MOBILITY BASED ALGORITHMS FOR DETECTION OF CLONE NODES IN MOBILE SENSOR NETWORKS

[1]Benazir Banu, (M.E)., [2]Anitha Angayarkanni, (M.E)

Abstract – Sensor networks are composed of a number of sensor nodes with limited resources and used in application such as environment monitoring and object tracking. This allows a situation where the adversary can compromise one sensor node, fabricate many replicas having the same identity (ID) from the captured node, and place these replicas back into strategic positions in the network for further malicious activities. This is called clone node attack. The proposed system focuses on clone node attacks. Several clone node detection schemes have been proposed in the literature to defend against such attacks in static sensor networks. XED and EDD have been implemented for the detection of clone nodes. In XED and EDD, each node not only detects the replicas by its own effort, but also can revoke the replica in a communication-efficient way. XED algorithm have storage overhead of O(n). EDD algorithm's detection time increases based on the number of neighbor nodes. SED and BRSL algorithms are proposed to overcome the above mentioned drawbacks and to detect malicious anchor nodes. BRSL algorithm effectively distinguish malicious anchor node and increases detection accuracy. This algorithm improves localization accuracy.

Keywords- Clone Attack, XED (eXtremely Efficient Detection), EDD (Efficient Distributed Detection), BRSL (BRS- based Robust Secure Localized algorithm), SED(Storage Efficient Detection).

I. INTRODUCTION

A MOBILE WIRELESS SENSOR NETWORKS (MWSN) is a particular class of Wireless Sensor Networks (WSN) in which mobility plays a key role in the execution of application. Sensor nodes operate in hostile environments such as battle fields and surveillance zones. Due to their operating nature, MWSNs are often unattended, hence prone to several kinds of novel attacks. The proposed system aims to increase the detection accuracy of clone nodes and to reduce its influence in mobile sensor network. Mobile wireless sensor network is a class of WSN in which mobility plays a key role in the execution of the application. WSN deployments were never envisioned to be fully static, mobility was initially regarded as having several challenges that needed to overcome, including connectivity, coverage, and energy consumption, among others. Mobility enables sensor nodes to target and track moving phenomena such as chemical clouds, vehicles, and packages.

The adversary can compromise one sensor node, fabricate many replicas having the same identity (ID) from the captured node, and place these replicas back into strategic positions in the network for malicious activities. This is called node replication attack (or) clone attack. This node replication attack is the basis for launching a variety of attacks such as DoS attacks and Sybil attacks. If there are many replicated nodes, they can multiply the damage to the network. Therefore, the replicated nodes should be detected quickly. Due to the advances in robotics, mobile sensor networks have become feasible and applicable.

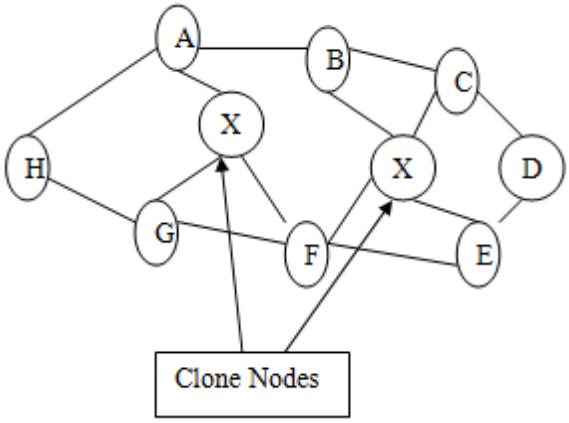

Fig 1: Clone Attack

II. RELATED WORKS

Replication attack detection protocols are classified into two categories of static WSN: Centralized and Distributed approaches. These approaches have their own merits and demerits. The main idea of these schemes is to have nodes report their location and claims that they identify their positions and attempts to detect conflicting reports that signal one node in multiple locations. This requires every node to sign and send a location claim, and verify and store the signed location claim of every other node. These protocols, except knowledge about deployment order, are not suitable for mobile WSN since location changes time to time. In static WSN, the centralized approach is simple, local detection (SET)[2] and with the context of random key pre-distribution and in mobile WSN[3], fast detection scheme with sequential probability ratio test (SPRT)[4] have been analyzed.

In SET: solutions rely on local detection using localized voting mechanism, a set of neighbors can agree on the replication of a given node that has been replicated within the neighborhood. However, this kind of method fails to detect replicated nodes that are not within the same neighborhood. SET manages to reduce the communication cost of the preceding approach by computing set operations of exclusive subsets in the network[2].

The basic idea with the context of random key pre-distribution is that, the keys that are present on the cloned nodes are detected by looking at how often they are used to authenticate nodes in the network. First each node makes a counting Bloom filter of the keys it uses to communicate with neighboring nodes and appends a nonce. Then Bloom filter and nonce are transferred to the base station, which will count the number of times that each key is used in the network[3].

In SRPT, technique have been proposed to detect replica attacks in mobile sensor networks. In static sensor networks, a sensor node can be considered to be replicated if it is placed at more than one location. However, if nodes are allowed to freely roam throughout the network, the above technique does not work because the mobile node's location will continuously changes as it moves. Hence, it is imperative to use sensor networks. Mobility provides us with a clue that can help resolve the mobile replica detection problem. Specifically, a mobile sensor node should never move faster than the system-configured maximum speed. Accordingly, if the mobile node's speed is over the maximum speed, it is then highly likely that at least two nodes with the same identity are present in the network[4].

In simple broadcast protocol, each node in the network uses an authenticated broadcast message to flood the network with its location information. Each node stores the location information for its neighbors and if it receives a conflicting claim, revokes the offending node. This protocol achieves 100% detection of all duplicate location claims if the broadcasts reach every node. This assumption becomes false when the adversary jams key areas or interferes with communication paths through the network[5].

Most of the existing distributed detection protocols [6], [7], [8] adopt the witness finding strategy, in which each node finds a set of sensor nodes somewhere as the witnesses for checking whether there are the same IDs used at different locations, to detect the replicas. In Deterministic Multicast (DM) [6], to improve on the communication cost of the previous protocol, describes a detection protocol that only shares a node's location claim with a limited subset of deterministically chosen "witness" nodes. When a node broadcasts its location claim, its neighbors forward that claim to a subset of the nodes called 'witnesses'. The witnesses are chosen as a function of the node's ID. If the Adversary replicates a node, the witnesses will receive two different location claims for the same node ID. The conflicting location claims become evidence to trigger the revocation of the replicated node.

In the Random Multicast (RM) [6], when a node broadcasts its location, each of its neighbors sends (with probability p) a digitally signed copy of the location claim to a set of randomly selected nodes. Assuming there is a replicated node, if every neighbor randomly selects O(_n) destinations, then exploiting the birthday paradox, there is a non negligible probability at least one node will receive a pair of non coherent location claims. The node that detects the existence of another node in two different locations within the same time-frame will be called witness. The RM protocol implies high communication costs: Each neighbor has to send O(_n) messages.

In the Line Selected Multicast (LSM)[6]protocol, uses the routing topology of the network to detect replication, each node which forwards claims also saves the claim. That is, the forwarding nodes are also witness nodes of a node which has the node ID in a claim. Therefore, LSM gives a higher detection rate than that of RM. However, both protocols have relatively lower detection rates compared with RED.

In the Randomized Efficient Distributed detection (RED) protocol [8], a trusted entity broadcasts a one-time seed to the whole network. The location of the witness node of a node is determined from the node ID and the seed. Because the seed changes every time, an attacker cannot specify the location of a witness node in advance. In Localized Multicast – (SDC, M-PMC)[7] scheme, each node sends a location claim message to a predetermined cell which is grouped in a geographically separated region. Upon arriving at a cell, this message is broadcasted and stored probabilistically at the witness nodes within the cell. Therefore, the detection rate and the communication overhead are tightly related to the number of nodes and the fraction of witness nodes, which store the location claim message in a cell. However, this scheme is not robust when all nodes within a predetermined cell are compromised.

Bekara and Laurent-Maknavicious proposed a new protocol for securing WSN against nodes replication attacks by limiting the order of deployment [9] and no knowledge of nodes deployment locations. Their scheme requires sensors to be deployed progressively in successive generations. Each node belongs to a unique generation. In their scheme, only newly deployed

nodes are able to establish pair-wise keys with their neighbors, and all nodes in the network know the number of highest deployed generation. Therefore, the clone nodes will fail to establish pair-wise keys with their neighbors since the clone nodes belong to an old deployed generation.

III. SYSTEM MODEL

A. Network Model

Assume that the sensor network consists of sensor nodes with IDs,{1,….,n}. The communication is assumed to be symmetric. In addition, each node is assumed to periodically broadcast a beacon containing its ID to its neighbors. This is usually required in various applications, for example, object tracking. The time is divided into time intervals, each of which has the same length. The sensor nodes have mobility and move according to the Random Way Point (RWP) model, which is commonly used in modeling the mobility of adhoc and sensor networks. A Random Way Point model is shown in the below Fig. 2.

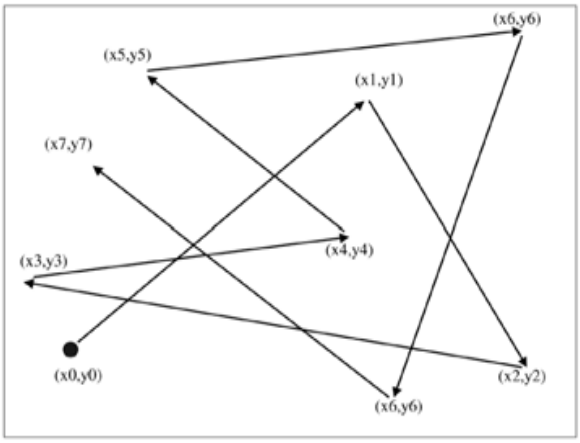

Fig 2: Random Way Point (RWP) Model

Each node is assumed to be aware of its geographic position by means of anchor node connected to GPS. In this model, each node randomly chooses a destination point (waypoint) in the sensing field, and moves toward it with velocity, randomly selected from a predefined interval.

B. Security Model

In this method, sensor nodes are not tamper-resistant. In other words, the corresponding security credentials can be accessed after sensor nodes are physically compromised. Sensor nodes could be compromised by the adversary immediately after sensor deployment. The adversary has all of the legitimate credentials from the compromised nodes. After that, the adversary deploys two or more nodes with the same ID; i.e., replicas into the network. The system architecture is shown in the below Fig.3.

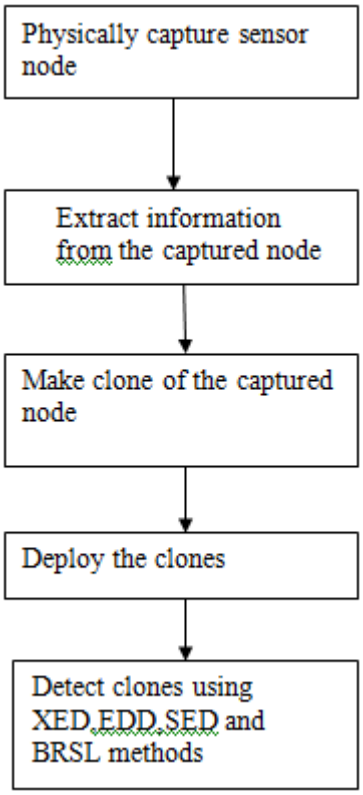

Fig 3: Security Model

IV. THE PROPOSED WORK

A. EXTREMELY EFFICIENT DETECTION (XED)

The idea behind XED is that if a sensor node u meets another sensor node v, u sends a random number to v, and when u and v meet again, u can ascertain whether this is the node met before by requesting the random number. The XED scheme is composed of two steps: an offline step and an online step. The former is executed before sensor deployment while the later is executed by each node after deployment.

Offline Step: A security parameter b and a cryptographic hash function h(.) are stored in each node. Additionally, two arrays $L_r^{(u)}$ and $L_s^{(u)}$ of length n, which keep the received random numbers are used to check the legitimacy of received random numbers, along with a set $B^{(u)}$ representing the nodes having been blacklisted by u, and are stored in each node u. Arrays are initialized to be zero-vectors.

Online Step: If encounters for the first time, u randomly generates $\alpha \in [1,2^{b-1}]$, computes $h(\alpha)$, sends $h(\alpha)$ to v, and stores $L_s^{(u)}[v] = \alpha$. u knows that it encounters v for the first time if $L_s^{(u)}[v] = 0$.

When u encounters v, it first checks v if is in the blacklist $B^{(u)}$. If so, v is considered as a replica by u and u refuses to communicate with v. If not, the following procedures are followed. They exchange the random numbers, $L_r^{(u)}[v]$ and $L_r^{(v)}[u]$. From the viewpoint of node u, after the reception of the random number $L_r^{(v)}[u]$ sent by v, u checks if $L_r^{(v)}[u]$ is the random number u sent to v last time. This can be accomplished by verifying if $h(L_s^{(u)}[v]) = L_r^{(v)}[u]$ holds. Node v is added into $B^{(u)}$ if the verification fails. Otherwise, the same procedure is repeated, including randomly generating a new α, computing $h(\alpha)$, sending to $h(\alpha)$ to v, and replacing the old random number with a new random number is performed.

Algorithm 1. XED

//this algorithm is performed by the node u at each time t
//$v_1,\ldots,v_d$ are the neighbors of u
//$\{v_1,\ldots,v_d\} \in B^{(u)}$
1: send $L_r^{(u)}[v_1], \ldots L_r^{(u)}[v_d]$ to $v_1,\ldots,v_d$, respectively
2: receive $L_r^{(V_1)}[u],\ldots, L_r^{(V_d)}[u]$
3: for k=1 to d
4: if $h(L_s^{(u)}[v_k]) = L_r^{(v_1)}[u]$
5: choose $\alpha \in [1,2^{b-1}]$ and set $L_s^{(u)}[v_k] = \alpha$
6: calculate $h(\alpha)$ and send $h(\alpha)$ to V_k
7: else
8: set $B^{(u)} = B^{(u)} \cup \{V_k\}$

B. EFFICIENT DISTRIBUTED DETECTION (EDD)

The idea behind EDD is motivated by the following observations. The maximum number of times Y1, that node u encounters a specific node v should be limited during a fixed period of time, while the minimum number of times Y2, that u encounters the replicas with the same ID v, should be larger than a threshold during the same period of time. According to these observations, if each node can discriminate between these two cases, it has the ability to identify the replicas. EDD scheme composed of an offline step and an online step.

Algorithm 2. EDD

Offline Steps

1: set T=1 and $B^{(u)} = 0$, $u \in [1,n]$
2: set $L^{(u)}[i]=0$ $1 \le i \le n$, $u \in [1,n]$
3: repeat
4: T=T+1
5: calculate $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$
6: set $Y_1=\mu_1+3\sigma_1$ and $Y_2=\mu_2-3\sigma_2$
7: Until
8: set threshold= $(Y_2-Y_1)/2$

Online steps

// this algorithm is performed by node u at each time *t*
//$V_1,\ldots, V_d$ are the neighbors of u
//$\{V_1\ldots,V_d\} \in B^{(u)}$
1: broadcast beacon b_u //b_u = (u) contains the ID of u
2: if $t \neq t_0$
3: receive beacons $b_{V1}\ldots.. b_{Vd}$
4: for $k = 1$ to d
5: $L^{(u)}[v_k] = L^{(u)}[v_k] + 1$
6: if $L^{(u)}[v_K] > threshold$ then set $B^{(u)} = B^{(u)} \cup \{V_K\}$
7: else //t = t_0
8: set $L^{(u)}[S_k] =0$, k = l,...,n

C.SORAGE EFFICIENT DETECTION(SED) ALGORITHM:

It can be observed from EDD that each node should maintain a list *L,* leading to O(n) storage overhead. SED scheme is proposed to reduce storage overhead. The basic idea behind SED is that instead of monitoring all nodes, each node only monitors a subset of nodes, called monitor set, in a specific time interval. Monitor set contains only one hop neighbour nodes. Therefore storage overhead is reduced to cardinality of the monitor set in the SED scheme.

D.BRSL(BRS BASED ROBUST SECURE LOCALIZED) ALGORITHM.

Sensor Node Examines each of anchor nodes that are available to it. Algorithm is based on BRS and it considers two parameter ai and bi to represent genuine and malicious anchor node. Set value of this parameter based on communication range of anchor node. Malicious anchor node is detected after sensor node obtains the final trust value.

Algorithm 3.BRSL

- Step1: compute ai and bi. Intially ai and bi are set to zero.
- Step2: compute d_{ij} and d_{ij}'
- Step3: if $0<d_{ij}<2R$ is satisfied, then $a_{ij}=1$ otherwise $a_{ij}=0$.
- Step4: if $d_{ij}>2R>0$ is satisfied, then $b_{ij}=1$ otherwise $b_{ij}=0$.

- Step5: Compute trust value of anchor node A_i.
  Trust(Ai)= ai′+1/ (ai′+ bi′+ 2)

V. SIMULATION RESULTS

A mobile wireless sensor network consisting of 50 nodes have been simulated using NS-2. These nodes are connected by wireless links. Nodes moves randomly according to random way point model. Simulation results are shown.

•Detection Accuracy—Detection accuracy is used to represent the false positive ratio and false negative ratio of the underlying detection algorithm, which are the ratios of falsely considered genuine node as a replica and falsely considering a replica as a genuine node, respectively.

• Detection Time—Detection time is evaluated according to the average time required for a genuine sensor node u to add the replica's ID into blacklist.

• Computation Overhead—Computation overhead accounts for the number of operations required for each node to be executed per move.

• Communication Overhead—Communication overhead accounts for the number of records required for each node to be transmitted.

• Storage Overhead—Storage overhead is counted in terms of the number of records required to be stored in each node.

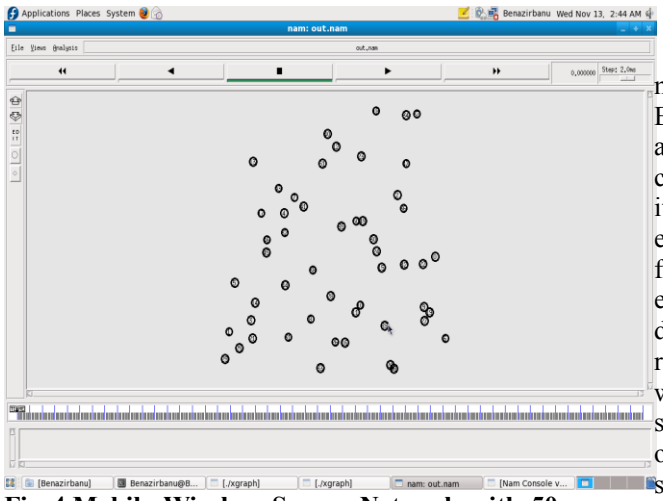

Fig 4.Mobile Wireless Sensor Network with 50 odes

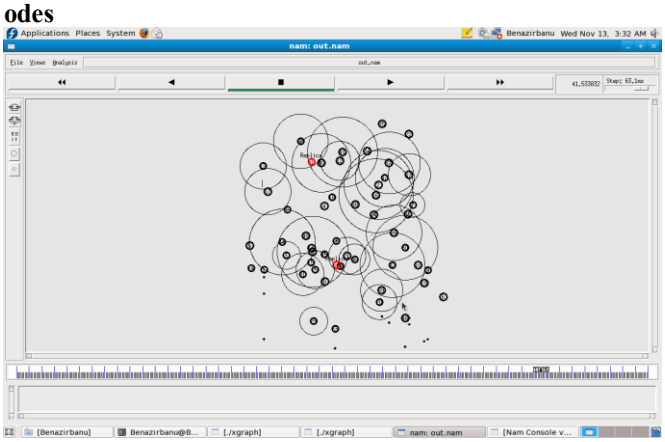

Fig 5. Detection of Clone Nodes

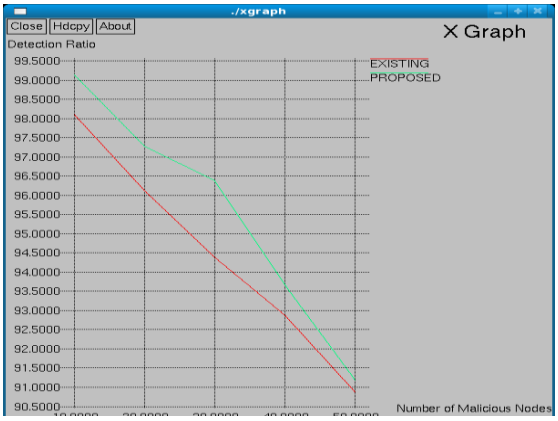

Fig 6. Detection Ratio

VI. CONCLUSION

Clone node detection algorithms for mobile sensor networks, XED,EDD, SED and BRSL are proposed. Although XED is not resilient against collusive replicas, its detection framework, challenge-and-response, it is considered novel and it is compared with the existing algorithms. EDD encounter based detection approach, which is fundamentally different from those used in the existing. In XED and EDD, each node not only detects the replicas by its own effort, but also can revoke the replica in a communication-efficient way. It can be observed from EDD that each node should maintain a list *L,* leading to O(n) storage overhead. SED scheme is proposed to reduce storage overhead. BRSL algorithm have been designed for detection of clone anchor nodes and it improves localization accuracy.

VII. REFERENCES

[1] Chia-Mu Yu, Yao-Tung Tsou, Chun-Shien Lu, "Localized Algorithm for detection of node replication attack in mobile sensor networks" ieee transactions on information forensics and security, vol. 8, no. 5, may 2013.

[2] H. Choi, S. Zhu, and T. F. La Porta, "SET: Detecting node clones in sensor networks," in *Proc. Int. ICST Conf. Security and Privacy in Communication Networks (Securecomm)*, Nice, France, 2007, pp. 341–350

[3] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key predistribution," *IEEE Trans. Syst., Man, Cybern. C, Applicat. Rev.*, vol. 37, no. 6, pp. 1246–1258, Nov. 2007.

[4] J. Ho,M.Wright, and S. K. Das, "Fast detection of replica node attacks in mobile sensor networks using sequential analysis," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Brazil, 2009, pp. 1773–1781.

[5] M. Conti, R.Di Pietro, L. V. Mancini, andA.Mei, "Arandomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proc. ACMInt. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Montreal, Canada, 2007, pp. 80–89.

[6]. Parno B, Perrig A, Gligor V. "Distributed Detection of Node Replication Attacks in Sensor Networks" In: Proceedings of the IEEE Symposium on Security and Privacy; 2005. p. 49 – 63.

[7]. Zhu B, Addada VGK, Setia S, Jajodia S, Roy S. "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks" In: Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007); 2007. p. 257–267

[8]. M. Conti, R. Di Pietro, L.V. Mancini, and A. Mei "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks" In ACM MobiHoc, pages 80–89, 2007

[9]. Yuichi Sei , Shinichi Honiden , "Distributed Detection of Node Replication Attacks resilient to Many Compromised Nodes in Wireless Sensor Networks", 2008 ICST

[10] K. Xing, F. Liu, X. Cheng, and D. Du, "Real time detection of clone attack in wireless sensor networks," in *Proc. IEEE Int. Conf. Distributed Computing Systems (ICDCS)*, Beijing, China, 2008, pp. 3–10.

[11] J. Yi, J. Koo, and H. Cha, "A localization technique for mobile sensor networks using archived anchor information," in *Proc. IEEE Conf. Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, California, USA, 2008, pp. 64–72.

[12] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Mobile sensor network resilient against node replication attacks," in *Proc. IEEE Conf. Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, California, USA, 2008, pp. 597–599, (poster).

[13] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Efficient and distributed detection of node replication attacks in mobile sensor networks," in *Proc. IEEE Vehicular Technology Conf. Fall (VTC-Fall)*, Anchorage, AK, USA, 2009, pp. 1–5.

[14] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Noninteractive pairwise key establishment for sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 556–569, Sep. 2010.

[15] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Hong Kong, China, 2004, pp. 2446–2457.

Benazir Banu studying M.E(computer science and engineering) in Meenakshi college of engineering, Anna University. She completed her B.E(computer science and engineering) in Syed Ammal Engineering College, Anna University. Her research interests include sensor network security and applications.

Anitha Angayarkanni, Assistant Professor, Meenakshi college of engineering. She completed her M.E in Crescent engineering college, Anna University. Her research interests include network security, cloud computing and applicat*ions*

