Probabilistic cells in Replica Node Detection in Wireless Sensor Networks

R. Sheeba[#1], A. Anish Prem Jani[#2]
[#]*Information Technology, Lord Jegannath College of Engineering and Technology*

Abstract— Wireless sensor networks (WSNs) are deployed in potentially hostile environments where enemies may be present, especially in the military. Since WSNs are mostly left without any notice since they are available in remote areas. Due to this unnoticed nature of wireless sensor networks, an attacker can easily capture and get access to the sensor nodes and replicate them, and then impose a variety of attacks with these replicas. These attacks are dangerous because they allow the attacker to extend the control over few nodes to much of the network. A fast and effective mobile replica node detection scheme using the localized multicast is proposed. Here, a scheme called P-MPC is used. The efficiency and security of this approach are evaluated theoretically. The results show that, compared to previous approaches proposed, this approach is more efficient in terms of communication and memory costs in large-scale sensor networks, and at the same time achieve a higher probability of detecting node replicas.

Keywords— Replica detection, mobile sensor networks, security, distributed protocol, efficiency

I. INTRODUCTION

Advances in robotics have made it possible to develop a variety of new architectures for autonomous wireless networks of sensors. Mobile nodes, essentially small robots with sensing, wireless communications, and movement capabilities, are useful for tasks such as static sensor deployment, adaptive sampling, network repair, and event detection [4]. These advanced sensor network architectures could be used for a variety of applications including intruder detection, border monitoring, and military patrols. In potentially hostile environments, the security of unattended mobile nodes is extremely critical. The attacker may be able to capture and compromise mobile nodes, and then use them to inject fake data, disrupt network operations, and eavesdrop on network communications. In this scenario, a particularly dangerous attack is the replica node attack [11], in which the adversary takes the secret keying materials from a compromised node, generates a large number of attacker-controlled replicas that share the compromised node's keying materials and ID, and then spreads these replicas throughout the network. With a single captured node, the adversary can create as many replica nodes as he has the hardware to generate. Note that replica nodes need not be identical robots; a group of static nodes can mimic the movement of a robot and other mobile nodes or even humans with handheld devices could be used. The only requirement is that they have the software and keying material to communicate on the network, all of which can be obtained from the captured node. The time and effort needed to inject these replica nodes into the network should be much less than the effort to capture and compromise the equivalent number of original nodes. The replica nodes are controlled by the adversary, but have keying materials that allow them to seem like authorized participants in the network. Protocols for secure sensor network communication would allow replica nodes to create pair wise shared keys with other nodes and the base station, thereby enabling the nodes to encrypt, decrypt, and authenticate all of their communications as if they were the original captured node. The adversary can then leverage this insider position in many ways. For example, he can simply monitor a significant fraction of the network traffic that would pass through these nodes. Alternately, he could jam legitimate signals from benign nodes or inject falsified data to corrupt the sensor's monitoring operation. A more aggressive attacker could undermine common network protocols, including cluster formation, localization, and data aggregation, thereby causing

continual disruption to network operations. Through these methods, an adversary with a large number of replica nodes can easily defeat the mission of the deployed network. A straightforward solution to stop replica node attacks is to prevent the adversary from extracting secret key materials from mobile nodes by equipping them with tamper-resistant hardware. We might expect such measures to be implemented in mobile nodes with security-critical missions. However, although tamper-resistant hardware can make it significantly harder and more time-consuming to extract keying materials from captured nodes, it may still be possible to bypass tamper resistance for a small number of nodes given enough time and attacker expertise. Since the adversary can generate many replicas from a single captured node, this means that replicating attacks are even more dangerous when compared with the possibility of compromising many nodes. We thus believe that it is very important to develop software-based countermeasures to defend mobile sensor networks against replica node attacks. Several software-based replica node detection schemes have been proposed for static sensor networks [3], [11]. The primary method used by these schemes is to have nodes report location claims that identify their positions and for other nodes to attempt to detect conflicting reports that signal one node in multiple locations. However, this approach requires fixed node locations; it cannot be used when nodes are expected to move.

Previously, [14] two distributed algorithms for detecting node replication in which the witness nodes for a node's location information are randomly selected among all the nodes in the network was proposed. In the Randomized Multicast algorithm each location has $\sqrt{n}$ witnessed nodes. Thus, in a network of n nodes, according to the Birthday Paradox, in the event of a node replication attack, at least one witness node is likely to receive conflicting location claims about a particular node. The communication costs of this protocol are $O(n^2)$ (for the entire network) and the memory requirements per node are $O(\sqrt{n})$. The Line-Selected Multicast exploits the routing topology of the network to select witnesses for a node's location and uses geometric probabilities to detect replicated nodes. It has a communication cost and memory requirements per node $O(\sqrt{n})$.Also, another replica detection protocol was proposed, i.e., RED [2]. Compared to Previous work [14], in RED each location has a smaller number of witnesses. The set of witnesses is uniformly chosen from the whole network due to the usage of a pseudorandom function, the inputs of which include the identity of the node, the number of locations (of witnesses) that have to be generated by any neighbor of this node that decides to forward the location claim, and a random number and which is changed per iteration. In other words, within each iteration the set of witnesses for any node is fixed and is known to anyone who has the knowledge of rand through either node compromise or sniffing the broadcast message containing the value of rand at the beginning of each iteration. Therefore, there exists a dilemma in selecting an appropriate value of the number of locations (of witnesses) that have to be generated so as to achieve the balance between efficiency and robustness against node compromise. In this paper, we present a novel distributed protocol for detecting node replication attacks that take a different approach for selecting witnesses for a node.

Recently, a mobile replica detection scheme based on the Sequential Probability Ratio Test (SPRT) [15] was proposed. Using, the fact that an uncompromised mobile node should never move at speeds in excess of the system-configured maximum speed. As a result, a benign mobile sensor node's measured speed will nearly always be less than the system-configured maximum speed as long as we employ a speed measurement system with a low error rate. On the other hand, replica nodes are in two or more places at the same time. This makes it appear as if the replicated node is moving much faster than any of the benign nodes, and thus the replica nodes' measured speeds will often be over the system-configured maximum speed. Accordingly, it is observed that a mobile node's measured speed is over the system-configured maximum speed, it is then highly likely that at least two nodes with the same identity are present in the network. However, if the system decides that a node has been replicated based on a single observation of a node moving faster than it should, we might get many false positives because of errors in speed measurement. Raising the speed threshold or other simple ways of compensating can lead to high false negative rates. To minimize these false positives and false negatives, the SPRT, a hypothesis testing method that can make decisions quickly and accurately was used. The SPRT is performed on every mobile node using a null hypothesis that the mobile node has not been replicated and an alternate hypothesis that it has been replicated. In using

the SPRT, the occurrence of a speed that is less than or exceeds the system-configured maximum speed will lead to acceptance of the null or alternate hypotheses, respectively. Once the alternate hypothesis is accepted, the replica nodes will be revoked from the network.

In our approach, which we call Localized Multicast, the witness nodes for a node identity are randomly selected from the nodes that are located within a geographically limited region (referred to as a cell). Our approach first deterministically maps a node's ID to one or more cells, and then uses randomization within the cell(s) to increase the resilience and security of the scheme. One major advantage of our approach is that the probability of detecting node replicas is much higher than that achieved in previous protocols [14].

II. P-MPC For Detecting Replica Nodes

A. Motivation

In this paper, we assume the existence of a monitoring mechanism that can detect a node compromising operating with a certain probability. Therefore, the larger number of nodes that an adversary attempts to compromise, the higher is the probability that the node compromising attack is detected, thereby triggering an automated protocol or human intervention for removing compromised nodes. However, in certain cases (e.g., when the number of nodes in a cell is relatively small), a determined adversary may be willing to take the risk of being detected in return for a high probability of controlling all the witness nodes for one or more identities. Another potential risk is that a smart adversary can take advantage of the knowledge that the destination cell for a given identity is deterministic and launch a blocking attack. Informally, after compromising a small set of sensors denoted as V, the adversary can generate replicas of members in V and deploy them in such a way that all the location claims of these replicas are forwarded through members of the V.

In the SDC approach, all the location claims are first forwarded from the neighbors of L to a deterministic cell. Therefore, there is a high probability that these forwarding paths intersect with each other. In particular, when L and the destination cell (i.e., cell C) are far from each other, there is a high probability that all the location claims will pass through one or a small set of nodes of size y. Therefore, the adversary only needs to compromise one or y nodes per replica so as to block the forwarding of a location claim. Hop-by-hop watchdog monitoring [12] may help mitigate this attack. However, it will fail if all or most of the neighbors of an intersection point are compromised. Even worse, the adversary can insert a replica in such way that its location claim will always be forwarded through a small set of compromised nodes. An example of blocking an attack against the SDC approach is shown in Fig. 1. Cell C1 and C2 are the deterministic cells for the identity IDC1 and IDC2 , respectively, and B is an area in which all the nodes have been compromised (referred to as a black hole). In this example, three replicas (i.e., L^1C1, L^2 C1 , and L^3 C1) claiming the same identity that is mapped to cell C1 are added to the network sequentially, with a certain time interval between any pair of consecutive joins. In the SDC approach, nodes en route between the replica and the deterministic cell do not store the location claim. As a result, as long as the location claims from different replicas do not arrive at the same time, forwarding nodes are not able to detect the conflicts. Finally, all the location claims are delivered to the black hole and blocked. In other words, adversaries can insert replicas without being detected. Note that the same black hole may be used to insert replicas for multiple identities. As shown in Fig. 1, two replicas (i.e., L^1C2 and L^2C2) claiming the same identity that is mapped to cell C2 are inserted into the network and their location claims are also blocked by the black hole B.

B. Description of the P-MPC Scheme

Like SDC, in the P-MPC scheme, a geographic hash function [15] is employed to map node L's identity to the destination cells. However, instead of mapping to a single deterministic cell, in P-MPC, the location claim is mapped and forwarded to multiple deterministic cells with various probabilities. Let C = {C1, C2, ..., Ci, ., Cv} denote the set of cells to which an identity (denoted as IDL) is mapped. Let pci denote the probability that the location claim of L is forwarded to cell Ci. Without loss of generality, in the rest of this paper, we assume that set C is sorted by pcis. The following two conditions should be satisfied while determining pcis:

1) $\sum_{i=1}^{v} p_{ci} = 1$ and 2) $p_{ci} \geq p_{cj}$ when i < j for I, j ∈ {1, 2, . . . , v}.

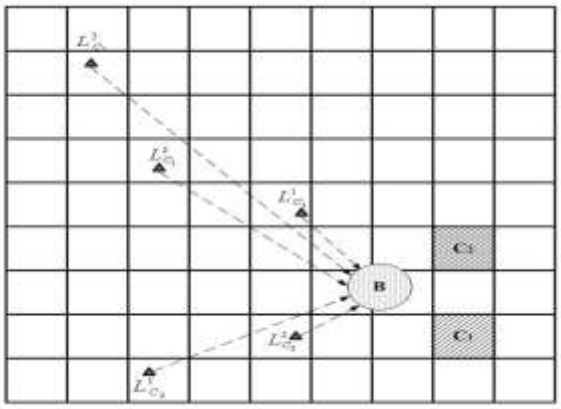

Fig.1 The blocking attacks.

The second condition is introduced to enhance the efficiency of the protocol. An example of P-MPC is shown in Fig. 2. When L broadcasts its location claim, each neighbor independently decides whether to forward the claim in the same way as the SDC scheme. Afterwards, each neighbor helping forward the claim first calculates the set of cells (i.e., C) to which L are mapped, based on a geographic hash function with the input of IDL. For example, by using a one way hash function H (), node L is mapped to the set of cells C= {C1, C2, Ci, . . . , Cv}, where $C_i = [H(ID_L \| i) mod(a-b)] + 1(i \in \{1, 2, \ldots, v\})$

Then, each neighbor that forwards the claim independently generates a random number $z \in [0, 1)$. Assume that j is the smallest amount that satisfies $z < \sum_{i=1}^{j} p_{ci} (j \in \{1, 2, \ldots, v\})$, this neighbor chooses the jth cell (i.e., Cj) as the destination cell for the location claim. For example, if z= 0.8 and the predetermined distribution of pci's is (pc1=50%, pc2 =25%, pc3 = 15%, and pc4 = 10%), the claim will be forwarded to cell C3. Once the location claim arrives at cell Cj, the sensor receiving it first verifies whether Cj is a member of C which can be calculated based on the geographic hash function and the identity listed in the claim message. In addition, this sensor needs to verify the validity of the signature on the location claim.

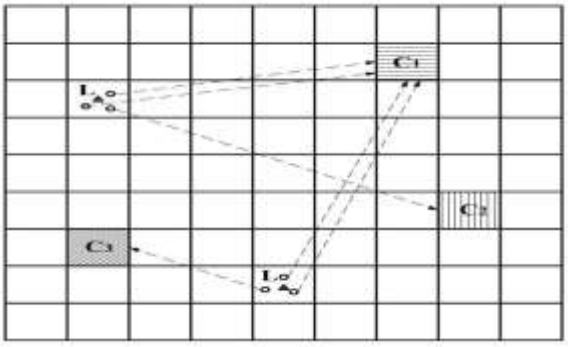

Fig. 2. The parallel multiple probabilistic cells approach

If both the verifications succeed, the claim is flooded within the cell and probabilistically stored at w nodes in the same manner as in the SDC scheme. For example, in Fig. 2, there are two replicas with the same identity in the network. In this example, an identity is mapped to three cells (i.e., C1, C2, C3) with different probabilities (i.e., pc1 > pc2 > pc3). The neighbors of one replica forward the location claims to cell C1 and C2, while the neighbors of the other replica forward the location claims to cell C1 and C3. Therefore, any witness node with cell C1 can detect the node replication.

III. Analysis Of The Parallel Multiple Probabilistic Cells Scheme

In this section, we analyze the security and efficiency of the P-MPC scheme. In addition, a summary of the communication cost and memory overhead of our approach and the algorithms proposed in [14] is shown at the end of this section.

A. *Security Analysis*

For simplicity, in this section we assume that the number of neighbors (r) forwarding the location claim is a fixed number. We assume that the adversary creates x-1 replicas of a given compromised node with id IDL and deploys them in the network. We assume that adversaries do not reposition the compromised node, l1, and the replicas are added in sequence from l2 to lx. Let pir denote the probability that the node replication attack is not detected by our scheme after the ith node with the same identity has been added to the network.

TABLE I

	p_{c1}	p_{c2}	p_{c3}	$1-p_{2r}$	$1-p_{3r}$
Set. I	80%	15%	5%	99.77%	100%
Set. II	70%	20%	10%	99.38%	100%
Set. III	50%	30%	20%	98.88%	99.98%

Detection Rates When There Are 2 or 3 Nodes with the Same Identity, Given Different Settings of the Distribution of Forwarding Probabilities.

1) *Detecting Replicas*

Let Cs1 denote the set of all combinations of choosing 1 to v - 1 elements from C, i.e., the set of cells to which IDL is mapped. If the node replication attack is not detected when the adversary adds replica l2 to the network, it implies that the location claims for l2 have been forwarded to a set of cells, none of which contains any node storing a location claim from l1. Let Ce1 denote a subset of the cells in C that do not store the

location claims of l1. Let pi; 1 denote the probability that the location claim of l1 is forwarded to all the cells in C except the cells in Ce1, which is an element of Cs1. Let pi, 2 denote the probability that the location claim of l2 is forwarded to any cell(s) in Ce1. Therefore, we have:

$$p_{2r} = \sum_{i=1}^{|C_{s1}|} p_{i,1} \cdot p_{i,2}.$$

Now, we consider further the case that the adversary adds l3 to the network. Let Cs1b denote the set of all the combinations of choosing 2 to v - 1 element from C. For a given Ce1 € Cs1b, let Cs2 denote all the combinations of choosing 1 to |Ce1|-1 elements from Ce1. We denote Ce2 as the set of cells that store the location claim from l2 but not l1, and Ce2 € Cs2. Let pi denote the probability that the location claim of l1 is forwarded to all the cells in C except the cells in Ce1, which is an element of Cs1b. Let pi, 1 denotes the probability that the location claim of l2 is forwarded only to all the cells in Ce2. Let pij, 2 denote the probability that the location claim of l3 is forwarded to any cell(s) in Ce1 except those in Ce2. Thus, we have:

$$p_{3r} = \sum_{i=1}^{|C_{s1b}|} \sum_{j=1}^{|C_{s2}|} p_i \cdot p_{ij,1} \cdot p_{ij,2}$$

Let r = 3 and v = 3. In Table 1, we show the estimated success rate of detecting node replications. According to Table 1 (where "Set" Is a short notation for "Setting"), the P-MPC scheme can achieve a very high replica detection rate, even when an identity is mapped to three destination cells. Moreover, we notice that the larger the differences between the probabilities pcis, the higher are pir.

TABLE III

	Set. A	Set. B	Set. C	Set. D	Set. E
Set. I	9.69e-04	2.04e-05	1.73e-06	6.39e-06	2.37e-07
Set. II	2.37e-04	5.08e-06	5.36e-07	5.11e-05	1.51e-05
Set. III	7.01e-05	1.60e-06	3.72e-07	7.01e-05	7.01e-05

Probability that the Adversary Controls All witnesses for a Given Identity after Compromising t_ Nodes in a Cell of Sizes in the P-MPC Scheme (s = 100, w = 5, t_Δ= 30)

2) Resilience against Node Compromise

Let $p_{ts}^{SDC}(t)$ and $p_{ts}^{P-MPC}(t)$ denote the functions that output the pts of the SDC scheme and the P-MPC scheme, respectively, when the number of the compromised nodes is t. Let $p_{tm}^{SDC}(t)$ and $p_{tm}^{P-MPC}(t)$ denote the functions that output the ptm of the SDC scheme and the P-MPC scheme, respectively, when the number of the compromised nodes is t. Assuming that the adversary's capability of compromising nodes is bounded by t_Δ, we have Pv $\sum_{i=1}^{v} t_i = t_\Delta$, where it is the number of nodes compromised in cell Ci. Let Ct1 denote the set of all the combinations of choosing 1 to v elements from C. For any element in Ct1 denoted as Cf1, the probability that the adversary controls all the witnesses of a given identity, when such a set of cells in C (i.e., Cf1) are chosen as the destination cell(s), is the product of all the individual probabilities ptss of the cells. Let pi denote the probability that exactly the cells in Cf1 are chosen as the destination cells by the r neighbors that forward the location claim. Let $p_{ts}^{SDC}(t_j)$ denote the pts of the jth cell of Cf1 when the number of nodes compromised in this cell is tj. Thus, $p_{ts}^{P-MPC}(t)$ can be calculated as follows:

$$p_{ts}^{P-MPC}(t) = \sum_{i=1}^{|C_{f1}|} \left(p_i \cdot \prod_{j=1}^{|C_{f1}|} p_{ts}^{SDC}(t_j) \right)$$

Note that in (7), |Ct1 | denotes the number of all the combinations of choosing 1 to v elements from C, while |Cf1 | denotes the number of cells contained in a chosen combination, i.e., Cf1. In additional, $p_{ts}^{SDC}(t_j) = 1$ when there is no witness in the jth cell of Cf1. Let r = 3 and v = 3. In Table 3, we show the estimated success rate that adversaries control all the witnesses under different compromising strategies (i.e., various distributions of ti) and probability distributions of the destination cells (i.e., pci) in the P-MPC scheme, when s = 100, w = 5, and t_Δ = 30. The settings on ti and pci are shown in Tables 4 and 5, respectively. From Table 3, we notice that the best strategy for adversaries is to compromise only nodes in the cell with the highest pci, i.e., setting A of ti, rather than spreading their limited capability of compromising nodes among multiple cells in C. Assuming that the adversary selects this optimal strategy, the larger the differences between pics, the larger is p_{ts}^{P-MPC} and thus the weaker the resilience of the scheme to node compromise. Compared to SDC, P-MPC is more robust to node compromise. Assuming that adversary's follow the best strategy just described, i.e., compromising only nodes in the cell with the highest pci, (7) can be converted into:

$$p_{ts}^{P-MPC}(t) = p_{c1}^{r} \cdot p_{ts}^{SDC}(t)$$

As a result, compared to the SDC approach, the success rate that adversaries control all the witnesses of a given identity is reduced by a factor of $1 - p_{c1}^{r}$. Unlike the SDC

scheme where each identity is mapped to only one cell, in P-MPC, each identity may be mapped to multiple cells. Since the cells for a given identity are determined by geographic hash functions, those cells are uniformly distributed. Therefore, on average for each cell, there are s identities choosing it with the probability pc1, pc2,..., pcv, respectively. Assuming that instead of spreading the limited capability of compromising nodes in multiple cells, adversaries only compromise the nodes in a given cell, we can calculate $p_{tm}^{P-MPC}(t)$ via (9).

$$p_{tm}^{P-MPC}(t) = \sum_{i=1}^{v} p_{ci}^{r} \cdot p_{tm}^{SDC}(t)$$

When the differences between pcis are high, e.g., Setting I in Table 5, $p_{tm}^{P-MPC}(t)$ can be approximated as $p_{c1}^{r} \cdot p_{tm}^{SDC}(t)$. In such cases, compared to the SDC scheme, the success rate that adversaries control all the witnesses for at least one identity is reduced by a factor of $1 - p_{c1}^{r}$ as well.

At P-MPC, even if adversaries compromise all the nodes in the cell to which the location claims are forwarded with the highest probability, i.e., pc1, node replication can still be detected by witnesses in the other cells. For example, assuming that pc1 = 80% and r = 3, the replica can still be detected with a probability of $1 - p_{c1}^{3} = 48.8\%$.

3) Denial-of-Service Attacks

Two possible Denial-of-Service (DoS) attacks against our approach are as follows: 1) An adversary inserts a large number of fake location claims into the network so as to exhaust the energy and computational resources of other nodes, who will verify the signatures included in the location claims according to the approach proposed. 2) If some of a node L's neighbors are controlled by the adversary, instead of choosing the destination cell based on the probabilistic distribution and the geographic hash function, the adversary may forward the location claim to as many cells as possible, leading to additional communication overhead when the claim is flooded within each cell.

TABLE IIIV

Settings on the Distribution of # of Compromised Nodes

	t_1	t_2	t_3
Set. A	30	0	0
Set. B	15	10	5
Set. C	10	10	10
Set. D	0	30	0
Set. E	0	0	30

In the first attack, in both SDC and MPC, any fake location claim would fail the verification process, and thus, will not be forwarded further. As to the latter, which is only applicable to MPC, if the destination cell chosen is not an element of C (i.e., The set of cells to which the given identity is mapped) or a neighbor forwards the same location claim to more than one cell, the attack would be detected by other neighbors of L, although it requires the neighbors to listen promiscuously. To avoid detection based on signature verifications, the best strategy for this type of DoS attack is to ignore the probability distribution being used by P-MPC for selecting destination cells, and let different neighbors choose different destination cells in C. However, as shown by our analysis, a small number of cells (v = 3) are sufficient for P-MPC to provide a high level of resilience against node compromise while ensuring a very high detection rate on node replication. Therefore, the effectiveness of this attack is limited.

TABLE V

$\sum_{i=1}^{v} p_{si}$ n Terms of Different Settings of n p_{ci}s $(v = 3)$

p_{c1}	p_{c2}	p_{c3}	$\sum_{i=1}^{v} p_{si}$
0.8	0.15	0.05	1.5205
0.7	0.2	0.1	1.732
0.5	0.3	0.2	2.02
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	2.1111

B. *Efficiency Analysis*

When analyzing the efficiency of the P-MPC scheme, we follow the same metrics employed in Section 5.2.

1) Communication Cost

Similar to the SDC scheme, the communication cost for P-MPC has two components: the cost of propagating the location claim to the cells chosen and the cost of flooding the claim within these cells, denoted as COfw and COfl, respectively. Assuming that in the P-MPC scheme there are on average r neighbors forwarding a location claim, the communication complexity of COfw is $O(r \cdot \sqrt{n})$ in P-MPC, if we assume that the neighbors.

TABLE VI

	Communication	Memory
Randomized Multicast	$O(n)$	$O(\sqrt{n})$
Line-Selected Multicast	$O(g \cdot p_f \cdot d \cdot \sqrt{n})$	$O(g \cdot p_f \cdot d \cdot \sqrt{n})$
RED	$O(g \cdot p_f \cdot d \cdot \sqrt{n})$	$O(g \cdot p_f \cdot d)$
SDC	$O(r \cdot \sqrt{n}) + O(s)$	w
P-MPC	$O(r \cdot \sqrt{n}) + O(s)$	w

Comparisons of Average Communication Cost and Memory Overhead of L forward the location claim independently and do not consider further optimizations, e.g., a node only forwards the location claims with the same identity and location information once within a certain time interval. The communication complexity of COfl in the P-MPC scheme can be estimated as follows: Since there are r neighbors of L forwarding the location claim, the probability that any cell in C (i.e., Ci) is chosen by at least one out of r neighbors is:

$$p_{si} = 1 - (1 - p_{ci})^r$$

Therefore, the complexity of COfl in the P-MPC scheme can be described as

$$O(s \cdot \sum_{i=1}^{v} p_{si}).$$

Table 6 shows the value of $\sum_{i=1}^{v} p_{si}$ in terms of different settings on pcis when v = 3.According to Table 6, the larger the differences between pcis, the smaller the extra overhead of flooding the location claim, when compared to the SDC scheme.

2) Memory Overhead

In a similar fashion, we can see that the the memory overhead of the P-MPC scheme is given by

$$s \cdot p_s \cdot \sum_{i=1}^{v} p_{si}$$

IV. CONCLUSIONS

In this paper, we proposed two variants of the Localized Multicast approach for distributed detection of node replication attacks in wireless sensor networks. Unlike the two randomized algorithms approach previously proposed [14], our approach combines deterministic mapping (to reduce communication and storage costs) with randomization (to increase the level of resilience to node compromise). Our theoretical analysis and empirical results show that, compared to previous algorithms, our schemes are more efficient in large-scale sensor networks, in terms of communication and memory costs. Moreover, the probability of replica detection in our approach is higher than that achieved in these two algorithms. Our preliminary analysis also shows that, our approaches are more robust than RED against selective node compromise, and the communication and memory overheads of our approaches are similar or slightly higher than that of RED. One of our future works is to simulate the RED protocol and then have a more detailed comparison of efficiency based on empirical results.

REFERENCES

[1] Conti. M, R.D. Pietro, L.V. Mancini, and A. Mei, "A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks," Sept. 2007..

[2] Ho. J, M. Wright, and S.K. Das, "Fast Detection of Replica Node Attacks in Mobile Sensor Networks Using Sequential Analysis," Apr. 2009.

[3] Ho. J, D. Liu, M. Wright, and S.K. Das, "Distributed Detection of Replicas with Deployment Knowledge in Wireless Sensor Networks," Ad Hoc Networks, vol. 7, no. 8, Nov. 2009.

[4] Hu. L and D. Evans, "Localization for Mobile Sensor Networks," Proc. ACM MobiCom, pp. 45-57, Sept. 2004.

[5] Jung. J, V. Paxon, A.W. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing," Proc. IEEE Symp. Security and Privacy, pp. 211-225, May 2004.

[6] Liu. A and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," Apr. 2008.

[7] PalChaudhuri. S, J.-Y.L. Boudec, and M. Vojnovi_c, "Perfect Simulations for Random Trip Mobility Models," Apr. 2005.

[8] Parno. B, A. Perrig, and V.D. Gligor, "Distributed Detection of Node Replication Attacks in Sensor Networks," May 2005

[9] Song. H, S. Zhu, and G. Cao, "Attack-Resilient Time Synchronization for Wireless Sensor Networks," Ad Hoc Networks, vol. 5, no. 1, pp. 112-125, Jan. 2007.

[10] Sun. H, P. Ning, C. Wang, A. Liu, and Y. Zhou, "TinySeRSync: Secure and Resilient Time Synchronization in Wireless Sensor Networks," Proc. 13th ACM Conf. Computer and Comm. Security (CCS), pp. 264-271, Oct. 2006.

[11] B. Zhu, V.G.K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks,"Proc. 23rd Ann. Computer Security Applications Conf. (ACSAC '07),2007.

[12] H. Sabbineni "Location-Aided Flooding: An Energy-Efficient Data Dissemination Protocol for Wireless Sensor Networks,"IEEE Trans. Computers, vol. 54, no. 1, Jan. 2005

[13] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses,"Proc. Third Int'l Symp. Information Processing in Sensor Networks , 2004

[14] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,".

[15] T.J.Kwon and M.Gerla, "Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks,"ACM SIGCOMM Computer Comm. Rev.,vol. 32, no. 1, pp. 44-56, 2002.