IJCN Special Issue - Data Management and Network Control in Wireless Networks (SICN)
Singaporean Journal of Scientific Research(SJSR)
Vol 6.No.3 2014 Pp.199-207
Available at:www.iaaet.org/sjsr
Paper Received:10-03-2014
Paper Accepted:29-03-2014
Paper Reviewed by:1Prof .Dr.Ponnambalam 2. Chai Cheng Yue
Editor: Dr. Binod Kumar

EFFECTIVE AND EFFICIENT DETECTION OF NODE REPLICATION ATTACKS IN MOBILE SENSOR NETWORK

S.Silambarasi, Dr. G. J. Joyce Mary
Dept. of CSE,
PRIST University.,
Thanjavur-613 403.
Email: simbu.s90@gmail.com dr.joycedani@gmail.com

ABSTRACT

We deal with the challenging problem of node repli-cation detection. Although defending against node replication attacks demands immediate attention, compared to the extensive exploration on the defense against node replication attacks in static networks, only a few solutions in mobile networks have been presented. Moreover, while most of the existing schemes in static networks rely on the witness- finding strategy, which cannot be applied to mobile networks, the velocity-exceeding strategy used in existing schemes in mobile networks incurs efficiency and security problems. Therefore, based on our devised chal-lenge-and-response and encounter-number approaches, localized algorithms are proposed to resist node replication attacks in mo-bile sensor networks. The advantages of our proposed algorithms include 1) localized detection; 2) efficiency and effectiveness; 3) network-wide synchronization avoidance; and 4) network-wide revocation avoidance. Performance comparisons with known methods are provided to demonstrate the efficiency of our pro-posed algorithms. Prototype implementation on TelosB mote demonstrates the practicality of our proposed methods.

Index Terms— Attack, security, wireless sensor networks.

1. INTRODUCTION

A. Node Replication Attacks

Sensor networks, which are composed of a number of sensor nodes with limited resources, have been demon-strated to be useful in applications, such as environment mon-itoring and object tracking. As sensor networks could be de-ployed in a hostile region to perform critical mission s, the sensor networks are unattended and the sensor nodes normally are not equipped with tamper-resistant hardware. This allows a situa-tion where the adversary can compromise one sensor node, fab-ricate many replicas having the same identity (ID) from the cap-tured node, and place these replicas back into strategic positions in the network for further malicious activities. This is a so-called *node replication attack*. 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. Recently, due to advances in robotics, mobile sensor net- works have become feasible and applicable. Nevertheless, al-though the problem of node replication detection in static net-works has been extensively studied, only a few schemes have been proposed for mobile sensor networks. Even worse, as indi-cated in [22], the techniques used in detecting replicas in static environments are not useful in identifying replicas in mobile environments. With the consideration of nodes' mobility and the distributed nature of sensor networks, it is desirable, but very challenging, to have efficient and effective distributed al- gorithms for detecting replicas in mobile sensor networks.

B. Related Work

ased on the assumption that a sensor node, when attempting to join the network, must broadcast a signed location claim to its neighbors, most of the existing distributed detection proto-cols adopt the *witness-finding* strategy to detect the replicas. In particular, the general procedure of applying wit-ness-finding to detect the replicas can be stated as follows. After collecting the signed location claims for each neighbor of the node here and denote the location of and the digital signature function [15], [16], respectively, sends the collected signed location claims to a properly selected subset of nodes, which are witnesses. When there are replicas in the network, the witnesses, according to the received location claims, have possibility to find a node ID with two distant locations, which implies that the node ID is being used by replicas. Afterward, the detected replicas can be excluded using, for example, network-wide revocation. The de-tection algorithms proposed all be-long to this category. For example, RM and LSM, were pro-posed in to determine the witnesses randomly. The differ-ence between RM and LSM is that the witness nodes that find the conflicting location in the former are primarily affected by the number of witness nodes and the ones in the latter are pri-marily affected by the forwarding traces of location claims

2. SYSTEM MODEL

A. Network Model

Assume that the sensor network consists of sensor node withIDs $\{1, \ldots, n\}$ The communication is assumed to be sym- metric. In addition, each node is assumed to periodically broad-cast 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 .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 which is commonly used in modeling the mobility of *ad hoc* and sensor networks . 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[v(max)-v(min)] . After reaching the destination point, the node remains static for a random time and then startsmoving again according to the same rule. To simplify the analysis, we assume each node has neighbors on average permove. Finally, we follow the conventional assumption in prior works that the network utilizes an identity-based public key system , so signature generation and verification are feasible. In general, the models used in this paper are the sam e as the ones in prior works.

Owing to the use of the digitalsignature function the replicas cannot create a new ID or disguise them selves as the nodes being not compromised before, because it is too difficult for the adver-sary to have the corresponding security credentials. Since the Focus of this paper is on the node replication attacks

3. THE PROPOSED METHODS

In this section, our proposed algorithms, eXtremely Efficient Detection (XED)1 and Efficient Distributed Detection (EDD),

A. XED

The idea behind XED is motivated by the observa tion that, if a sensor node meets another sensor node at an earlier time and sends a random number to at that time, thenwhen and meet again, can ascertain whether this is the node met before by requesting the random number. Note that, in XED, we as-sume that the replicas cannot collude with each other but this as-sumption will be removed in our next solution in Section III-B. In

B. EDD

Algorithmic Description of EDD: The idea behind EDD is motivated by the following observations. The maximum number of times, , that node encounters a specific node should be limited with high probability during a fixed period of time, while the minimum number of times, , that 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 and the threshold 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 encounterednodes are replicas by comparing with the corresponding number of encounters. In the following, we somewhat abuse the notation; we denote the start time of each interval as u. **Offline Step**. The offline step of EDD is shown in Fig. 2. The array of length is used to store the number of encounterswith every other node in a given time interval,while set contains the IDs having been considered by as replicas.

Algorithm: XED-On-line-Step
// this algorithm is performed by the node u at each time t
// $v_1, \ldots, v_d$ are the neighbors of u
// $\{v_1, \ldots, v_d\} \notin \mathcal{B}^{(u)}$
1: send $\mathcal{L}_r^{(u)}[v_1], \ldots, \mathcal{L}_r^{(u)}[v_d]$ to $v_1, \ldots, v_d$, respectively
2: receive $\mathcal{L}_r^{(v_1)}[u], \ldots, \mathcal{L}_r^{(v_d)}[u]$
3: **for** $\kappa = 1$ to d
4: **if** $h(\mathcal{L}_s^{(u)}[v_\kappa]) = \mathcal{L}_r^{(v_\kappa)}[u]$
5: choose $\alpha \in [1, 2^b - 1]$ and set $\mathcal{L}_s^{(u)}[v_\kappa] = \alpha$
6: calculate $h(\alpha)$ and send $h(\alpha)$ to v_κ
7: **else**
8: set $\mathcal{B}^{(u)} = \mathcal{B}^{(u)} \cup \{v_\kappa\}$

Figure 1. Online step of the XED scheme.

addition, all of the exchanged messages should be signed un-less specifically noted. Specifically, the XED 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.

u

Algorithm: EDD-Off-line-Step
1: set $T = 1$ and $\mathcal{B}^{(u)} = \emptyset$, $u \in [1, n]$
2: set $\mathcal{L}^{(u)}[i] = 0$, $1 \le i \le n$, $u \in [1, n]$
3: **repeat**
4: $T = T + 1$,
5: calculate μ_1, μ_2, σ_1^2, and σ_2^2
6: set $Y_1 = \mu_1 + 3\sigma_1$ and $Y_2 = \mu_2 - 3\sigma_2$
7: **until** $Y_1 < Y_2$
8: set $\psi = \frac{Y_2 - Y_1}{2}$

Figure. 2. Offline step of the EDD scheme.

4. PERFORMANCE EVALUATION

Five performance metrics are used in our evaluation:

• *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 falselyconsidering a genuine node as a replica and falsely considering a replica a genuine node, respectively.
• *Detection Time*—Detection time is evaluated according tothe average time (or, equivalently, the number of moves)
required for a genuine sensor node to add the replica's
ID into .B(u) • *Storage Overhead*—Storage overhead is counted in terms of the number of records required to be stored in each node. Here, the *records* differ in

different algorithms.For example, a record is a tuple containing an ID, time, location o(log n),and signature in while a record involves only an ID, location, and signature in . If the storage overhead is counted in terms of the number of *bits*, a multiplicative factor is obviously needed due to the space for IDs. Nonetheless, for fair comparison, we do not use such bit-based storage estimation.

• *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 .

Algorithm: EDD-On-line-Step
// this algorithm is performed by node u at each time t
// $v_1, \ldots, v_d$ are the neighbors of u
// $\{v_1, \ldots, v_d\} \notin \mathcal{B}^{(u)}$
1: broadcast beacon b_u //$b_u = \langle u \rangle$ contains the ID of u
2: **if** $t \neq t_0$
3: receive beacons $b_{v_1}, \ldots, b_{v_d}$
4: **for** $\kappa = 1$ to d
5: $\mathcal{L}^{(u)}[v_\kappa] = \mathcal{L}^{(u)}[v_\kappa] + 1$
6: **if** $\mathcal{L}^{(u)}[v_\kappa] > \psi$ **then** set $\mathcal{B}^{(u)} = \mathcal{B}^{(u)} \cup \{v_\kappa\}$
7: **else** //$t = t_0$
8: set $\mathcal{L}^{(u)}[s_\kappa] = 0, \kappa = 1, \ldots, n$

Figure 3. Performance Evaluation

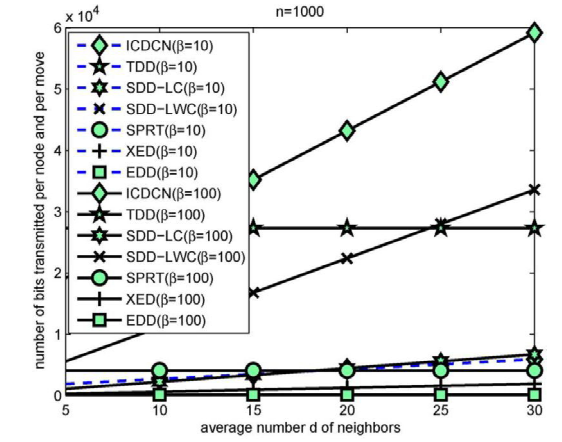
Our strategy for calculating and is as follows. We fix the location of a sensor node and the corresponding communication disk. Consider a moving sensor node. What we do is to derive the distribution of the number of times the moving node stays in the communication disk of the static node. In essence, this is the distribution of the number of encounters with a genuine node. Once the distribution can be obtained, and can be easily derived. The same strategy applies in the calculation of and sume that the moving node takes steps during a time interval of length . Let the length (the distance between and) of -th step be . The time taken by the -th step is . We can derive that:

E[ti]= E[l1/v +w] = E[l2]/v+w

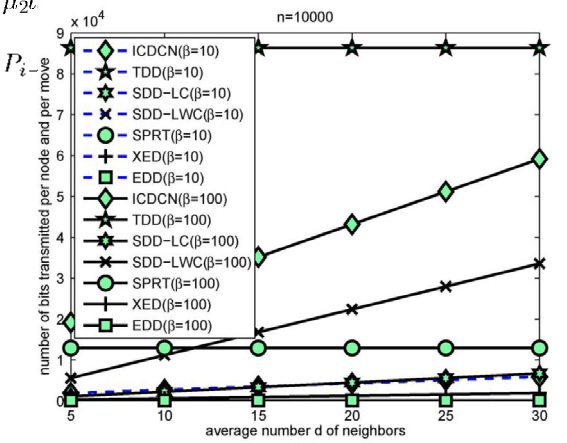
exactly from (2), the parameters and , which represent the expected value and variance of the number of encounters with the genuine node, respectively, can be obtained by assuming one single communication disk in the sensing field as described above.

By a similar calculation, the parameters and can be calculated by placing two or more communication disks in the sensing field. For example, when a network with two replicas is considered, the probability , instead of , is used in the derivation.

In EDD, for each encountered node, the computati on required for each node per move is to update its CM sketch and to check if the number of encounters with the encountered node exceeds the threshold . As indicated in [3], the number of operations required for the update and for the query isAs a result, EDD incurs computations, which implies computation overhead.



(a)



(b)

5. SIMULATION RESULTS

The effectiveness of XED relies on the simple challenge-and-response framework, which obviously holds. Nevertheless, the performance of EDD varies according to different network set-tings. This section is devoted to validating the effective-ness of EDD through a simulation. Within a period of time with length properly chosen according to the offline step of EDD, the number of encounters with the genuine node and the number of encounters with the replicas can be distinguished well if the threshold is set in a way indicated in Fig. 2. We discuss how the parameters, such as communication range and node velocity, affect the

detection.

First of all, with the simulation plots in Fig. 6 where the dif-ferent network settings used are also described, we know that the number of encounters really looks like a Gaussian distribu-tion. Recall that this result has been theoretically demonstrated in Section III-B2. In the following, the relationship between the detection time, detection accuracy, and the other parameters in the EDD algo-rithm can be unified in a framework of the law of large numbers.

A. Detection Time and Detection Accuracy

As shown in each row of Fig. 6, it can be easily observed that when the number of movements in an interval grows larger, it becomes easier to distinguish between the genuine node and replicas. Here, the term "easier" means higher detec-tion accuracy (defined in Section IV). This phenomenon can be explained because, from the law of large numbers point of view, the number of encounters with the replicas will get closer to its expected value, which is actually twice as many as the number of encounters with the genuine node. This also means that, although increasing the time interval size can be useful in enhancing the detection accuracy, however, the improvement of detection accuracy cannot be unlimited.

As the foundational truth is that there are two replicas in the simulation (Fig. 6), the mean value of the distribution of the number of encounters with the replicas must concentrate on the value double that of the number of encounters with the genuine node even if is set to be quite large. Our experience shows at least the detection accu-racy of both false positive and false negative ratios lower than 3% is achievable even if there are only two replicas in the net-work.

B. The Effect of Communication Range

By comparing the first row and third row (or the second row and fourth row) of Fig. 6, we can observe that, when sensor nodes have a larger communication range, the distributions of the number of encounters with the genuine node and r eplicas can be better separated.

In the extreme but unrealistic case where the communication range is set to be infinity, since the genuine node may be simultaneously aware of two signal sources from the same node ID, the replicas can be easily identified. Such an effect caused by the increased communication range can also be explained by the law of large numbers. In particular, the larger Work with a larger communication range can have more samples than the one with a shorter communication range. Thus, simi-larly, from the law of large numbers point of view, the expected number of encounters with the replicas will more concentrate on the value that is twice as many as the value corresponding to the expected number of encounters with genuine node, resulting in the better separation.

C. The Effect of Movement Velocity

The comparison between the first row and second row (or the third row and the fourth row) of Fig. 6 shows that the network with faster node velocity is more resilient against the node repli-cation attack if EDD is used. The rationale behind this can be explained.

The process of sensor movements where a node has certain neighboring nodes can be thought of as a sampling process that a subset of balls from a bin containingdifferent balls is re-cursively sampled with replacement. In this analogy, the ball is the sensor node in the network. When the node velocity is low, there is greater possibility for two nodes to meet at a certain time instance and meet again at the next time instance. Con-sider the extreme but unrealistic case that the nodes move with infinite velocity.

This is equivalent to uniformly sampling the balls. According to the law of large numbers, in such a case, the expected number of encounters with the replicas will more con-centrate on the value that is twice the value corresponding to the expected number of encounters with a genuine node.

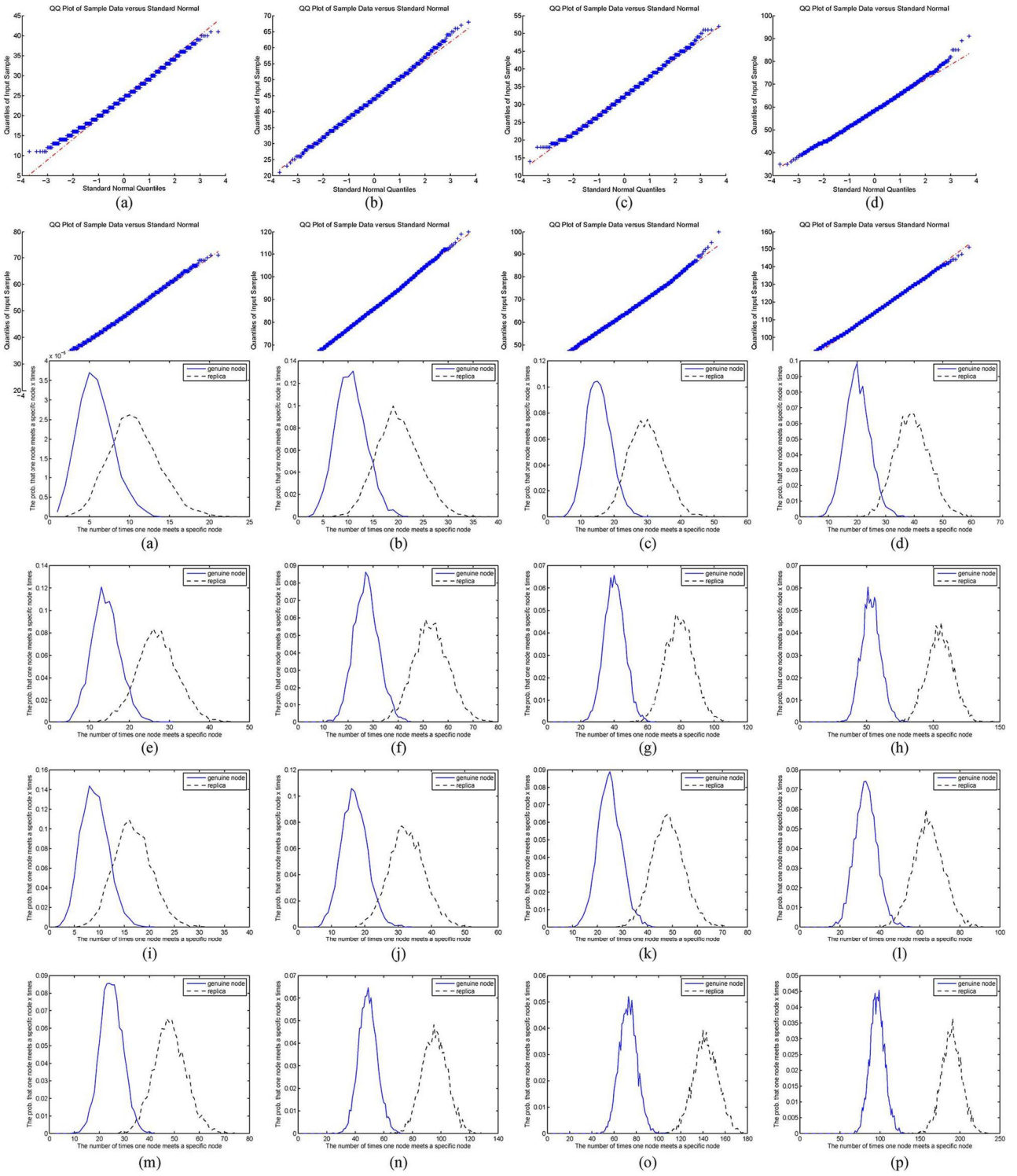


Figure. 6. Distribution of the number of encounters in the different scenario

TABLE II DETECTION SPEED OF THE PROPOSED SCHEMES

	ROM	RAM
XED	13434 Bytes	633 Bytes
EDD	11418 Bytes	531 Bytes

	execution speed
XED	≈ 0.25s
EDD	≈ 0.043s

practicality of our proposed XED and EDD schemes for the current generation of sensors, we have imple-mented a prototype of our schemes.

In the implementation, our focus is only on the memory over-head incurred by the proposed XED and EDD schemes,. Each and every sensor node sensing the data and send it to main system. Meanwhile at any cost sensing data don't leak to others (Any replicated node). Normally in mobile sensor network data transfer to hop by hop at that time any replicated node act as original node, our algorithm find it and avoid them. Again find alternate hop using candidate list and execute proposed algorithm alternate node is original then data send to them. This process made up to when data reach the main system.

TABLE III MEMORY CONSUMPTION OF THE PROPOSED SCHEMES

	CPU	Radio	Overall
XED	1201.3004 mJ	1901.8623 mJ	3103.1627 mJ
EDD	964.9375 mJ	1580.1480 mJ	2545.0855 mJ

the local memory of current sensor nodes is limited. In the im-plementation of XED, we assume that the arrays, and , are of 128 bits length. The packet each node sends contains the node ID and the corresponding replied number. In our experi-mental setting, each node is assumed to have five neighboring nodes. On the other hand, we assume that the array is of 128 bits length in the implementation of EDD. The packet each node sends merely contains the node ID. Together with the AES encryption function in a CC2420 chipset, CBC-MAC mode is used to implement the hash function. Table II reports the re-sults of the prototype implementation of the proposed XED and EDD schemes. Note that the program size reported in Table II includes the program code for not only checking mechanism in-troduced by proposed methods but also communication mecha-nism that is commonly required by many other sensor network applications. Thus, the program size reported in Table II could be an overestimation.

Table III presents the detection time of proposed methods. As mentioned above, each node is assumed to have five neigh-boring nodes. This means that each node needs to send five packets for detection purpose. Since only one packet will be sent for every second for our mote hardware, the reported detection time will be affected by such setting. If we ignore the time delay incurred by our hardware setting, we can observe and infer that the detection time incurred by the additional calculation is less than one second in XED and is even less than 0.1 seconds in EDD.

Our program code was also run on TOSSIM in Tin yOS 1.1.15 to evaluate the energy consumption of XED and EDD. Note that TOSSIM is a discrete-event simulator especially for TinyOS WSNs, on which TinyOS code can be executed directly. Due to the above feature, though TOSSIM is, in essence, a simulator, its estimation of energy consumption is rather accurate. The re-sults of TOSSIM simulation are depicted in Table IV, where the period we conducted the simulation was 100 seconds. Sim-ilar to memory consumption in Table II, the energy consump-tion here could be an overestimation because the energy con-sumption incurred by packet transmission and reception is also counted.

7. DISCUSSION

Since all of the existing detection algorithms for mobile net-works rely on the verification of sensors' locations at different times, time synchronization is essential. Otherwise, the detec-tion accuracy could significantly drop because the genuine node (the replica) may be falsely regarded as the replica (genuine node). Thus, that XED and EDD do not need time synchroniza-tion among nodes to achieve detection works as a distinguished feature of our methods.

Centralized detection algorithms detect the replicas at the base station. To provide the information about the replicas, the base station should flood the revocation information into the en-tire network. Although there are distributed algorithms for the detection of node replication, in these algorithms, actually only few witness nodes can find the replicas in a communication-in-tensive detection period and these witness nodes are responsible for broadcasting the evidence of the replicas. If such a broad-cast is prohibited, only few nodes can be aware of the replicas, and the communication-intensive detection would need to be applied many times so that all of the nodes could

be aware of the replicas. Nevertheless, in XED and EDD, each node not only can detect the replicas by its own effort, but also can revoke the replica in a communication-efficient way within a short time period, as shown in Section IV.

One characteristic that deserves to be mentioned is that the solutions for static networks provide a detection algorithm that "can detect the replicas" without mentioning "when the network owner should apply the detection algorithm." The drawback is that the network owner has to be aware of the existence of the replicas. Afterward, the network owner resorts to the detection algorithms to identify the replicas. In contrast, our proposed algorithms automatically detect the replica anytime and any-where.

In the algorithms adopting the witness-finding strategy, the spatial distribution of witness nodes is usually an evaluation metric of the underlying detection algorithms. Ideally, it is uni-formly distributed over the sensing region. Nevertheless, this evaluation metric is specific for the algorithms adopting the wit-ness-finding strategy due to the need of witness nodes in their methods, and is not required in our proposed algorithms.

8. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, two replica detection algorithms for mobile sensor networks, XED and EDD, are proposed. Although XED is not resilient against collusive replicas, its detection framework, *challenge-and-response*, is considered novel as compared with the existing algorithms. Notably, with the novel *encounter-number* detection approach, which is fundamentally different from those used in the existing algorithms, EDD not only achieves balance among storage, computation, and com-munication overheads, which are all , but also possesses unique characteristics, including network-wide time synchro- nization avoidance and network-wide revocation avoidance, in
the detection of node replication attacks. In this project ourmain goal is detection of node replication attacks. After detecting the replication node, that ID is add block list and furthermore is again find another hop and send object accordingly it's fulfilling. In addition the detecting replication node will eliminate the entire networks. Note that we are not eliminate whole replication node for that particular replicated ID, we eliminate particular detecting node only. In that process we made last module that is "Eliminate Replicated Hop"

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 predistribution," IEEE Trans. Syst., Man, Cybern. C, Ap-plicat. Rev., vol. 37, no. 6, pp. 1246–1258, Nov. 2007.

[2] 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.

[3] G. Cormode and S. Muthukrishnan, "An improved data stream sum-mary the count-min sketch and its applications," J. Algorithms, vol. 55, no. 1, pp. 56–75, 2005.

[4] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "A randomized, ef-ficient, and distributed protocol for the detection of node replication at-tacks in wireless sensor networks," in Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc), Montreal, Canada, 2007, pp. 80–89.

[5] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Distributed de-tection of clone attacks in wireless sensor networks," IEEE Trans. De-pend. Secure Comput., vol. 8, no. 5, pp. 685–698, Sep./Oct. 2012.

[6] M. Conti, R. D. Pietro, and A. Spognardi, "Wireless sensor replica detection in mobile environment," in Proc. Int. Conf. Distributed Computing and Networking (ICDCN), Hong Kong, China, 2012, pp. 249–264.

[7] 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.

[8] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," Performance Evaluation, vol. 62, no. 1, pp. 210–228, 2005.

[9] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in Proc. IEEE Int. Conf. Computer Communications (INFOCOM), 2003, pp. 1976–1986.

[10] 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.
[11] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," Mobile Comput., pp. 153–181, 1996.
[12] R. A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.
[13] T. Karagiannis, J. L. Boudec, and M. Vojnovic, "Power law and expo-nential decay of inter contact times between mobile devices," in Proc. ACM Int. Conf. Mobile Computing and Networking (MobiCom), Mon-treal, Canada, 2007, pp. 183–194.
[14] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: A secure sensor network communication architecture," in Proc. Int. Conf. Infor-mation Processing in Sensor Networks (IPSN), Cambridge, MA, USA, 2007.
[15] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in Proc. Int. Conf. Informa-tion Processing in Sensor Networks (IPSN), Missouri, USA, 2008, pp. 245–256.