

Detection of Node Replication Attacks in Mobile Sensor Networks Using Efficient Localized Detection Algorithm

Glory Rashmi. A¹, Mr.C.Muruges²

¹PG Scholar, Department of CSE, National College of Engineering, Tirunelveli

²Assistant Professor, Department of CSE, National College of Engineering, Tirunelveli

¹gloryrash@gmail.com ²cmuruges07@gmail.com

Abstract

Detection of node replication attacks is a challenging task in mobile sensor networks. Most of the existing system used to detect node replication using witness finding strategy which cannot be applied to mobile networks, because in mobile sensor networks nodes frequently change its location within the network or move to other network. Velocity exceeding strategy used in existing system in mobile networks incurs efficiency and security problems. Other techniques used in mobile sensor network incur storage and communication overhead. To improve the performance of the existing algorithms ELD algorithm is proposed.

Keywords—Attacks, Mobile Sensor Network, Node Replication Attack, Security, Sensor Node

a. INTRODUCTION

A. Mobile Sensor Networks

Mobile Sensor networks composed of number of sensor nodes which are having the property of mobility. Each sensor node has 3 parts: a radio transceiver with an internal antenna and connection to an external antenna, a microcontroller and a battery. Mobile Sensor Networks are adapted to many different functions or activities than static sensor networks as they can be deployed in any scenario and cope with rapid topology changes. Sensor nodes are useful in the fields of environmental data collection, security monitoring and object tracking.

A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motest" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.[12]

B. Node Replication Attack

An attacker compromise one node and generate number of replicas with the same id and place them into the network for further malicious activities. This type of attack is known as Node replication attack or node clone attack. The Node replication attack can be exceedingly injurious to many important functions of the sensor network such as routing, resource allocation and misbehavior detection.

Since the credentials of replicas are all clones of the captured nodes, the replicas can be considered as legitimate members of the network, making detection difficult. From the security point of view, the node replication attack is extremely harmful to networks because replicas, having keys, can easily launch insider attacks, without easily being detected. Such attacks may have severe consequences; they may allow the adversary to corrupt network data or even disconnect significant parts of the network.

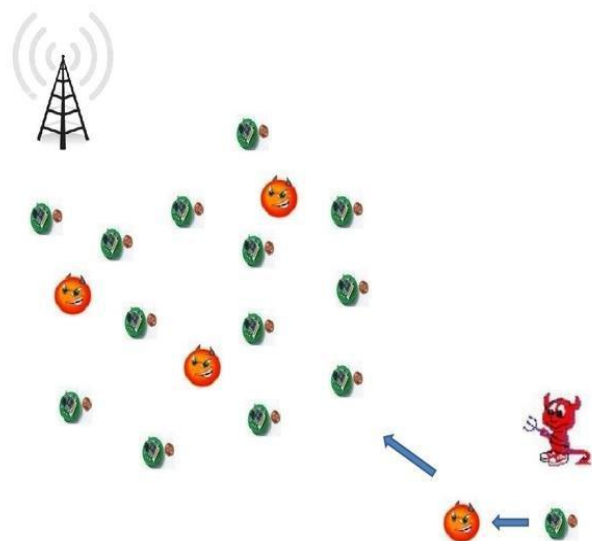


Fig 1: An attacker compromise one node, create clones of it and insert them in the network for further malicious attack.

Unfortunately, sensor nodes typically employ low-cost commodity hardware components unprotected by the type of physical shielding that

could preclude access to a sensor's memory, processing, sensing and communication components. Cost considerations make it impractical to use shielding that could detect pressure, voltage, and temperature changes that an adversary might use to access a sensor's internal state. Deploying unshielded sensor nodes in hostile environments enables an adversary to capture, replicate, and insert duplicated nodes at chosen network locations with little effort. Thus, if the adversary compromises even a single node, she can replicate it indefinitely, spreading her influence throughout the network. If left undetected, node replication leaves any network vulnerable to a large class of insidious attacks. Using replicated nodes, the adversary can subvert data aggregation protocols by injecting false data or suppressing legitimate data.

Since a clone has legitimate information (code and cryptographic material), it may participate in the network operations in the same way as a non-compromised node; hence, cloned nodes can launch a variety of attacks. A few have been described in the literature [13], [14]. For instance, a clone could create a black hole, initiate a wormhole attack [15] with a collaborating adversary, or inject false data or aggregate data in such a way to bias the final result [15]. Further, clones can be leak data. The threat of a clone attack can be characterized by two main points: First thing is, A clone is considered totally honest by its neighbors. In fact, without global countermeasures, honest nodes cannot be aware of the fact that they have a clone among their neighbors. Second thing, To have a large amount of compromised nodes, the adversary does not need to compromise a high number of nodes. Indeed, once a single node has been captured and compromised, the main cost of the attack has been sustained. Making further clones of the same node can be considered cheap. While centralized protocols have a single point of failure and high communication cost, local protocols do not detect replicated nodes that are distributed in different areas of the network. In this work, we look for a network self-healing mechanism, where nodes autonomously identify the presence of clones and exclude them from any further network activity.

C. Related Work

Most of the existing systems used witness finding strategy to detect node replicas. Algorithms varied by means of selecting witnesses. Random key pre distribution security schemes are well suited for use in sensor networks due to their low overhead. However, the security of a network using pre distributed keys can be compromised by cloning attacks. Cloning gives the adversary an easy way to build an army of malicious nodes that can cripple the sensor network. Brooks proposed an algorithm that a

sensor network can use to detect the presence of clones. Keys that are present on the cloned nodes are detected by looking at how often they are used to authenticate nodes in the network.[1]

Two approaches proposed in [6] by Parno et al are Randomized Multicast and Line-Selected Multicast (LSM). In Randomized Multicast the neighbors of each node randomly select \sqrt{n} nodes as that node's witnesses. Then if a node is replicated, according to the birthday paradox problem, at least one witness will receive two conflicting location claims with high probability. In LSM the nodes in the paths from a node's neighbors to the randomly selected witnesses are used; these nodes become the node's witnesses too. Such change reduces the communication cost per node from $O(n)$ to $O(\sqrt{n})$. Zhu et al. [16] divided the network into cells, and proposed two approaches: SDC and P-MPC. In SDC, each node ID is mapped to one cell, and the location claim of each node is forwarded to the mapped cell and broadcasted within the cell. Nodes in the cell store the claim and become that node's witnesses with some probability. P-MPC is different from SDC in that each node ID is mapped to multiple cells with different probabilities, however, the set of possible mapped cells is still deterministic. Melchor et al. [18] proposed an active detection approach, in which witness nodes actively obtain location claims. Each node first randomly chooses several nodes and becomes their witness node. Then if a node is node a 's witness node, it will send location-claim request through several relay nodes to node a . These relay nodes are randomly chosen by the witness node for a . Thus if a has a replica, the replica will have high probability to receive the request as well, and reply a conflicting location claim to the witness node.

Based on random walk, two NDFD protocols [8], RANdomWaLk (RAWL) and Table-assisted RANdomWaLk (TRAWL), are proposed which fulfill the requirements while having only moderate communication and memory overheads. It performs the following steps: Each node broadcasts a signed location claim. Each of the node's neighbors probabilistically forwards the claim to some randomly selected nodes. Each randomly selected node sends a message containing the claim to start a random walk in the network, and the passed nodes are selected as witness nodes and will store the claim. If any witness receives different location claims for a same node ID, it can use these claims to revoke the replicated node. The random walk strategy outperforms previous strategies because it distributes a core step, the witness selection, to every passed node of random walks, and then the adversary cannot easily find out the critical witness nodes.

Another one technique use Bloom filters to compress the information stored at the sensors, and use two new techniques, called cell forwarding and

cross forwarding, to improve detection probability, further reduce memory consumption, and in the mean time distribute the memory and energy overhead evenly across the whole network.[9]

A novel scheme for detecting clone attacks in sensor networks is proposed which computes for each sensor a social fingerprint by extracting the neighborhood characteristics, and verifies the legitimacy of the originator for each message by checking the enclosed fingerprint. The fingerprint generation is based on the superimposed s-disjunct code, which incurs a very light communication and computation overhead. The fingerprint verification is conducted at both the base station and the neighboring sensors, which ensures a high detection probability [7].

SET algorithm is to detect clones by computing set operations (intersection and union) of exclusive subsets in the network. First, SET securely forms exclusive unit subsets among one-hop neighbors in the network in a distributed way. This secure subset formation also provides the authentication of nodes' subset membership. SET then employs a tree structure to compute non overlapped set operations and integrates interleaved authentication to prevent unauthorized falsification of subset information during forwarding. Randomization is used to further make the exclusive subset and tree formation unpredictable to an adversary.[4]

Wright proposed a scheme called Sequentially Probability Ratio Test[5], which performs the following steps: Each time a mobile sensor node moves to a new location, each of its neighbors asks for a signed claim containing its location and time information and decides probabilistically whether to forward the received claim to the base station. The base station computes the speed from every two consecutive claims of a mobile node and performs the SPRT by taking speed as an observed sample. Each time maximum speed is exceeded by the mobile node, it will expedite the random walk to hit or cross the upper limit and thus lead to the base station accepting the alternate hypothesis that the mobile node has been replicated. On the other hand, each time the maximum speed of the mobile node is not reached, it will expedite the random walk to hit or cross the lower limit and thus lead to the base station accepting the null hypothesis that mobile node has not been replicated. Once the base station decides that a mobile node has been replicated, it initiates revocation on the replica nodes.

D. Challenges in Detecting Node Replication Attacks in Mobile Sensor Networks

The witness-finding strategy exploits the fact that one sensor node cannot appear at different locations, but, unfortunately, the sensor nodes in

mobile sensor networks have the possibility of appearing at different locations at different times, so the above schemes cannot be directly applied to mobile sensor networks. Slight modification of these schemes can be helpful for applicability to mobile sensor networks. For instance, the witness-finding strategy [5] can adapt to mobile environments if a timestamp is associated with each location claim. In addition, setting a fixed time window in advance and performing the witness-finding strategy for every units of time can also keep witness-finding feasible in mobile sensor networks. Nevertheless, accurate time synchronization among all the nodes in the network is necessary. Moreover, when witness-finding is applied to mobile sensor networks, routing the message to the witnesses incurs even higher communication cost. After identifying the replicas, a message used to revoke the replicas, possibly issued by the base station or the witness that detects the replicas, is usually flooded throughout the network. Nevertheless, network-wide broadcast is highly energy-consuming and, therefore, should be avoided in the protocol design. Time synchronization is needed by almost all detection algorithms. Nevertheless, it is still a challenging task to synchronize the time of nodes in the network, even though loose time synchronization is sufficient for the detection purpose.

The effectiveness of witness-finding could be reduced when a large number of sensor nodes have been compromised, because the compromised nodes can block the message issued by the nodes near the replicas. Hence, the witness nodes cannot discover the existence of replicas. To cope with this issue, localized algorithms could enhance the resilience against node compromise. There are two localized algorithms eXtremely Efficient Detection (XED)[10] and Efficient Distributed Detection (EDD)[11] already proposed to detect the node replication in mobile sensor networks.

Section II describes about XED and EDD algorithm which is the base for our proposed algorithm. Section III provides the system model and network model of the proposed algorithm. Section IV depicts the two steps in ELD (proposed algorithm) and its benefits. Section V deals with the performance analysis and the following section conclude the paper.

II. XED AND EDD ALGORITHM

A.XED

The idea behind XED is motivated by the observation that, if a sensor node meets another sensor node at an earlier time and sends a random number to at that time, then, when and meet again, can ascertain whether this is the node met before by requesting the random number. Note that, in XED, we assume that the replicas cannot collude with each

other. Specifically, the XED scheme is composed of two steps: In *offline step* 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 and the materials used to check the legitimacy of received random numbers, respectively, along with a set $B^{(u)}$ representing the nodes having been blacklisted by u , are stored in each node. $L_r^{(u)}$ and $L_s^{(u)}$ are initialized to be zero-vectors. $B^{(u)}$ is initialized to be empty.

In *online step* If u encounters v for the first time, u randomly generates α , computes $h(\alpha)$, sends $h(\alpha)$ to v and stores $L_s^{(u)}[v] = \alpha$. The second time when they encounter, they exchange the random numbers $L_s^{(v)}[u]$ and $L_r^{(u)}[v]$. u checks that the random number received from v by verifying $h(L_s^{(u)}[v]) = L_r^{(v)}[u]$. If it fails u add v to its blocking list.

B. EDD

The idea behind EDD is motivated by the following observations. The maximum number of times, Y_1 that node encounters a specific node, should be limited with high probability during a fixed period of time, while the minimum number of times, Y_2 , that u encounters the replicas with the same ID, 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. Different from XED, EDD assumes that the replicas can collude with each other. In addition, all of the exchanged messages should be signed unless specifically noted. Particularly, the EDD scheme is composed of two steps: an offline step and an online step. The offline step is performed before sensor deployment. The goal is to calculate the parameters, including the length of the time interval T and the threshold ψ used for discrimination between the genuine nodes and the replicas. On the other hand, the online step will be performed by each node at each move. Each node checks whether the encountered nodes are replicas by comparing with the corresponding number of encounters.

III. SYSTEM MODEL

A. Network Model

Assume that the sensor network consists of sensor nodes with IDs $\{1, 2, \dots, n\}$. The communication is assumed to be symmetric. In addition, each node is assumed to periodically broadcast a beacon containing its ID to its neighbours. 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. Nonetheless, the time among sensor nodes does not need to be synchronized. The sensor nodes have mobility and

move according to the Random WayPoint (RWP) model [17] which is commonly used in modeling the mobility of ad-hoc and sensor network. Each node is assumed to be able to be aware of its geographic position. 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. After reaching the destination point, the node remains static for a random time and then starts moving again according to the same rule.

B. Security Model

Assume that 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.

IV. ELD ALGORITHM

ELD (Efficient Localized Detection) algorithm is a modified version of EDD (Efficient Distributed Detection). In this algorithm each node detects the replica by itself and will detect the replica at different time. ELD scheme is composed of two steps: an offline step and an online step. The former is executed before sensor deployment while the latter is executed by each node after deployment.

A. Offline Step in ELD

There are two lists used to find replicas in this algorithm. They are L_s and B used to store number of times a node send beacon message to its neighbours and nodes that are detected as replica by original nodes respectively. In offline step the lists are initialized to zero.

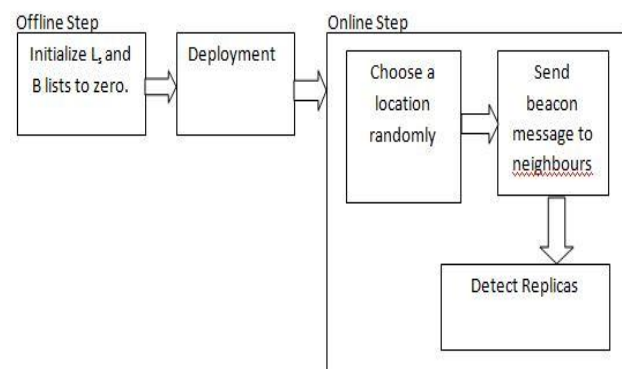


Fig 2: Architecture of ELD Algorithm

B. Online Step in ELD

In online step, each node has to send the beacon message to its neighbours. Beacon message

contains the id of the sending node. The number of times a node send the beacon message to other nodes is store in a list L_s . Based on the value of L_s replicas are find out by following: Number of times a node send beacon message to original node should be less than number of times a node send beacon message to clone nodes. The nodes which are detected as replicas by original nodes are stored in B list. Here replicas are detected by each node.

```

Algorithm: ELD-Online-Step
for k=1 to d
    send beacon message
    if  $L_s(d) > 1$ 
        store d to B
    else
        start transferring of data;
        Set  $L_s$  to zero
    
```

C. Advantages of ELD Algorithm

1. Localized Detection: Each node in the network have to communicate with its one hop neighbor to detect replicas. There is no base station needed for this algorithm. Thus there is no single point failure.
2. Efficiency and Effectiveness: ELD algorithm detects replicas with high accuracy.
3. Network-Wide Revocation Avoidance: Detect replicas without flood the entire network.
4. Computation Overhead: There is no Computation overhead in this algorithm.
5. Detection Time: Detection time also very less.
6. Time Synchronization: Time Synchronization is not needed. Because nodes work independently

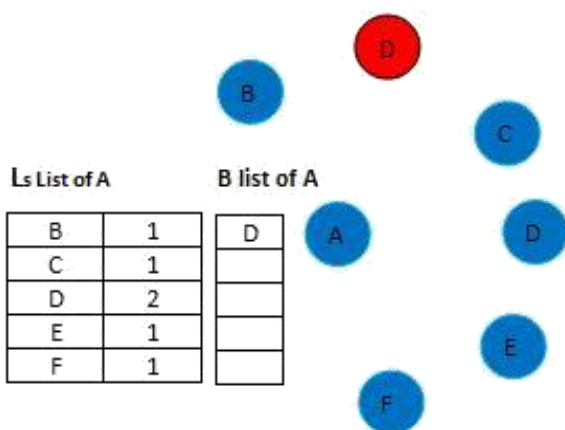


Fig:3 Ls List contains the number of times a node sends its beacon message to its neighbours. Here D is a replica node. Thus value of D in Ls list is greater than 1. So D is added to A's Blacklist.

V. PERFORMANCE ANALYSIS

Four performance metrics are used for evaluation:

A. Detection Time

Detection time is evaluated according to the average time (or, equivalently, the number of moves) required for a genuine sensor node to add the replica's ID into Black list.

A node detect replica as soon as the replica node as its neighbour. In Mobile Sensor networks nodes move using Random WayPoint model. Thus detection time of replicas is less than the previous methods.

B. Storage Overhead

Storage overhead is counted in terms of the number of records required to be stored in each node. In ELD algorithm only two lists are needed to detect replicas. There is no additional parameters needed like other algorithms.

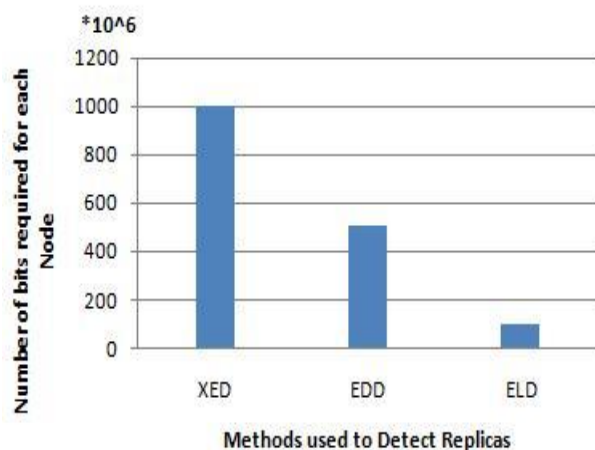


Fig 4: Comparison of storage overhead for the proposed and existing system.

C. Computation Overhead

Computation overhead accounts for the number of operations required for each node to be executed per move. There is no computation needed to detect replicas. Sending of beacon message only used to detect replicas.

D. Communication Overhead

Communication overhead accounts for the number of records required for each node to be transmitted. Communication overhead does not occur in this algorithm.

VI. CONCLUSION

In this paper an algorithm ELD (Efficient Localized Detection) is proposed to detect replicas in mobile sensor networks. ELD algorithm detects replicas by itself. It does not require base station or

any other nodes help to detect replicas. Records used to detect replicas also less than the existing algorithms. It detects replicas by sending beacon message to its neighbours. So it does not flood the entire network. Thus the efficiency of the algorithm is better than the previous algorithms that are used to detect node replication attacks in mobile sensor network.

References

- [1] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key pre distribution," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1246–1258, Nov. 2007.
- [2] M. Conti, R. DiPietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proc. ACM Int. Symp. Mobile AdHoc Networking and Computing (MobiHoc)*, Montreal, Canada, 2007.
- [3] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," *IEEE Trans. Depend. Secure Comput.*, vol. 8, no. 5, pp. 685–698, Sep./Oct. 2012.
- [4] 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.
- [5] 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.
- [6] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proc. IEEE Symp. Security and Privacy (S&P)*, Oakland, CA, USA, 2005, pp. 49–6.
- [7] B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: Efficient and distributed replica detection in large-scale sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 913–926, Jul. 2010.
- [8] Y. Zeng, J. Cao, S. Zhang, S. Guo, and L. Xie, "Random-walk based approach to detect clone attacks in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, pp. 677–691, Jun. 2010.
- [9] M. Zhang, V. Khanapure, S. Chen, and X. Xiao, "Memory efficient protocols for detecting node replication attacks in wireless sensor networks," in *Proc. IEEE Int. Conf. Network Protocols (ICNP)*, Princeton, NJ, USA, 2009, pp. 284–293.
- [10] 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.
- [11] Chia-Mu Yu, Yao-Tung Tsou, Chun-Shien Lu and Sy-Yen Kuo, "Localized Algorithms for Detection of Node Replication Attacks in Mobile Sensor Network," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, VOL. 8, NO. 5, MAY 2013
- [12] http://en.wikipedia.org/wiki/Wireless_sensor_network
- [13] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks," *Proc. Int'l Conf. Security in Pervasive Computing (SPC '06)*, pp. 104–118, 2006.
- [14] S. Capkun and J.-P. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," *Proc. IEEE INFOCOM '05*, pp. 1917–1928, 2005.
- [15] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," *Proc. IEEE INFOCOM '03*, pp. 1976–1986, 2003.
- [16] B. Zhu, V. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in *Proc. 23rd Ann. Computer Security Applications Conference (ACSAC '07)*, Dec. 2007, pp. 257–26.
- [17] C. Bettstetter, H. Hartenstein, and X. P. Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Netw.*, vol. 10, no. 5, pp. 555–567, 2004.
- [18] C. A. Melchor, B. Ait-Salem, P. Gaborit, and K. Tamine, "Active detection of node replication attacks," *Int. J. of Computer Science and Network Security*, vol. 9, no. 2, pp. 13–21, 2009.