Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766205

INTRUSION DETECTION IN MANET USING CO-OPERATIVE BAIT BAED APPROACH

Prolay Ghosh[1], Dr. Nisarg Gandhewar[2]

1 Research Scholar, Department of Computer Science and Engineering, Dr. A.P.J Abdul Kalam University, Indore, M.P.& Assistant Professor, JIS College of Engineering, Kalyani, West Bengal

2 Research Guide, Department of Computer Science and Engineering, Dr. A. P. J. Abdul Kalam University, Indore, M.P.

ABSTRACT:

The collaboration of all participating Nodes relies on the mobile ad hoc networks (MANETs). The more traffic nodes collaborate, the stronger a MANET is. However, MANET support is a cost-intensive mobile node operation. Roads and packets are detected by local CPUs and need time, memory, network bandwidth, and ultimately but not least energy. It is thus highly motivated that a node denies packet forwarding to others while simultaneously providing its own data utilizing their services. Data analyzes the various ways to conserve resources in a MANET using the DSR routing protocol for an ego node. It utilizes Bruce Schneider's attack-tree rating which categorizes assaults, all of which lead an assailant to a particular objective.

Keywords: Co-Operative, Bait, Intrusion, Manet, Data

INTRODUCTION:

To achieve this objective, alternatives are indicated by OR, which involves several stages using AND. We can simply explain various assaults by using the numbers in the table. Attack 3.1, for example, means "Drop data packets" In the absence of two assaults in the attack-tree that can be readily identified, most attacks based on routing data manipulation using a secure routing protocol such as Ariadne SRP, ARAN or SA ODV remain [1]. All secure routing methods fail when nodes just discard the packets (case 1.1 and 3.1 in the attack tree), since they are focused

Table 1: Attack Tree: Save own resources

Attack tree: Save own resources							
OR	1.	Do not participate in routing					
	OR	1.	Do not relay routing data (case A)				
		OR	1.	Do not relay route requests			
			2.	Do not relay route replies			
			3.	Set hop limit or TTL value in route request/reply to smallest possible value			
		2.	Modify routing data/topology				
		OR	1.	Modify route request			
			OR	1.	Insert additional hops		
				2.	Modify route reply		
				OR	1.	Replace own ID in returned route with detour leading through neighboring nodes	
					2.	Return completely wrong route, provoking RERR and salvaging	
					3.	Insert additional hops	
					4.	Declare own ID in source route as external	
	2.	Stop participation in current route					
	AND	1.	Provoke route error				
		OR	1.	Create arbitrary RERR messages			
			2.	Do not send ACK messages (causing RERRs in other nodes)			
		2.	Do not participate in following route request (A.1)				
	3.	Do not relay data packets					
	OR	1.	Drop data packets (case B)				
		2.	Set hop limit/TTL to 0/1 (causing a RERR)				

Table provides details about simulation parameters.

Table 2: Simulation parameters

Parameter	Value
Number of Nodes	50
Area X (m)	1500
Area Y (m)	300
Traffic Model	cbr
Sending rate (packets/s)	4.0
Max. number of connections	20
Packetsize (byte)	512
Simulationtime (s)	900

The rate of delivery decreases as the network generally becomes weaker. When movement speed grows [2].

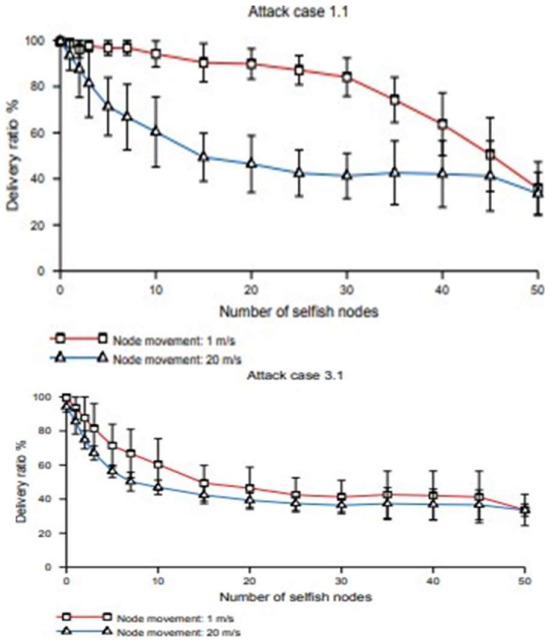

Figure 1: Selfish attack simulation

This study develops a Token network (MANETs) which is usually utilized for various Co-operative Bait-Detection systems in Ring-based applications such as military crises operations and emergency node attack identification in MANET. The token activities for preparation and response [3]

NODE ATTACKS

A mobile network is an infrastructure which does not have networks controlled by a centralized company. The network nodes are separate from their needs. For the route via the middle node, the cell nodes in the neighborhood are accountable. Not only do these transmitters endanger the safety of the network, they also increase the load on mobile nodes. The needs and behavior of each node must be followed in order to improve connection efficiency [4].

A) Malicious Node

B) Selfish Node: -The selfish node is one of the popular types of anonymous malicious node. A node supporting egotist conduct does not pass the data or services to other nodes

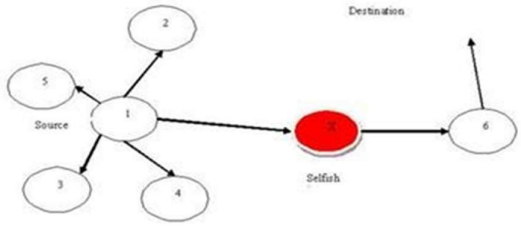

Figure 2: Selfish Node

Node Detection Token based Method

The node sent the packet RREQ and sets the current location of the neighbour. The status is displayed as red when RREQ network packets are sent across the node, thus the node can't pass and the Node is an auto-ego node. The procedure takes place at each route node up to the definition of the whole safe path. Table 3 provides the functionality for creating paths utilizing a token-based method [5].

Table 3: Algorithm for Token based Method

1. Set the source and destination for path generation
2. Destination D set the umpire node for observation
3. Umpire node forward list of neighbors to previous node
4. Previous node tally the list respective to own list and identify the interaction with received neighbor list
5. Perform interaction analysis to identify next umpire
6. The neighbor list is transferred to the adjacent umpire
7. Each time the interaction analysis is done to identify the effective neighbor
8. The process is repeated till the source node not occurs and the path is not formed.

Agent based Method

Table 4 contains a method for an actor to identify selfish nodes.

1. Define the centralized controller node
2. Distribute the k agents in the network
3. Each agent identify the nodes in the coverage
4. Share the routing table amount the cover nodes
5. Find the current status of node
6. Observe the load and loss rate for each node
7. Apply threshold limit to identify the selfish and safe node
8. Agent will exclude the selfish node
9. Connect with other agents to generate the safe path

Table 4: Algorithm for Agent based Method

Table 4 provides the agent-based auto-node identification and route creation method. The method shows that a controller node in the network distributes agents. Every agent has its own coverage area [6].

4.3 TOKEN RING CO-OPERATIVE BAIT DETECTION

A star topology that connects nodes to the circular center of your token may solve this issue.

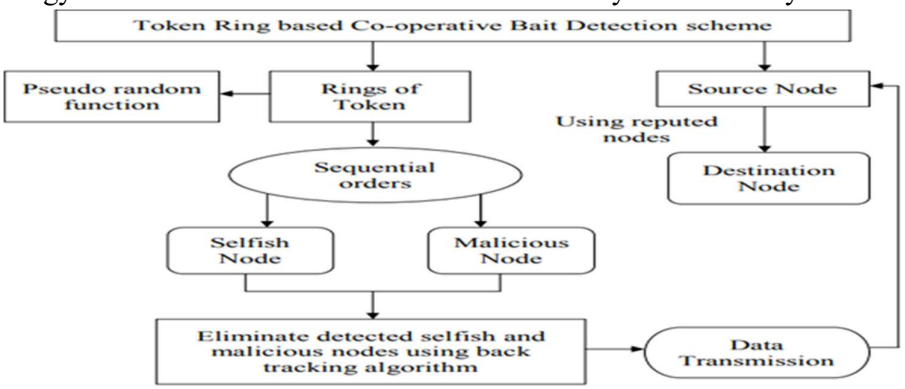

Figure 3: Architecture of proposed Token Ring based Co-operative Bait Detection

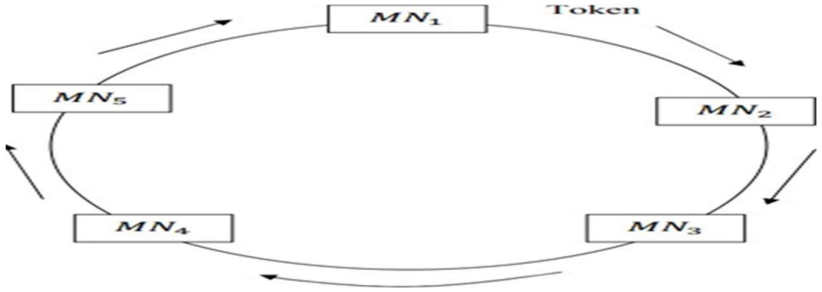

Figure 4: Token ring distribution among mobile nodes

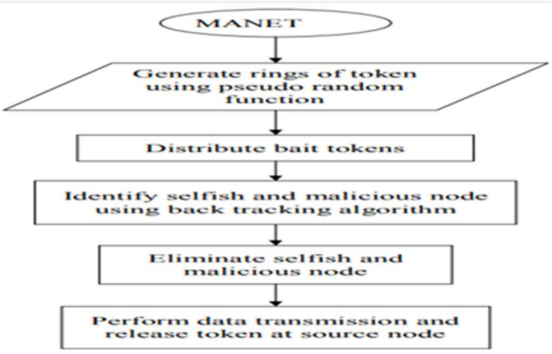

Figure 5: Flow diagram of proposed TR-CBD method

MANET consists of three stages in the TR-CBD technique. First, the token ring is produced using a pseudo-random function and then distributed by the suggested TR-CBD technique among the surrounding nodes [7].

Input: Mobile nodes '$\{MN_1, MN_2 \ldots\ldots, MN_n\}$', Source node '$SN$', Destination Node '$DN$', Ring of tokens
Output: Detecting selfish and malicious node in MANET
Step 1: Begin
Step 2: For all nodes 'MN_i' in MANET
Step 3: Generate ring of tokens with the help of pseudo random function
Step 4: Distribute bait tokens to the neighbour nodes including selfish and malicious nodes with RREQ packets
Step 5: Identify selfish and malicious nodes based on their behaviour by using back tracking algorithm
Step 6: Eliminate the detected selfish and malicious nodes
Step 7: Perform data transmission using reputed mobile nodes
Step 8: Release the token at 'SN' when the data is received at 'DN'
Step 9: Continuously distribute the tokens in the ring of mobile nodes
Step 10: End for
Step 11: End

Figure 6: Algorithm for proposed Token Ring based Co-operative Bait Detection

EXPERIMENTAL EVALUATION

The TR-CBD method is proposed and implemented with the help of NS2 simulator. Simulations are conducted with 500 mobile nodes which are placed in the square area of 1500 m * 1500 m. The selfish node and collaborative attacks are detected using DSR protocol. Table 5 lists various parameters used in the simulation [8].

Table 5: Simulation parameters

Parameter	Value
Simulation area	1500 m * 1500 m
Number of mobile nodes	50 to 500
Number of data packets	10 to 100
Simulation time	70 ms
Traffic model	Constant Bit Rate (CBR)
Mobility model	Random Way Point
Node speed	0 – 20 m/s
Routing protocol	DSR
Pause time	500 s
Data rate	20 kbps

With the following measures such as overhead routing, throughput rate, time for intrusion detection, and intrusion detection rate, the performance of the proposed TR-CBD technique will be evaluated as follows:

RESULT, ANALYSIS AND DISCUSSION

The suggested Cooperative Bait Detection (TR-CBD) technique, created by Prachi Arya and Token Based Umpiring Techniques (TBUT), developed by Jebakumar Mohan Singh Pappaji Josh Kumar, is compared to the method of co-operative bait Detection Scheme established by the TR-CBDS. Tables and graphs are used to demonstrate that the TR-CBD technique is superior than the other state-of-the-art systems [9].

Performance of Routing Overhead

Table 6: Measurement of Routing Overhead

Number of data packets	Routing overhead (%)		
	Existing CBDS	Existing TBUT	Proposed TR-CBD
10	24	27	20
20	27	31	24
30	29	32	26
40	31	34	29
50	34	37	32
60	37	39	34
70	39	42	36
80	40	43	37
90	42	45	40
100	44	47	42

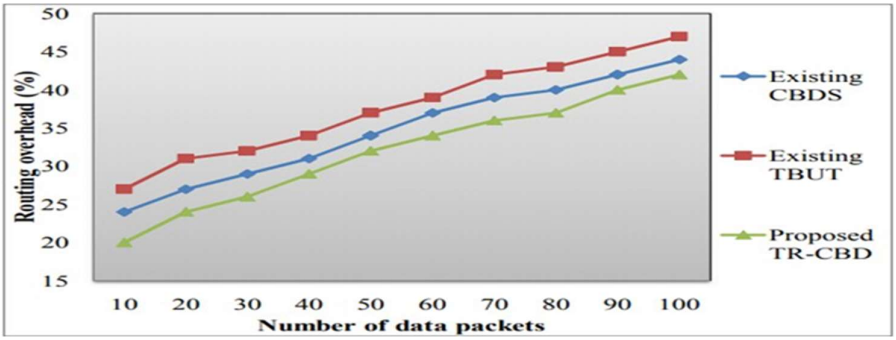

Figure 7: Estimation of routing overhead

Function of Throughput Rate

The rate of performance is expressed as follows mathematically.

$$T = \frac{DP_r}{DP_s} * 100$$

From equation, throughput 'T' is computed in terms of percentage (%). In case of high throughput rate, more efficiency is achieved

Table 7: Calculation of throughput rate

Number of data packets	Rate of throughput rate (%)		
	Existing CBDS	Existing TBUT	Proposed TR-CBD
10	66	61	70
20	67	63	72
30	69	64	73
40	70	65	75
50	71	67	76
60	73	68	78
70	74	70	79
80	76	71	82
90	77	72	83
100	80	74	85

Table 7 indicates throughput rate with the number of data packets for the proposed TR-CBD method and the existing CBDS and TBUT methods.

10 to 100 number of packets has been considered as input for conducting test [10].

Table 8 shows that throughput is also increased for all the methods along with increasing the number of data packets.

However, proposed TR-CBD method significantly enhances the throughput rate when compared to existing methods CBDS and TBUT.

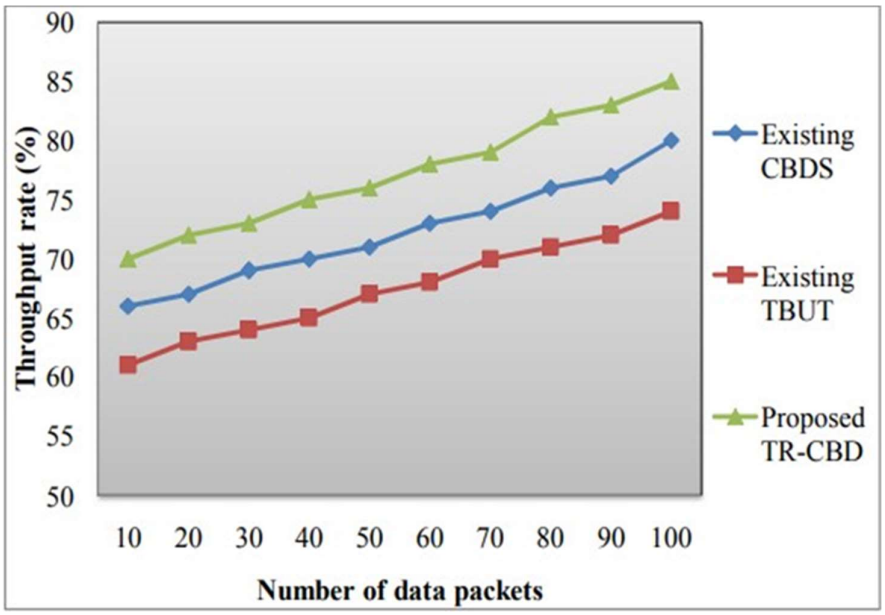

Figure 8 Computation of throughput rate Analysis of Execution Time

Execution time is defined as the amount of time taken to perform the intruder nodes detection in MANET with respect to the number of mobile nodes. Execution time for intruder nodes detection is mathematically computed as follows.

$$ET = number\ of\ mobile\ nodes * Time\ (intruder\ nodes\ detection)$$

From equation, execution time 'ET' is measured in terms of milliseconds (ms). In case of low execution time, more efficiency is achieved [11].

Table 8: Comparison of execution time

Number of mobile nodes	Execution time (ms)		
	Existing CBDS	Existing TBUT	Proposed TR-CBD
50	21.5	24.1	18.6
100	23.6	26.2	20.3
150	24.5	27.9	22.5
200	26.9	29.1	23.9
250	27.3	30.3	25.4
300	28.6	31.4	26.7
350	30.2	33.2	28.2
400	32.4	34.9	30.1
450	34.8	36.8	32.5
500	36.1	38.7	33.2

Figure 9: Assessment of execution time Function of Intrusion Detection Rate

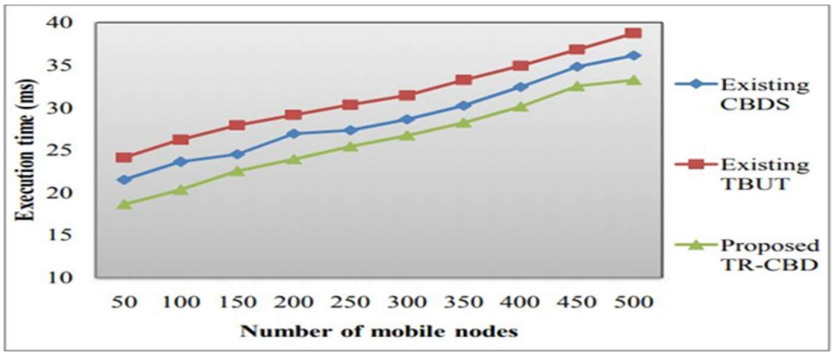

Table 4.9 Calculation of intrusion detection rate

Mobile nodes	Rate of Intrusion detection (%)		
	Existing CBDS	Existing TBUT	Proposed TR-CBD
50	64	58	69
100	67	61	72
150	69	63	74
200	71	66	75
250	72	67	78
300	74	70	81
350	77	71	82
400	78	72	84
450	80	74	87
500	83	76	88

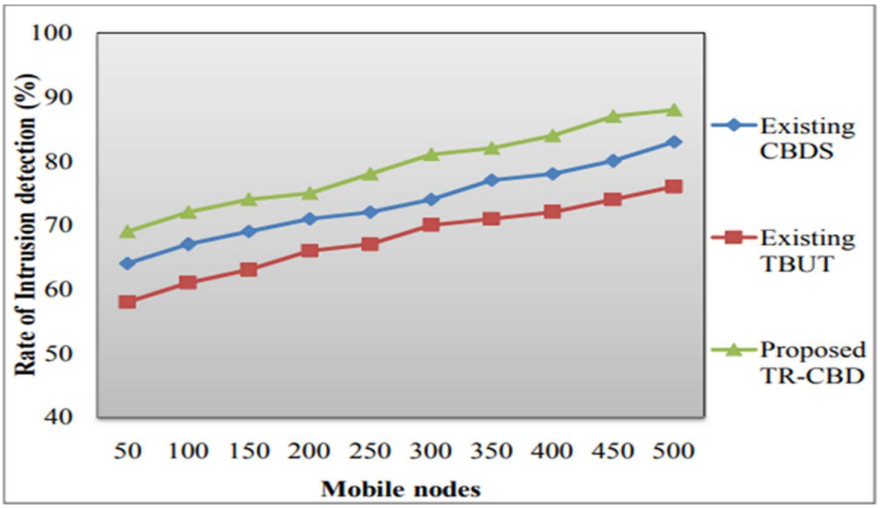

Figure 10 Computation of intrusion detection rate

ELLIPTIC CLEY GENERATION ADAPTIVE INFORMATION (IG-SAEKG)

PKC uses mathematical functions that are so-called uniform or simple to compute, but the opposite is rather complex for them to calculate. Two simple examples I'll send you:

Hash Functions: Irreversible mathematical transformation is used to "encrypt" information, providing a digital fingerprint, which is mainly used for message purity and integrity.

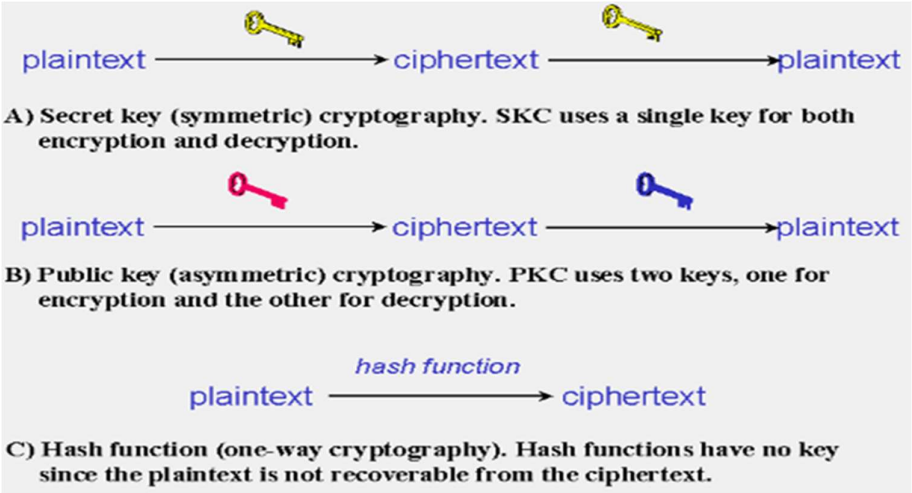

Figure 11: Types of cryptography

Proposed ECDSA Processor Design

Input: message m, domain parameters $(a, b, n, G = (x_G, y_G) \in E)$

Output: private key d, public key Q

1. Choice of elliptic curve $E(a, b)$.
2. Choice of a point $G(x_G, y_G) \in E(a, b)$ of order n.
3. Choice of a big integer d, with $1 \leq d \leq n$.
4. Choice of a point $Q(x_Q, y_Q) = d.G$ (the Montgomery scalar multiplication).

Return private key d and public key Q.

Algorithm private and public key generation

An entity "A" executes the Algorithms steps with chosen domain parameters while signing a message "m." The scalar multiplication and the hash function are based on this.

Input: private key d, message m, domain parameters $(n, G(x_G, y_G))$, public key $Q(x_Q, y_Q)$

Output: signature (r, s)

1. Choice of a random integer k, with $1 \leq k \leq n-1$.
2. Calculate $kG = (x_1, y_1)$.
3. Calculate $r = x_1 \mod n$. If $r = 0$, so return to step 1.
4. Calculate $k^{-1} \mod n$.
5. Calculate $e = H(m)$ such that: $H(m)$ is cryptographic hash result using SHA-1 or SHA-2 of the message m.
6. Calculate $s = k^{-1}(e + d.r) \mod n$. If $s = 0$, return to step 1.

Return (r, s) the signature of the message m

Algorithm ECDSA signature generation

Algorithm shows the recipient's signature verification by computing the hash function to digest the message, then utilizing the sender's public key on the message.

Input: a signature (r,s), $Q(x_Q, y_Q)$ public key, domain parameters $(a, b, G(x_G, y_G), n)$, message m

Output: signature verification or rejection

1. Verify that integer r and s are both in $[1, n-1]$.
2. Calculate $e = H(m)$ such that: $H(m)$ is cryptographic hash result using SHA-1 or SHA-2 of the message m.
3. Calculate $w = s^{-1} \mod n$.
4. Calculate $u_1 = ew \mod n$ and $u_2 = rw \mod n$.
5. Calculate $X = u_1 G + u_2 Q$. (using the point addition formula on the elliptic curve).
6. If $X = 0$, so signature will be rejected. Else, calculate $v = x_1 \mod n$.
7. Signature will be accepted only if $v = r$.

Algorithm ECDSA signature verification

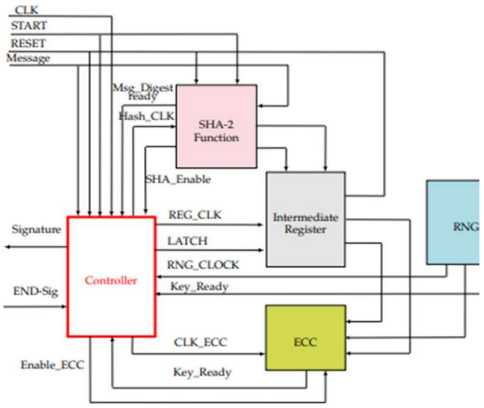

Figure 12: Proposed ECDSA Architecture

4.6.4 Security Analyses of ECDSA Processor

- Fault Injection Attack
- Restart Attack

Identity-based Cryptosystem

In 1984 Shamir developed a version of conventional CA-PKC/PKI called Identification Cryptography (IBC) which allows users to utilize their identity such as their e-mail address, IP address, etc. This version was introduced [12]

One-way Hash Function (OWHF)

Suppose, H(•) is a one-way haash function, which requires an input string x of a length variable and a fixed-length y, the hash value, to alter the haash value of the output, accidentally or

intentionally. Generally the safe cryptographic OWHF, MD4, MD5, SHA-1, etc.., has the following characteristics:

- The Hash value H(m) for a particular message is quite simple to calculate.
- Altering a message without changing the hash value H is possible (m).
- A message with a certain hathh H (m), called as preimage resistance, may be generated.
- An additional inputm2, which1 stated as H(m1) = H should be very hard to locate in given aninputm12 (m2). This characteristic is known as a weak resistance to collision.
- Two alternative messages, sqm and sqm (H(m1) =H) are possible (m2). The pair is known as a hash collision using cryptography. This characteristic is known as a high resistance to collision.

SELF ADAPTIVE ELLIPTICAL KEY GENERATION (SA-EKG) ALGORITHM

The suggested HECC message is coded using a symmetrical code and encrypted with the asymmetric coding method. The key above the coded message. Now the creator takes the data key and decodes it to the public key of the receiver using the asymmetrical coding method. The operation is a crucial component. But the benefit is that the amount of the data key that we take is very computed, so in just a fraction of a second the entire operation is done. The maker

collects the data block and block key in a file and transfers it to the receiver. In two operating phases, the new Hybrid ECC (HECC) is described in the following subparts:

STAGE 1

The author and data recipient must first accept both ECC parameters, i.e. domain parameters of the scheme [13]. In the primary instance, the area of the ECC domain is p and c and'd' in the binary case. The elliptical curve is characterized by the variables "a" and "b" in its defining equation. The elliptical curve is now a fundamental plane curve spread across a limited region. It consists of points which comply with the equation.

$$y^2 = x^3 + ax + b$$

STAGE 2

Generator G defines the single unit monogenous group. Certain logarithm procedures rely on the restored (Zp)x with an elliptical curve. Here we choose the curve Elliptical Diffie-Hellman uses the main agreement mechanism of Diffie Hellman (Diffie–Hellman, 1976).

For G, the lowest unfeatured number n is in the majority of cases a direct number for crypothetical implementations, namely, nG. The number h may be viewed as the integer, as n is the size of a subgroup E(Fp). The h in eqn is visible

$$h = \frac{|E(F_p)|}{n}$$

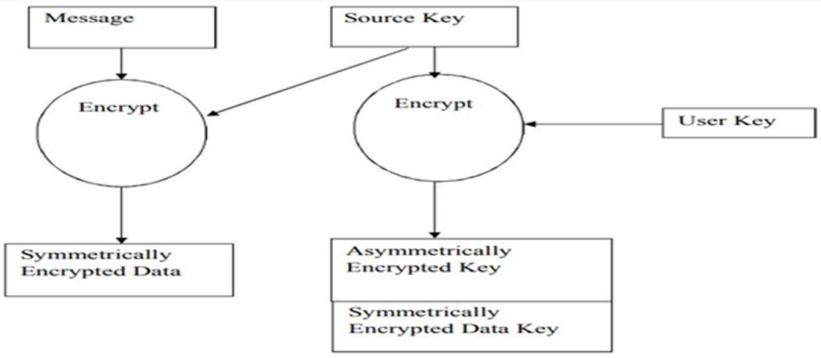

Figure 13: Proposed hybrid ECC

ALGORITHM

Step 1: Begin

Step 2 : Initialization: Both the parties involved in data transmission as sender and receiver must agree on ECC components and define an elliptical curve

Step 3: ECC='p' in case of a binary 'm' and 'f'

Step 4: Constants in elliptic curve definition are a and b

$$y^2 = x^3 + ax + b$$

Step 5: The elliptical curve becomes a basic plane curve, extending across a limited field. It comprises locations that meet the equation referred to in step 4

Step 6: The order of G is the lowest non negative integer n for cryptographical applications, such as nG= mit, which is usually prime.

Step 7: n is the size of an E(Fp) subgroup, according to the Lagrange theorem, the value h is an integer. The h can be expressed through this equation

$$h = \frac{|E(F_p)|}{n}$$

Where (h≤4) and, preferably, h=1

Step 8: Thus the most commonly used parameters are (p, a, b, G, n, h) and in the binary case (m, f, a, b, G, n, h).

Step 9: End

RESULTS AND DISCUSSION

In NS2, the renowned tool for event networking simulation, the suggested work is carried out. Experimental findings comparing HECC's memory and runtime needs with current procedures are obtained. The comparison of runtime and memory requirements shown in Fig. 4.14 and 4.15 correspondingly:

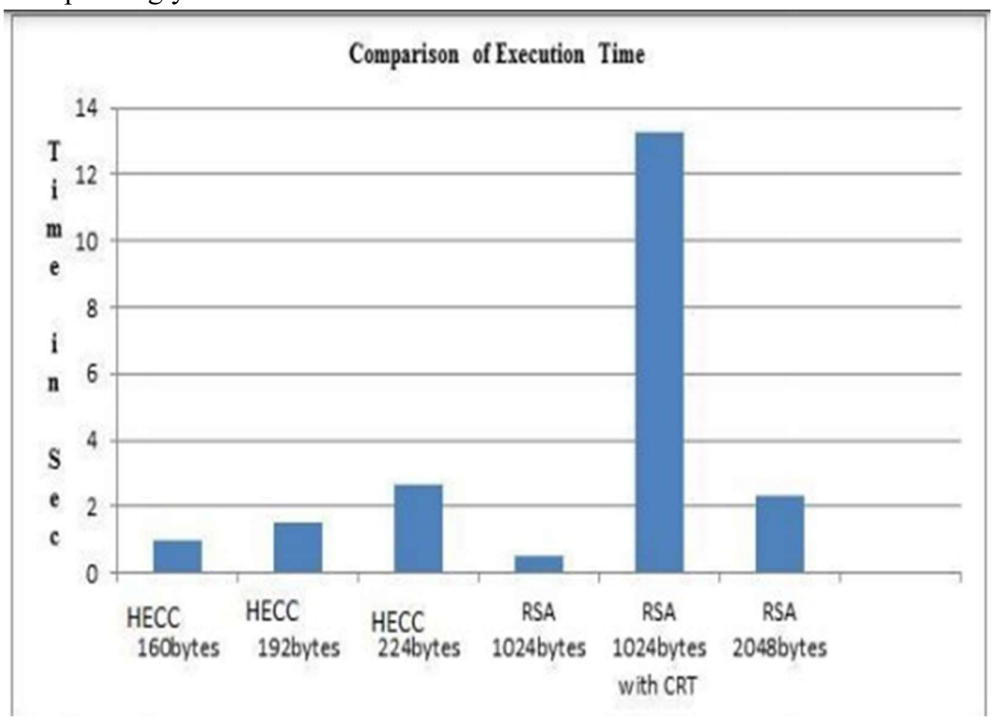

Fig. 14: Execution time for performance analysis

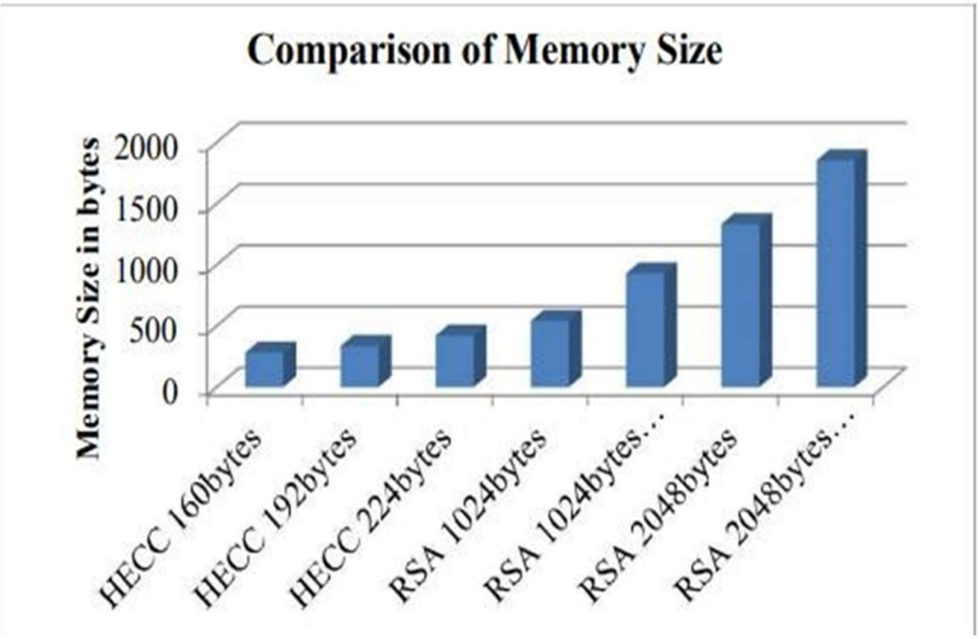

Fig 15: Memory requirements for performance analysis

The performance of the proposed method for a maximum of 200 nodes is compared with current algorithms. Tables 4.10 and 4.11 show the results of different storage space and consumption techniques.

Table 10 Storage Space Required (in MB)

No of Nodes	RSA	PKC	RSA-CRT	ECC	HECC
20	0.2	0.1	0.15	0.14	0.12
40	0.26	0.13	0.2	0.18	0.16
60	0.34	0.17	0.25	0.23	0.20
80	0.42	0.21	0.32	0.29	0.26
100	0.53	0.26	0.4	0.36	0.32
120	0.66	0.33	0.5	0.45	0.41
140	0.79	0.4	0.59	0.53	0.48
160	0.95	0.48	0.71	0.64	0.58
180	1.14	0.57	0.86	0.77	0.70
200	1.25	0.63	0.94	0.85	0.76

Table 11 Time Consumption (in ms)

No of Nodes	RSA	PKC	RSA-CRT	ECC	HECC
20	273.8	412.3	102.7	93.5	84.1
40	350.46	527.74	126.32	115.0	103.5
60	448.59	675.51	155.37	141.4	127.2
80	574.2	864.66	191.11	173.9	156.5
100	734.98	1106.8	235.07	213.9	192.5
120	940.77	1416.7	289.13	263.1	236.8
140	1204.2	1813.3	355.63	323.6	291.3
160	1541.4	2321	437.43	398.1	358.3
180	1972.9	2970.9	538.04	489.6	440.7
200	2525.4	3802.8	661.78	602.2	542.0

Current and suggested methodology are presented in Table 4.12 and Table 4.13 as time consumption in the mobile and MANET nodes.

Table 12 Time taken in mobile node for Key Exchange (in bits)

No of Nodes	RSA	PKC	RSA-CRT	ECC	HECC
160	3	4	3	3	2
256	7	11	8	7	7
512	16	18	15	14	12
1024	22	28	19	17	16
2048	166	178	122	111	100

Table 13: Time taken over MANET for key exchange (in bits)

No of Nodes	RSA	PKC	RSA-CRT	ECC	HECC
160	1.2	1.8	1.1	1.0	0.9
256	4.3	5.2	4.1	3.7	3.3
512	7.4	8.6	7.0	6.4	5.7
1024	10.4	12.0	10.0	9.1	8.2
2048	13.5	15.5	12.9	11.7	10.6

It is noted from the findings that, compared with current techniques, the suggested HECC produces optimum performance time and memory demands. For seamless, speedy and secured functions, the portable devices approaching MANET will make optimum use of the HECC. In many discrete mathematical issues in engineering there are additional difficulties to develop optimum solutions [14].

CONCLUSION:

These parameters are common to MANET simulations and are utilized for every subsequent simulation. These simulations are shown in Figure. The number of selfish nodes has changed between 0 and 50 (the total number of nodes in the network). It is clear that the rate of packets delivered correctly in the network has a major impact. The movement rate also has a noticeable impact. The quicker the nodes travel, the lower the supply ratio. Lastly, we find that case B nodes are more damaging to the network at lower speeds than type A, while there are no large differences at greater speeds [15].

REFERENCES:

1. Tarek Sheltami, Abdulsalam Basabaa & Elhadi Shakshuki 2014, 'A3ACKs: "Adaptive three Acknowledgments intrusion detection system for MANETs", Springer, Journal of Ambient Intelligence and Humanized Computing, vol. 5, no. 4, pp.611-620.
2. Umang, S, Reddy, B V R & Hoda, M N 2010, "Enhanced intrusion detection system for malicious node detection in ad hoc routing protocols using minimal energy consumption", IET Communications, vol. 4, no. 17, pp.2084-2094.
3. Umesh Kumar Singh, Kailash Phuleria, Shailja Sharma& Goswami DN2014, "A Comparative study of Collaborative Attacks on Mobile Ad-Hoc Networks", International Journal of Emerging Technology and Advanced Engineering, vol. 4, no. 8, pp.183-187.
4. Vishnu Balan, E, Priyan, M K, Gokulnath, C & Usha Devi, G 2015, "Fuzzy Based Intrusion Detection Systems in MANET", Journal of Scientific Research, vol. 50, pp.109-114.
5. Yang Yang 2014, "Broadcast encryption based non-interactive key distribution in MANETs", Elsevier, Journal of Computer and System Sciences, vol. 80, no. 3, pp.533-545.
6. Yongguang Zhang, Wenke Lee & Yi-An Huang 2003, "Intrusion Detection Techniques for Mobile Wireless Networks", Journal Wireless Networks archive, vol. 9, no. 5, pp.1-16.
7. Zheng Yan, Peng Zhang & Teemupekka Virtanen 2011, "Trust evaluation based security solution in ad hoc networks", Proceedings of the Seventh Nordic Workshop on Secure IT Systems, pp. 1- 14
8. A.Agalya,C.Nandini, S. Sridevi, "DETECTING AND PREVENTING BLACK HOLE ATTACKS IN MANETS USING CBDS (Cooperative Bait Detection Scheme)‖" , International Journal of Modern Trends in Engineering and Research (IJMTER), Volume 02, Issue 04, [2015].
9. Akshita Rana,Deepak shrivastava, "A defending of wormhole attack in wireless mesh network based on epigraph relay method and cooperative threading technique‖", International Journal of

Advanced Research in Computer Engineering & Technology (IJARCET)Volume 1, Issue 9, November 2012

10. Manjeet Singh, Apurva Sharma, "Security in MANET Using ECBDS on Resource Consumption Attack and Byzantine Attack‖", IJITKM Volume 8 2015 pp. 4-7.

11. C.Krishna Priya1, Prof.B.Satyanarayana, "A REVIEW ON EFFICIENT KEY MANAGEMENT SCHEMES FOR SECURE ROUTING IN MOBILE AD HOC NETWORKS‖" , International Journal of Computer Engineering and Applications, Volume V, Issue I, Jan 14.

12. Anshika Garg, Shweta Sharma, "A Study on Wormhole Attack in MANET‖", International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882 Volume 3 Issue 2, May 2014.

13. Muskan Sharma, Chander Prabha ,‖ "Combating Resource Consumption and Byzantine Attacks in MANET through Enhanced CBDS Technique‖", American International Journal of Research in Science, Technology, Engineering & Mathematics AIJRSTEM 14-543; © 2014.

14. C. Deepika Shiny *, I. Muthumani, — "Detection and Recovery of Packet Drop under Network Layer Attack in MANET‖", International Conference on Electrical, Information and Communication Technology, 28 February 2015.

15. Aditya Bakshi, A.K.Sharma, Atul Mishra, "Significance of Mobile AD-HOC Networks (MANETS)" , International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-4, March 2013.