commit_{s_{k_j}}\}_{j\in[1,n]}$
Remove Node
Add Node to session list
Group Formation and Verification

- Parallel Subshare Calculation: In Parallel sub-share calculation, the time that each shareholder preliminarily generates own sub-shares instead of waiting for the sub-share transmission in a parallel fashion.
  The cost of sub-share calculation is much smaller than the delay of sub-share transmission over TCP in a wireless network.The newly selected Dealer node has to generate the secret key for distribution among the shareholder nodes. Let S_k be the secret. The dealer divides the secret key S_k into shares to be distributed

among the shareholder nodes. The shareholders $N_1 \ldots N_n$ holds the shares of S_k and the dealer *D* knows the secret of S_k.

a. Share generation:

Let $S_k = (s_{k1}, s_{k2}) \in Z_q$ be the secret to be shared. The dealer *D* initiates the protocol by committing to S_k. (s_{k1}, s_{k2}) uniquely determines the two polynomials *f(x)* and *r(x)*, such that $f(0)=S_k$.The secret S_k is distributed among *N* shareholders where $N \in \{N_1, N_2 \ldots N_n\}$.If some shareholder$N_i$ *is* malicious, then it can input a wrong share to the recovery protocol. There by the recovered secret S_k will be incorrect. There is also a possibility of malicious dealers who may give inconsistent shares to any shareholder nodeN_i .shareholders commit to the shares they have received. A malicious share holder $N_i$may send wrong key to one share holder and the correct share of S_k to the other. In such cases, the secret key S_k cannot be recovered. For discovering honest shareholders and honest dealer, verifiable secret sharing is used. For verification process, Homomorphic commitments are used. This verification is depicted in Fig.2.

b. Key Distribution:

The dealer generates the share pair(s_{ki}, r_{ki}) and distributes to each shareholder node. Each user can compute a correct pair (s_{ki}, r_{ki}) on the committed polynomials where

$s_{ki} = f(i)$

$r_{ki} = r(i)$

Where S_{ki} is the share of the secret key and $r_{ki}$is the random polynomial. The encryption of $s_{kj}$and r_{kj} is performed by the ElGamal public key encryption. Though, ElGamal has disadvantage that the Ciphertext is twice as long as the plaintext, it has the advantage that the same plaintext gives a different Ciphertext each time it is encrypted. The dealer computes the commitment of the secret key S_k and also for the subshares$S_{kj[j \in [1,n]]}$. These encrypted share pairs along with the commitment are signed by the dealer and sends to each shareholder. The dealer *D* then sends the encrypted commitments along with their signatures:

$\rho\{C_{skj}\}$, $\rho\{C_{rkj}\}$, $\rho\{Commit_{skj}\}$ for all $j \in [1,n]$ and $\rho\{Commit_{sk}\}$ to every share holder.

c. Verification

Verifiable secret sharing schemes are based on Shamir's work. These schemes allow shareholders to determine whether the dealer sent them valid shares of the secret or not.

Step-1: Generate the share pair (s_{kj}, r_{kj}) by performing the description and forward the triplet $\{\rho\{C_{skj}\}, \rho\{C_{rkj}\}, \rho\{commit_{skj}\}\}$ to N_j where $N_j \in N$ and

$s_{kj} = D_{pkj}\{C_{skj}\}$

$r_{kj} = D_{pkj}\{C_{rkj}\}$

Step-2: Wait for the triplet to be forwarded by some party. On receiving compute the share pair (s_{kj}, r_{kj}) as in step-1. Send the share pair (s_{kj}, r_{kj}) to every N_j where $N_j \in N$. Verify whether

$commit_{skj} \overset{?}{=} commit(s_{kj}, r_{kj})$

$commit_{skj} = f^{skj} g^{rkj}$

If verification is successful, then include the shareholder node that produced s_{kj} in the Session List (SL_j), otherwise remove the shareholder from the Session List.

d. Reconstruction

1. Each node sends its share pair to all the nodes, which verifies with the corresponding commits available with the nodes if $|SL_j| = t+1$
2. Once *'t+1'* correct share pairs are received, the sharing polynomial, hence s_k is reconstructed.
3. Construct a *t*-degree polynomial *f(x)* by interpolating$\{(j, s_{kj})\}_{skj \in slj}$.

Therefore f(0) = S_k where the secret key is reconstructed.

e. Session Key Generation

Once the honest dealer and honest shareholders are formed, the new session key is generated for encrypting the messages. Each shareholder has the partial session key, from which the session key is generated for decrypting the message. The module for performing encryption and decryption of the message is shown in Figure 3.

f. Key Reconstruction

Each shareholder sh_i uses his pair of private key $a(x_i)$, $b(x_i)$ to compute a partial session key y_i. If all the participating users are legitimate shareholders and act honestly, the cipher text (c_1, c_2) of the message $'m'$can be successfully decrypted.If all shareholders act honestly, the real session key k_2. After collecting all the partial session keysy_i's from other shareholder, the member can decrypt the cipher text as $m = c_2 k_2^{-1} \bmod p$

Since the exponent of each partial session key is a linear combination of two shares, $a(x_i)$, $b(x_i)$ the attackers cannot separate these two shares to obtain the modular exponentiation of each share. The attackers need to collect all partial session keys to be able to decrypt the cipher text of the group. Therefore, if there are attackers participated in the decryption, the cipher text can't be decrypted. All the shares must be in consistent state after refreshing with the new shares. That is, if the shareholders are not consistent with the PSS procedure then some shareholders may use old shares while the others may use new shares. If this inconsistency is carried out then the secret key cannot be reconstructed. To achieve consistency in PSS all shareholders must be synchronized. For this a simple technique is used for triggering PSS. The module for PSS synchronization is shown in Figure 3.

iv. Triggering PSS

1. Shareholders request to the Dealer by sending a PSS_REQ request message with multicast group address
2. Dealer after receiving PSS_REQ message verifies the multicast address and calculates the next sec time interval and make shareholders to wait for waiting time by sending PSS_WAIT message
3. For this the dealer sends PSS_WAIT msg along data1 and data2 by encrypting with its session key and computes hash value and attaches these ciphers
4. Shareholders retrieves session key and then decrypt the cipher using session key to verify the data
5. Then shareholders starts waiting for nextSEC time interval

v. Share Refreshment

1. Now Shareholders calculates new share values and then sends notify message to the dealer
2. Once the dealer receives notify message from all the shareholders, it indicates that PSS procedure is performed successfully and sends PSS_SUCCESS message. Then performs PSS_SUCCESS
3. Else performs PSS_Suspend
4. Shareholders periodically generates the token and trigger PSS process
5. Shareholders receives this triggered PSS and initiate to send PSS_REQ, now check for stop token then rebroadcast PSS
6. Once PSS done all Shareholders start using its new subshare value
7. If PSS is suspended then all share holder stop rebroadcasting the token value

vi. Detection of Selfish nodes

In a MANET, each mobile node has features like autonomous, limited battery power, dynamic topology etc. The mobile node transfers packet directly to another node or through some intermediate nodes. The mobile node has limited energy resources like battery power, limited bandwidth. Due to this, the intermediate nodes may acts as a selfish node or malicious nodes which does not forward packet instead drops them. A mechanism for detection, mitigation of packet dropping attack is presented based on the cooperative participation of nodes in a MANET. Within the honest nodes, if any misbehaving node acts selfishly to preserve its resources and drops the packet without forwarding to the next hop or modifies the packet, a new Directed Acyclic Graph (DAG) is created to eliminate that selfish node from the group. Selfish nodes may or may not collude with each other. A compromised node can launch the following two attacks:

- Packet dropping: a compromised node drops all or some of the packets that it is supposed to forward. It may also drop the data generated by itself for some malicious purpose such as accusing innocent nodes.
- Packet modification: a compromised node modifies all or some of the packets that it is supposed to forward. It may also modify the data it generates to protect itself from being identified or to accuse other nodes.

vii. Dealing with Packet modifiers

The idea is to use a customized packet drop protocol [6][7] to forward the packets from one node to another node. The sender node initially checks to which node the packet should be forwarded. The node information of each node is known to another node. The sender node usually maintains the sequence number count when a packet is forwarded from one node to another. When a packet is forwarded from one node to another node a sequence number is added. The sender node usually tests the sample packet before forwarding the actual packet being sent

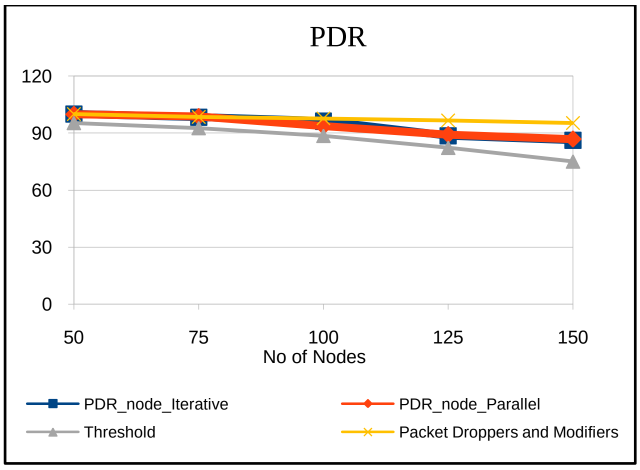
P	Packet to be sent
Bk1	Bloom Key1
Bk2	Bloom Key2
T1	Time within the acknowledgement should be received
C	Cipher text of original packet

Figure 4 shows the workflow of packet modifiers. At first, the sender node calculates the Bloom key (Bk1) by calculating the hash of the packet, encrypts the packet and forwards to next node. When a packet is forwarded from one node to another node, an acknowledgement should be received in a fixed time T1 and sequence number should be added at sender node. If the acknowledgement is not received in Time T1, the sender node checks for packet audit request. The Bloom key (Bk2) is generated after the packet is received. The difference in both the bloom keys detects the packet modification attack [10]. If the acknowledgement is received to sender in a fixed time T1, then there is no packet modification and there is a secured transmission of data in the network.

IV. RESULTS

The Simulation is performed using network simulator ns-2.
The SPSS model is compared with the existing models. The following are the graphs for the performance evaluation of the SPSS with metrics Packet delivery ratio, end-to-end delay and throughput.

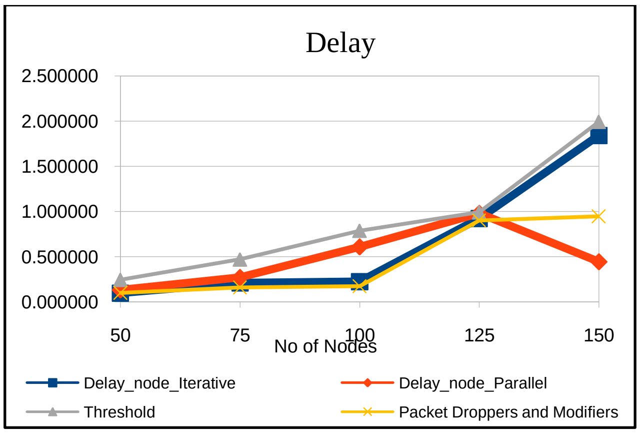
(a)Packet delivery ratio
Packet Delivery Ratio (PDR) is the ratio between the number of packets transmitted by a source node and the number of packets received by destination node.



Comparison of Packet Delivery Ratio with different models

In Fig.5. Graph indicates that SPSS model with Packet droppers and Modifiers has maximum packet delivery ratio when compared to other models.

(b) End to end delay
It indicates the time taken for one packet to travel from the source node to the destination node.

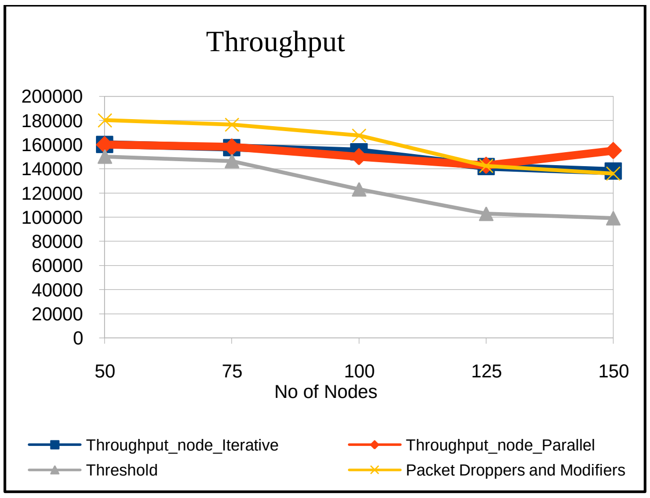


Performance End to End Delay with different models

The above graph indicates that as the number of nodes increases, the delay increases more in iterative subshare calculation when compared to parallel subshare calculation. Since in parallel calculation process, all nodes perform the operation parallelly The SPSS with Packet droppers and modifiers has minimum delay compared to other models.

(c)Throughput

It is defined as the total number of packets delivered over the total simulation time. The throughput comparison shows that the three algorithms performance margins are very close under traffic load of 50 and 100 nodes in MANET scenario. The graphs shows that SPSS with Packet droppers and modifiers gives better throughput compared to other models.



Throughput Comparison with different models

V.CONCLUSIONS

The SPSS model is developed for the secret key transmission by dividing the secret key into shares and distributes these shares to the nodes taking part in communication. The SPSS model is secure enough to protect against malicious nodes and selfish nodes. It is evident from the results that the SPSS model alone or with the addition of packet droppers and modifiers has better performance compared with other models with respect to evaluation parameters like throughput, packet delivery ratio, and end-to-end delay.

REFERENCES

[1] P. Swetha ,Dr. P. Premchand , " Secret Key Sharing Using Homomorphic Commitments and its application to Threshold Cryptography" , International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 9 (2016) pp 6598-6602.

[2] Hitoshi Asaeda, Musfiq Rahman, Yoshihiro Toyama, Structuring proactive secret sharing in mobile ad-hoc networks ,IEEE 2006 1st International Symposium on Wireless Pervasive Computing 0-7803-9410-0/06/©2006 IEEE.

[3] P. Swetha, Implementation of Threshold Cryptography in MANETS, International Journal of Engineering Research & Technology (IJERT)Vol. 3 Issue 1, January-2014 IJERT ISSN: 2278-0181

[4] A. Shamir, How to share a secret, Communication of the ACM, vol.22, 1979.

[5] J. C. Benaloh, "Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret," Advances in Cryptology—6th Annual International Cryptology Conference (CRYPTO '86), Santa Barbara, 17-21 August 1987, pp. 251-260.

[6] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,"*ACM MobiCom*, August 2000 P. Swetha, VinodBhupathi, "Unmasking Of Packet Drop

[7] Attack In Manet " , *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),*Volume 2, Issue 6, November – December 2013 ISSN 2278-6856

[8] E. F. Brickle and D. R. Stinson, "The Detection of Cheaters in Threshold Schemes," Advances in Cryptology—9th Annual International Cryptology Conference (CRYPTO '88), Santa Barbara, 21-25 August 1988, pp. 564-577.

[9] L. Zhou and Z. J. Haas, Securing Ad Hoc Networks, IEEE Network Magazine, vol.13, no.6, Nov/Dec 1999.

[10] J. Hubaux, L. Buttyan and S. Capkun, The Quest for Security in Mobile Ad Hoc Networks, Proc. ACM MobiHOC, October 2001.

[11] V. Bhuse, A. Gupta, and L. Lilien, "Dpdn: Detection of packet-dropping attacks for wireless networks,"*In the Trusted Internet Workshop,International Conference on High Performance Computing*, December 2005.

[12] Wu, B., Wu, J., Fernandez, E., Magliveras, S., and Ilyas, M. (2005). Secure and Efficient Key Management in Mobile Ad HocNetworks. Proc. of 19th IEEE International Parallel &Distributed Processing Symposium, Denver.