International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-11 Issue-3, September 2022

Clone Node Detection in Heterogeneous WSN with Low Memory Overhead

Sajitha M, D Kavitha, P Chenna Reddy

Abstract: ***In unattended areas, the wireless sensor network is deployed and the nodes are always open to attacks. An adversary can capture a node and can deploy many nodes, which are clone of the captured, in the network called clone node or replicated node by using the credential information retrieved from captured node. These clone nodes can damage the network directly or indirectly. This attack is called node replication o attack or clone node attack. In this area, so many works are introduced and all of these methods use a random key or code, or location information, to detect clone nodes. This paper presents a method that does not use any of this information. The simulation results show that it performs better than previous methods.***

Keywords: Clone Node or Replicated Node, Computational Overhead, Detection Probability, Storage Overhead, WSN

I. INTRODUCTION

A set of nodes is deployed randomly in an area and these nodes form a network called wireless sensor network. These nodes collect information from their sensing area and pass this information to the network node called sink node. Sink node is the only node capable of communicating with the outside world. The size of sensor nodes is too small and has limited data processing, computing and memory capacity. Each sensor node has a sensing and transmission range which varies for different type of nodes. The wsn has many applications including battlefield monitoring, machine monitoring, industrial and consumer monitoring [1].

Several types of attacks affect WSN. The attacks can be classified into two–layer-dependent and layer-independent [2] [3]. An attack may occur from outside the network and inside the network. External attackers are not part of the network, but internal attackers are residing within the network and attacking the network with analyzing the network. One of the layer-independent attacks is clone attack or node replication attack. Sensor node hardware is not resistant to manipulation. The attacker physically captures a sensor node and extracts data such as node id, replicates more nodes with this credential information, and returns to the network. These clone nodes can cause lots of network damage. Clone nodes can affect the integrity of the network as well as the data transmitted. Two types of methods are already being developed to detect clone nodes-centralized and decentralized. Centralized method is not commonly used due to a single center failure. Here the base station is given full responsibility. The base station or sink collects information from the network and processes this information to identify clone nodes. So the entire detection technique depends on the base station. If the base station fails, the entire detection technique fails. The next section deals with literature survey followed by deployment in section 3. The algorithm and probability analysis are described in section 4 and section 5. Simulation and results are included in section 6. Conclusion and future works are described in section 7.

II. LITERATURE SURVEY

Parno et al [5] described a centralized detection technique where each node sends a message with its neighbors and its location to sink and sink detects the replicated node based on this information. Brook et al [6] proposed a scheme based on random key pre distribution and bloom filter. In this method a node randomly choose one key from the set of keys and uses bloom filter to collect the key usage statistics. The usage time of key is restricted within a threshold and if it exceeds it becomes a clone node. It is also a centralized detection method. The main drawback of this method is high false negative and false positive.

A distributed detection method is introduced by Parno et al[4] called deterministic multicast(DM). Here witness nodes have the responsibility of detecting clone node. Nodes send its location claim to witness node, if witness node receives two location claims with same id and different locations is considered as clone nodes. The witness nodes are fixed in DM, so an adversary can attack witness node also. Parno et al [5] also proposed two more detection algorithms RM and LSM. In RM the witness nodes are randomly selected and location claims are sent to these randomly selected nodes. In LSM each forwarding node stores the location claims so the detection is faster compared to RM. Zhu et al [7] proposed two variations of LSM: SDC and P-MPC. The network is considered as cell in a grid and each node is associated in any of the cell. The broadcasted location claim is used for verifying the validity of the signature. Here every node decides whether to pass this claim. This algorithm uses geographic hash function to identify destination cell. After reaching the destination cell, the sensor receiving it verifies its legitimacy. This method improves detection probability over RM and LSM. But it is also location dependent. M. Conti et al [8] introduced RED which executes routinely at fixed intervals of time.

Manuscript received on 09 July 2022 | Revised Manuscript received on 16 July 2022 | Manuscript Accepted on 15 September 2022 | Manuscript published on 30 September 2022.
***** Correspondence Author
Sajitha M*, Department of CSE, JNTU Anantapur, Ananthapuramu, India. Email: sajitha139@gmail.com
D Kavitha, Department of CSE, G Pulla Reddy Engineering College, Kurnool, India. Email: dwaramkavithareddy@gmail.com
P Chenna Reddy, Department of CSE, JNTU Anantapur, Ananthapuramu, India. Email: pcreddy1@gmail.com

Every run of the protocol consists of two steps. In first step, a random value, rand, is shared between all the nodes through base station. The second step is called detection phase. In the detection phase, each node broadcasts its claim (ID and location) to its neighbouring nodes. The witness nodes will be different at different phases of execution. This method requires less memory and has high detection rate. This method also depends on location. A. K. Mishra [9] introduced a new method called zone based node replica detection scheme which divides the network into several zones and each zone has a zone-leader, which has the ability to detect replica nodes in the network. In this protocol detecting mechanism is done at two levels. The first level is called intra-zone detection and second is inter zone detection. When an intruder replicates the zone-leader, the wireless network will be compromised. Kwantae Cho et al [10] proposed solution uses neighboring node IDs, instead of location information, in order to detect replicas. Neighboring node IDs are presented with a constant size using a bloom filter. This solution used neighbors IDs to detect clone nodes. Neighboring node IDs are represented with a constant size using a bloom filter. The bloom filter output (BFO) act as a proof and is distributed to randomly selected nodes. The method Randomized Distributed Bloom filter using –Replica (RDB-R) scheme is efficient to a certain extent in which it is not depend on deployment strategy, so no need of attaching GPS. So which highly reduces the sensor node cost, provide comparatively better communication overhead. But still the detection level and memory overhead is a problem. Zhiming zhang et al. [11] describes a method called TDS with orthogonal code which does not require location information, but require more storage and computation cost for calculating orthogonal code. Chia Mu et al [12] Vandana Mohindu[13] describe entirely different technique based on sensing and authentication to detect clone nodes. H Choi et al [15] uses set operations such as intersection and union to detect clone nodes. It employs hierarchical structure to compute non overlapped set operations. The nodes are in two modes Ruled or Ruler. The set membership is used to detect clone nodes. K wantae Cho et al [16] classifies all clone detection in static and mobile WSN as centralized and distributed. Sathish et al[17] modifies RED with dynamic witness nodes. Farah et al [18] proposed MCD(Mobile Assisted Clone Detection) where mobile and static sensors are used simultaneously. Sachin Lalar et al [20] [21] explained a binary search tree based network and its clone detection. Here BST is constructed by using node ids. Resources are limited in WSN and need to be efficiently used as explained in [22][23][24]25].

III. PROPOSED METHOD

We propose a method, Low Computational overhead and Low Storage overhead (LCLS), to detect the clones in WSN. Here each cluster head is aware of nodes within the cluster. Say N is the number of nodes in a cluster. Cluster

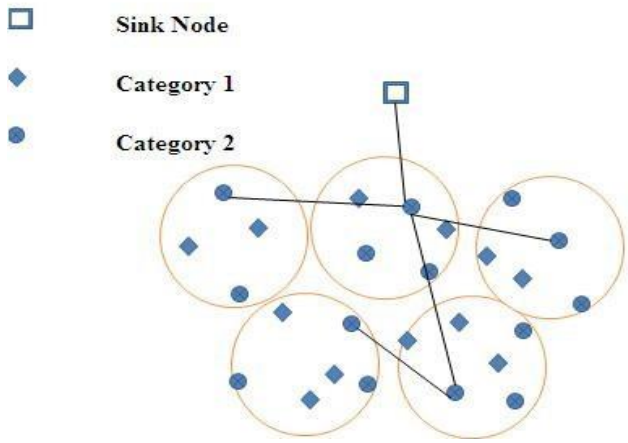

Fig. 1 Deployment of heterogeneous wireless sensor network

head stores a list of size N, the list is initialized to zero. The cluster head calculates i=id mod N and sets A[i] to A[i]+1. This calculation is done for all nodes. Then the total sum of all value of the array is equal to number of nodes in the cluster. The cluster head sends a hai message to all its nodes and all the nodes respond by sending a hello message. This hello message contains node id. Each cluster head have additional temporary list B[] of size N which is initialized to zero. After getting hello message, cluster head calculates j=id mod N and sets B[j] to B[j]+1. Cluster head compares A[] and B[] and if there is change then there exist clone node. If the difference occurs at ith position of the list, first identify the nodes which have been mapped to ith location of the list. The difference is below threshold Ti then the cluster head sends an alarm message indicating the suspicion of clone nodes to its entire neighboring cluster heads. If there is any other problem with these nodes in other cluster it alarms the network with node ids and these nodes are removed immediately. If the difference is above threshold Ti, it immediately removes that set of nodes and informs its neighbors about the suspected node ids.

AT CLUSTER HEAD- DURING DEPLOYMENT
Step 1. Initialize a list A and Sum_a to zero

A[] 0,Sum_a 0

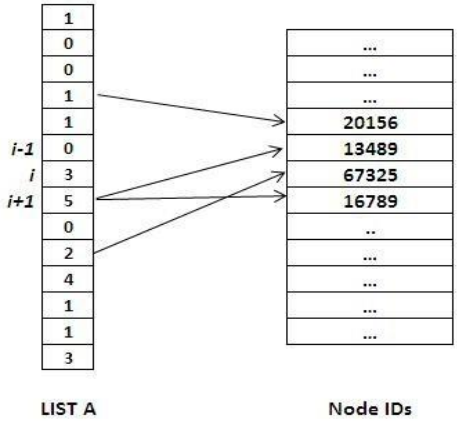

Fig. 2 Node id to list mapping

Retrieval Number: 100.1/ijrte.C72060911322
DOI: 10.35940/ijrte.C7206.0911322
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Step 2. Identifies the nodes in the cluster and id mod N
Step 3. A[i] A[i]+1
Step 4. If receives a sleep signal from node k j k mod N
A[j] A[j]-1
Step 5. Repeat step 2 – 4 for all nodes
Step 6: For i 0 to N-1
$Sum_a = Sum_a + A[i]$

AT CLUSTER HEAD- FOR DETECTION
Step 1: Set B[] 0, Sum_b 0
Step 2: For each node k in the cluster
Step 3: j k mod N
increment B[j]
Step 4: For i 0 to N-1
$Sum_b = Sum_b + B[i]$ Step 5: If Sum_b 6= Sum_a
For i 0 to N-1
If A[i] 6= B[i]
If B[i] –A[i] < T_i send an alert message to all clusters
Else
Discard all nodes that mapped to A[i] from the net-work.

This proposed method only requires a simple list and mapping function to detect clone nodes. It does not require any complex calculation. In this method there is no need of any extra code or any other data structure such as bloom filter to identify the clone node. In the previous works they stored a secret key or code for each node or otherwise location information is required. These are not required here and hence this method does not require any location service.

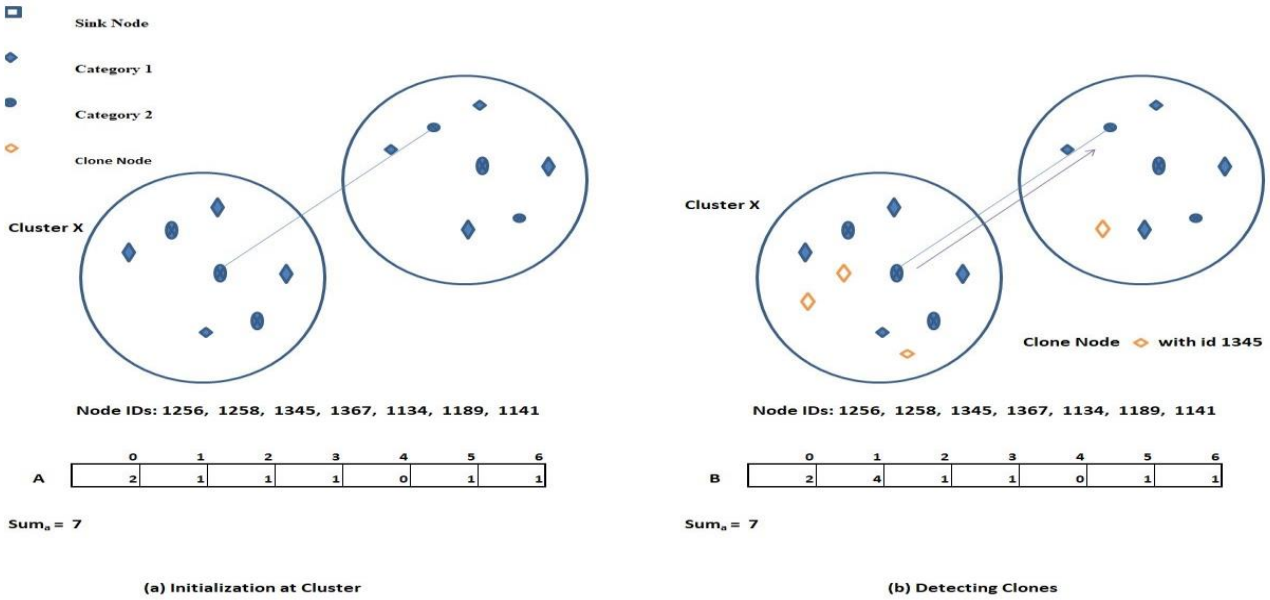

Fig. 3 Proposed Method

Figure 3 shows an instance of the LCLS algorithm. Cluster X, with seven nodes, initializes its list A and calculates Sum_a by collecting messages as shown in figure 3 (a). The cluster x calculates list B and compares each value with list A. when the difference between B and A exceeds a certain threshold, alert message is passed to neighboring clusters. The clusters remove all nodes mapped to that particular location. Node with ID 1345 is replicated in cluster x as shown in figure 3(b). The list A and B are compared and alert message is passed to neighboring clusters. The time complexity of this proposed LCLS algorithm is directly proportional to the size of list. The size of list is determined based on number of nodes in a cluster. So the complexity of computation is O (N) where N is the number of nodes in the cluster. This algorithm is executed in all clusters and C is the number of clusters in the network. Total computational complexity, T = O (C*N). The communication overhead of the proposed algorithm depends on the degree of a node. Once a cluster head identifies a clone node, this information is passed to all immediate cluster head. The cluster heads remove such nodes from its node list. The cost of communication and storage is compared in the table 3.1.

Table 1 Comparison of communication and storage cost

Algorithm)	Communication Cost	Storage Cost
RM[14]	O(n)	O(n)
LSM[14]	$O(\sqrt{n})$	$O(\sqrt{n})$
Random Key Distribution[10] RED[17]]	$O(gpdn\sqrt{n})$O(log n)	$O(gpd)$O(k)
TDS[12]	O(d)	O(d+M)
Proposed -LCLS	O(d)	O(N+d)

4.1 Probability of Hash Collision

Given k randomly generated values, where each value is a non negative integer less than M. We want to find the probability that at least two of them are equal. We can calculate it in a reverse manner, by finding the probability that is all equal. This can be subtracted from 1; we will get answer to the original question. Given a space for M possible hash values and we have picked one value. Then there are M-1 possible values that are unique from the value we picked. The probability of randomly generating two integers that are unique from each other is (M-1)/M. Generally, the probability *P* of randomly generating k integers that are all unique is

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

$$P = \frac{M-1}{M} * \frac{M-2}{M} \ldots\ldots \frac{M-(k-2)}{M} \quad (1)$$

For large k the above expression is approximately equal to $e^{\frac{-k(k-1)}{2M}}$ This expression gives probability that all are unique. So probability of collision is

$$P = 1 - e^{\frac{-k(k-1)}{2M}} \quad (2)$$

IV. RESULTS AND DISCUSSION

We considered a 500 x 500 area and deployed 200 category 2 nodes and 100 category 1 nodes. Only static sensors are considered. If the sensing and transmission range is set to .5m and each node can detect an area of 0.523m^2 $(Area\ of\ a\ Circle = \frac{2}{3}\prod r^2)$.

5.1 Clone Detection Probability

The clone nodes are deployed by the attacker in the network, which are to be identified and removed from the network. The clone nodes can introduce different types of problems or attacks in the network. The detection probability *P* is calculated in this proposed method as average of *P* in different execution.

$$P = \frac{Number\ of\ Clone\ nodes\ detected}{Total\ number\ of\ Clone\ nodes\ deployed} \quad (3)$$

It is found that the proposed method identifies clone nodes faster than TDS. The detection probability is compared in figure 4.

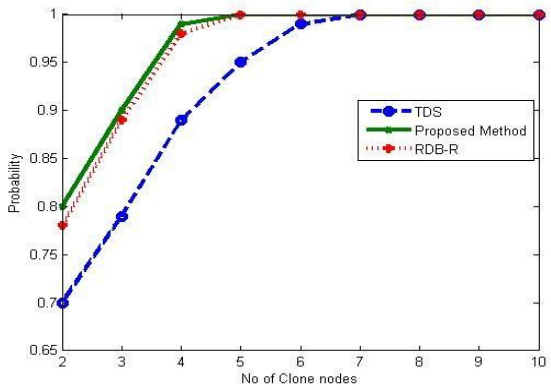

Fig. 4. Detection Probability

5.2 Storage Overhead

In this section, we compare the storage overhead with TDS. Storage overhead is calculated based on extra memory requirement of the algorithm, or extra code or information required. Storage requirement for each algorithm is very much important in wireless sensor network because of the constraint in the memory. The TDS method uses orthogonal matrix and we know that the orthogonal matrix is calculated recursively. Our proposed method uses a single list equal to the size of number of nodes. For a sensor network with 100 nodes, to generate orthogonal code, TDS need a matrix of 128x128. Thus, the total memory requirement is 128x128, which is 16,384bytes. Each node would store a code of size 128bytes. This code is used for communication. But LCLS doesn't require any code or extra data. Moreover, it consumes only N bytes at each cluster head where N is the size of cluster.

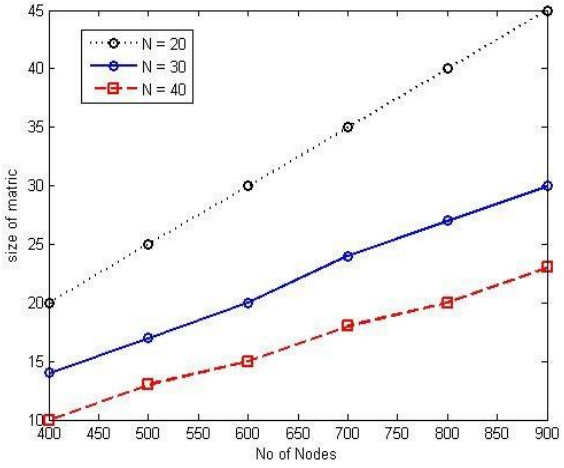

Fig. 5 Storage overhead

We simulated LCLS with different number of sensors and different cluster size. The number of sensors varied from 400 to 900 and average cluster size is taken as 20, 30 and 40. The result obtained is shown in figure 4. We can see that as cluster size increases the size of memory required per cluster head decreases.

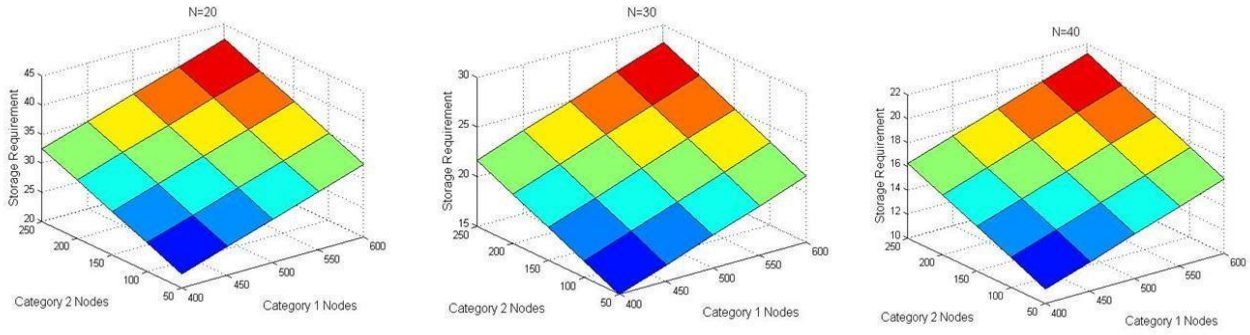

Fig. 6 Effect of change in number of nodes on storage overhead

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The figure 6 explains the extra storage space required by clusters with varying nodes. To investigate this, we have executed the simulation for 50 times with N=20, N=30 and N=40. The type 1 sensors are varied from 400 to 600 and type 2 nodes from 50 to 250. The storage overhead (in byte) at cluster head of the network is independent of type1 and type 2 nodes, but depends on the number of nodes per cluster.

5.3 Computational Overhead

The proposed method requires each cluster head to compute the list from the received messages from nodes. Figure 7 compares computational overhead of proposed method with RDB-R and TDS. The computational overhead is calculated with 500 total nodes and varied the number of clone nodes from 20 to 45. According to De Meulenaer et al. [14], the energy required per clock cycle is 0.0032 J.

$$E_{CP} = E * Clock\ Cycles \quad (4)$$

E_{CP} is computational overhead and E is energy required per clock cycle. Here in our proposed method each node to send its id to cluster head which require two bytes of data and 460 clock cycles are required to transmit one byte (Soderlund et al.[19]). E_{CP}= 2.994 J. This method requires only simple mode calculation which requires less computational overhead.

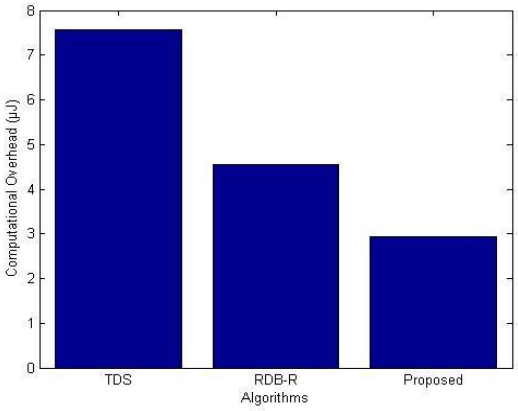

Fig. 7 Computational overhead comparison

V. CONCLUSION

In this paper, we introduced a novel method, LCLS, for detecting and removing clone nodes in WSN. The proposed method finds clone nodes in a distributed manner without using location information or any other extra code for nodes. LCLS method is simulated and results are compared with RDB-R [10] and TDS[11]. Extensive simulation is carried out and the results show that LCLS have high detection probability, less computational overhead and storage overhead. In future, the proposed method may be extended to find a clone node communicating with cluster nodes from outside the cluster areas.

REFERENCES

1. I.F. Akyildiz, *et al.,* " A Survey on Sensor Networks," in *IEEE Commun, Mag., vol. 40, no. 8,* , pp.102-114, Aug, 2002. [CrossRef]
2. J. Deng, et al., "A Survey on Sensor Networks," in security, privacy, and fault tolerance in wireless sensor networks. Artech House, August 2005.
3. T. Bonaci, *et al.,* " Distributed clone detection in wireless sensor networks: an optimization approach," in *Proceedings of the 2nd IEEE International Workshop on Data Security and Privacy in Wireless Networks (WoWMoM '11), Lucca, Italy,*, June 2011. [CrossRef]
4. W. T. Zhu *et al.,* "Detecting node replication attacks in wireless sensor networks: a survey, " in *: a survey, Journal of Network and Computer Applications,*, vol. 35, no. 3, pp. 1022–1034, 2012. [CrossRef]
5. B. Parno, *et al.,* "Distributed detection of node replication attacks in sensor networks, " in *In Security and Privacy, 2005 IEEE Symposium.* pages 49 -63, may 2005.
6. Brooks R, *et al.,* "On the detection of clones in sensor network using random key pre distribution, " in *IEEE Trans. Syst. Man. Cybern.* vol .37 No. 6, pp. 1246-125, 2007. [CrossRef]
7. Bio Zhu, *et al.,* "Localized multicast: efficient and distributed replica detection in largescale sensor networks, " in *IEEE Transactions on Mobile Computing,* Vo1. 9, No. 7, pp 913-926, July 2010. [CrossRef]
8. M. Conti, *et al.,* " A Distributed detection of clone attacks in wireless sensor networks, " in *IEEE Transactions on Dependable and Secure Computing,* vol. 8, no. 5,pp. 685-698, September/October 2011. [CrossRef]
9. A. K. Mishra and A. K. Turuk," A zone-based node replica detection scheme for wirelesssensor networks, " in *Springer, Wireless personal communications,* vol. 69, no. 2, pp. 601-621, 2013. [CrossRef]
10. Kwantae Cho, *et al.,* "A zone-based node replica detection scheme for wireless sensor networks, " *Low-Priced and Energy Efficient Detection of Replicas for Wireless Sensor Networks,* Vol. 11, NO. 5, September/October 2014.
11. Zhiming Zhang, *et al.,* " An efficient detection scheme of node replication attacks for wireless sensor networks, " in *International Journal of Security and Networks,* VOL. 10, NO. 4, MAY 2015. [CrossRef]
12. Chia-Mu Yu, *et al.,* "Compressed Sensing-Based Clone Identification in Sensor Networks, " in *IEEE Transactions on Wireless Communications,* VOL. 15, NO. 4, APRIL 2016. [CrossRef]
13. Vandana Mohindu, and Yashwant Singh," Node authentication algorithm for securingstatic wireless sensor networks from node clone attack, " in *International Journal of information and computer security,* Vol. 10, No 2/3, 2018. [CrossRef]
14. De Meulenaer, G.,*et al.,*"On the energy cost of communication and cryptography in wireless sensor network," in *IEEE international Conference on Wireless and Mobile Computing Networking and Communications,WIMOB'08,IEEE* October, pp. 580-585, 2008. [CrossRef]
15. H. Choi, S. Zhu, and T.F. L.Porta, "SET: Detecting clones in sensor networks," in *Proc. Security Privacy Communication Network, Workshops,* pp. 341-350, 2007. [CrossRef]
16. K wantae Cho, *et al.,*"Classification and experimental Analysis for Clone Detection Approaches in Wireless Sensor Networks," in *IEEE Sys. Journal,* vol 7, no. 1 Mar. 2013. [CrossRef]
17. Sathish R and Kumar D R,"Dynamic Detection of Clone Attacks in Wireless SensorNetworks," *International Conference on Communication Systems and Network Technologies(CSNT),* pp.501-505, 6-8 April 2013. [CrossRef]
18. K. Farah and L. Nabila,"The MCD Protocol for Securing Wireless Sensor Networksagainst Node Replication Attacks," *International Conference on Advanced Networking Distributed Systems and Applications,* Bejaia, pp. 58-63,2014. [CrossRef]
19. Soderlund, R., *et al.,*"Energy efficient authentication in wireless sensor networks, IEEE Conference on Emerging Technologies and Factory Automation," in *IEEE Conference on Emerging Technologies and Factory Automation, ETFA, IEEE,* September, pp. 14121416, 2007. [CrossRef]
20. Wendi Rabiner Heinzelman, *et al.,*"Energy- Efficient Communication Protocol for wireless micro sensor networks," in *33rd IEEE international Conference on system sciences,* 2000.
21. Sachin Lalar, *et al.,*"An efficient tree based clone detection scheme in wireless sensor networks," in *Journal of Information and Optimization Sciences,* vol. 40, pp. 1003-1023, 2019. [CrossRef]
22. Mohamed Elshrkawey and M. Elsayed Wahed,"An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks," *Journal of King Saud University- Computer and Information Sciences,* vol. 30 issue 2, pp. 259-267 April 2018. [CrossRef]

23. Bhupendra and Vidushi Sharma,"Energy efficient communication overhead algorithm inwireless sensor networks," *3rd IEEE International Advance Computing Conference*,2223 February, 2013. [CrossRef]
24. Segun O Olatinwo and Trudi H Joubert,"Efficient energy resource utilization in a wireless sensor system for monitoring water quality," *Journal on wireless Communication and Networking*,January 2019. [CrossRef]
25. Elhadi M. Shakshuki, *et al.*,"Resource Management Approach to an Efficient Wireless Sensor Networks," in *Elsevier Procedia Computer Science*, vol. 141, pp.190-198, 2018. [CrossRef]

AUTHORS PROFILE

Sajitha M., completed post-graduation in computer science and engineering from Bangalore University and pursuing PhD at Jawaharlal Nehru Technological University Anantapur, Ananthapuramu. Her area of interests is computer network and algorithm analysis.

D. Kavitha, obtained her B.Tech degree from S.K.University, Anantapur and M.Tech degree from Jawaharlal Nehru Technological University, Anantapur in the year 2001 and 2005 respectively. She pursued Ph.D at Sri Krishna Devaraya University, Anantapur, India. She is presently working as Professor in the Department of Computer Science and Engineering at G. Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India. She presented many research papers in international journals and conferences and her research areas include Computer Networks and Network Security.

P. Chenna Reddy, completed his masters, in computer science and engineering, and PhD from Jawaharlal Nehru Technological University, Hyderabad. He works as Professor, CSE Department, JNTU Anantapur. His area of interests covers computer network and computer programming. Dr. P. Chenna Reddy is active in research from 2004 onwards in the area of Computer Networks. He has 91 research publications in International Journals and 7 National Journals, in addition, 38 publications in international conference proceedings and 15 in National Conference proceedings. He has successfully supervised 13 Ph.D. theses in the area of Computer Networks. He worked as Director of Industrial Relations and Placements and School of Continuing and Distance Education, Director of Academic Audit and Director Skill Development Centre and Incubation Centre, JNT University Ananatapur, Ananthapuramu.