Efficient Distributed Detection of Node Replication Attacks in Sensor Networks

Bo Zhu
Concordia Institute for
Information Systems Engineering
Concordia University
zhubo@ciise.concordia.ca

Venkata Gopala Krishna Addada
Center for Secure Information Systems
George Mason University
vaddada@gmu.edu

Sanjeev Setia
Department of Computer Science
George Mason University
setia@cs.gmu.edu

Sushil Jajodia, Sankardas Roy
Center for Secure Information Systems
George Mason University
{jajodia, sroy1}@gmu.edu

Abstract

Wireless sensor nodes lack hardware support for tamper-resistance and are often deployed in unattended environments, thus leaving them vulnerable to capture and compromise by an adversary. In a node replication attack*, an adversary uses the credentials of a compromised node to surreptitiously introduce replicas of that node into the network. These replicas are then used to launch a variety of attacks that subvert the goal of the sensor application, and the operation of the underlying protocols. We present a novel distributed approach called* **Localized Multicast** *for detecting node replication attacks. We evaluate the performance and security of our approach both theoretically and via simulation. Our results show that Localized Multicast is more efficient than previous distributed approaches in terms of communication and memory costs. Further, in our approach, the probability of detecting node replicas is much higher than that achieved in previous distributed protocols.*

1 Introduction

A new set of security challenges arises in sensor networks due to the fact that current sensor nodes lack hardware support for tamper-resistance and are often deployed in unattended environments where they are vulnerable to capture and compromise by an adversary. A serious consequence of node compromise is that once an adversary has obtained the credentials of a sensor node, it can surreptitiously insert replicas of that node at strategic locations within the network. These replicas can be used to launch a variety of insidious and hard-to-detect attacks on the sensor application and the underlying networking protocols.

In a centralized approach for detecting node replication, when a new node joins the network, it broadcasts a signed message (referred to as a *location claim*) containing its location and identity to its neighbors. One or more of its neighbors then forward this location claim to a central trusted party [3] (e.g., the base station). With location information for all the nodes in the network, the central party can easily detect any pair of nodes with the same identity but at different locations. Hence, a distributed solution is desirable.

Distributed approaches for detecting node replications are based on location information for a node being stored at one or more *witness nodes* in the network. When a new node joins the network, its location claim is forwarded to the corresponding witness nodes. If any witness receives two different location claims for the same node identity (ID), it will have detected the existence of replica and can take appropriate actions to revoke the node's credentials.

The basic challenge for any distributed protocol for detecting node replicas is to minimize communication and per node memory costs while ensuring that the adversary cannot defeat the protocol. A protocol that deterministically maps a node's ID to a unique witness node would minimize communication costs and memory requirements per node, but would not offer much security because the adversary would need to compromise just a single witness node in order to be able to introduce a replica without detection.

Previously, Parno et al [9] presented two distributed algorithms for detecting node replications in which the witness nodes for a location are *randomly* selected among the nodes in the network. In the Randomized Multicast algorithm each location has $\sqrt{n}$ witness 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 for 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 of $O(n\sqrt{n})$ and memory requirements per node of $O(\sqrt{n})$.

In this paper, we present a novel distributed protocol for detecting node replication attacks that takes a different approach for selecting witnesses for a node. 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 Parno et al's protocols [9].

We describe and analyze two variants of the Localized Multicast approach: *Single Deterministic Cell* (**SDC**) and *Parallel Multiple Probabilistic Cells* (**P-MPC**). Both theoretical analysis and simulation results show that the Localized Multicast approach is more efficient than Parno et al's algorithms in terms of communication and memory costs, while providing a high level of compromise-resilience. Further, our approach also achieves a higher level of security in terms of the capability of detecting node replicas.

The rest of the paper is organized as follows. In Section 2, the system, network, and adversary model of our work are presented. Then, we propose two variants of the Localized Multicast approach in Section 3. Afterwards, the theoretic analysis on the security and efficiency of the Single Deterministic Cell scheme and the Parallel Multiple Probabilistic Cells scheme are presented in Section 4 and Section 5, respectively. The simulation results are shown in Section 6. In Section 7, we review previous research work related to detecting node replication in sensor networks. Finally, we draw our conclusion in Section 8.

2 Protocol Framework

2.1 System and Network Model

We consider a sensor network with a large number of low-cost nodes distributed over a wide area. In our approach, we assume the existence of a trusted base station, and the sensor network is considered to be a geographic grid, each unit of which is called a cell. Sensors are distributed uniformly in the network. New sensors may be added into the network regularly to replace old ones.

Each node is assigned a unique identity and a pair of identity-based public and private keys[1] by an offline *Trust Authority* (**TA**). In identity-based signature schemes like [5], the private key is generated by signing its public key (usually a hash on its unique identity) with a master secret held only by the TA. In other words, to generate a new identity-based key pair, cooperation from the TA is a must. Therefore, we assume that adversaries cannot easily create sensors with new identities in the sense that they cannot generate the private keys corresponding to the identities claimed and thus fail to prove themselves to the neighbors during the authentication of the location claims.

We require that, when a node joins into the network, it needs to generate a signed location claim and broadcast the claim to its neighbors. Only when the location claim is successfully verified, it will then be accepted as a valid network member.

2.2 Adversary Model

In this paper, we assume that the major goal of adversaries is to launch node replication attacks. To achieve this goal, we assume that adversaries may launch both passive attacks (e.g., eavesdropping on network traffic) and active attacks (e.g., modifying and replaying messages or compromising sensors), and the information obtained from the former can be used to enhance the effectiveness of the latter. For example, by sniffing the traffic, adversaries may deduce certain information about the witness nodes, which could help them evaluate the potential benefit of compromising a given node and the risk of being detected while launching the node replication attack at a given location.

We assume the existence of some monitoring mechanism that can detect a node compromising operation with a certain probability. We also assume that adversaries are rational, and thus may try to avoid triggering any automated protocol (e.g. SWATT [11]) that sweeps the network to remove compromised nodes, or drawing human attention or intervention while launching the attacks.

2.3 Notation

In Table 1, we list notations/symbols used in this paper.

3 The Localized Multicast Approach for Detecting Node Replications

We have designed two variants of the Localized Multicast approach, specifically *Single Deterministic Cell* (**SDC**)

[1] Recent work [6, 4] shows that public key algorithms are practical on new sensor hardware. In addition, similar to [9], we can use symmetric key cryptography instead to lower down the computation cost, at the cost of large communication overhead.

n	The number of sensors in the network
s	The number of sensors in a cell
L	The node sending the location claim
ID_i	The identity of a sensor i
l_i	The location information of a sensor i
d	The number of L's neighbors
p_f	The probability that any neighbor of L decides to forward the location claim from L
r	The number of L's neighbors that forward the location claim from L
w	The number of the witness nodes that store the local claim from L
p_s	The probability that a sensor in the cell stores the location claim
t	The number of sensors that have been compromised by adversaries
x	The number of sensors with the same identity (including the compromised sensor and its replicas)
PK_i, SK_i	The public key and the private key of a sensor i
$H()$	A collision-free one-way hash function
$SIG_{SK}(M)$	A message M is signed by a key SK

Table 1. Notation and Symbols

and *Parallel Multiple Probabilistic Cells* (**P-MPC**).

3.1 Single Deterministic Cell

In the Single Deterministic Cell scheme, a geographic hash function [10] is used to uniquely and randomly map node L's identity to one of the cells in the grid. For example, given that the geographic grid consists of $a \times b$ cells, a cell at the a'th row and the b'th column (where $a' \in \{1, \ldots, a\}$, $b' \in \{1, \ldots, b\}$) is uniquely identified as c (where $c = a' \cdot b + b'$). By using a one-way hash function $H()$, node L is mapped to a cell D, where $D = [H(ID_L) \; mod \; (a \cdot b)] + 1$.

The format of the location claim is $[ID_L, \; l_L, \; SIG_{SK_L}(H(ID_L||l_L))]$, where $||$ denotes the concatenation operation and l_L is the location information of L, which can be expressed using either the two-dimension or three-dimension coordinate.

When L broadcasts its location claim, each neighbor first verifies the plausibility of l_L (e.g., based on its location and the transmission range of the sensor) and the validity of the signature in the location claim. In identity-based signatures schemes [5], only a signature generated with the private key corresponding to the identity claimed can pass the validation process. Thus, adversaries cannot generate valid signatures unless they compromise the node with that identity.

Each neighbor independently decides whether to forward the claim with a probability p_f. If a neighbor plans to forward the location claim, it first needs to execute a geographic hash function [10] to determine the destination cell, denoted as D. The location claim is then forwarded towards cell D.

Once the location claim arrives at cell D, the sensor receiving the claim first verifies the validity of the signature, and then checks whether cell D is indeed the cell corresponding to the identity listed in the claim message based on the geographic hash function. If both the verifications succeed, the location claim is flooded within cell D. Each node in the cell independently decides whether to store the claim with a probability p_s. Note that the flooding process is executed only when the first copy of the location claim arrives at cell D, and the following copies are ignored. As a result, we have $w = s \cdot p_s$.

Whenever any witness receives a location claim with the same identity but a different location compared to a previously stored claim, it forwards both location claims to the base station. Then, the base station will broadcast a message within the network to revoke the replicas.

Compared to the Random Multicast and Line-Selected Multicast algorithms, a major advantage of SDC is that it ensures 100% success rate for detecting any node replication, as long as the location claim is successfully forwarded towards cell D and stored by at least one node in the cell.

An important limitation on the Random Multicast and Line-Selected Multicast algorithms is that both the communication/memory overhead and the security (in terms of the success rate of detecting node replications) of the two algorithms are tightly related to the number of witnesses (w). On the one hand, the larger w is, the higher the communication and memory overhead. On the other hand, the smaller w is, the lower the success rate of detecting node replication. To ensure a high success rate of detecting node replication, w has to be $O(\sqrt{n})$.

In contrast, in the SDC scheme the communication cost and memory overhead are related to the number of neighbors that forward a location claim (i.e., $r = d \cdot p_f$) and the number of the witnesses (i.e., $w = s \cdot p_s$), respectively. In addition, the success rate of detecting node replication is independent of w when $w \geq 1$. Moreover, the randomization against potential node compromise and low memory overhead are achieved through flooding the location claim within the destination cell while storing it on only a small number of randomly chosen nodes. Assuming that the capability of the adversary (in terms of the number of nodes that can be compromised without being detected) is limited, by appropriately choosing the cell size (s) and p_s, the probability that adversaries control all the witnesses for an identity is negligible. Consequently, SDC can achieve lower com-

munication costs by setting r to a small value, and at the same time ensure low memory overhead and good security (i.e. a high success rate of detecting node replication and high level of resilience against potential node compromise), by choosing an appropriate value for w (s and p_s actually). A detailed analysis of the security and efficiency achieved in SDC is presented in Section 4.

3.2 Parallel Multiple Probabilistic Cells

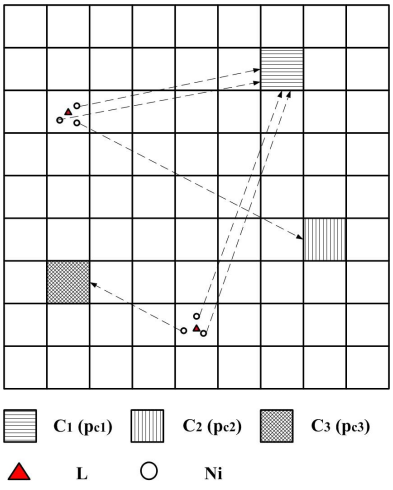

Figure 1. The Parallel Multiple Probabilistic Cells Approach

3.2.1 Description of The P-MPC Scheme

Like SDC, in the P-MPC scheme, a geographic hash function [10] is employed to map node L's identity to the destination cells. However, as shown in Figure 1, instead of mapping to single deterministic cell, in P-MPC the location claim is mapped and forwarded to multiple deterministic cells with various probabilities.

Let $C = \{C_1,\ C_2, \ldots, C_i, \ldots, C_v\}$ denote the set of cells to which a location claim (actually, the identity of the sender) is mapped. Let p_{ci} denote the probability that the location claim is forwarded to cell C_i. The following two conditions should be satisfied while determining p_{ci}'s: (i) $\sum_{i=1}^{v} p_{ci} = 1$; (ii) $p_{ci} \geq p_{cj}$ when $i < j$, for $i,\ j \in \{1,\ 2, \ldots,\ v\}$.

When L broadcasts its location claim, each neighbor independently decides whether to forward the claim in the same way as in the SDC scheme. The neighbors that forward the claim can determine the destination cell based on a geographic hash function and the predetermined probabilistic distribution of p_{ci}'s. More specifically, the neighbors first calculate the set of cells (C) to which the identity of the sender are mapped, based on a geographic hash function with the input of ID_L. Then, each neighbor that forwards the claim independently generates a random number $z \in [0,\ 1)$. Assume that j is the smallest number that satisfies $z < \sum_{i=1}^{j} p_{ci}$ ($j \in \{1,\ 2, \ldots,\ v\}$), this neighbor chooses the jth cell (i.e., C_j) as the destination cell for the location claim.

Once the location claim arrives at cell C_j, the sensor receiving it first verifies whether C_j 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 in the location claim. 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.

4 The Single Deterministic Cell Scheme

In this section, we theoretically analyze the security and efficiency of the Single Deterministic Cell scheme.

4.1 Security Analysis

The metrics used to evaluate the security of the SDC scheme are: 1) the probability of detecting node replication when adversaries put x replicas (including the compromised node) with the same identity into the network, which is denoted as p_{dr}; 2) the probability that adversaries control all the witnesses for a given identity after compromising t nodes, which is denoted as p_{ts}.

Same as [9], for the theoretical analysis in Section 4 and Section 5, we assume that there are r ($= d \cdot p_f$) neighbors forwarding L's location claim. Also, we assume that there are w ($= s \cdot p_s$) witnesses per destination cell storing L's location claim. Since $1 \geq p_f > 0$ and $1 \geq p_s > 0$, we have $r > 0$ and $w > 0$.

4.1.1 Detecting Replicas

Unlike the Random Multicast and Line-Selected Multicast algorithms, where the nodes storing the copies of a location claim are chosen randomly from the whole network, in SDC such nodes are chosen randomly from a small subset of all the nodes in the network, i.e., the nodes in the destination cell determined by the geographic hash function. In addition, since the location claim will be flooded within the destination cell, the SDC scheme can always detect any pair of nodes claiming the same identity. In other words, $p_{dr} = 100\%$ in SDC, when $r > 0$ and $w > 0$.

4.1.2 Resilience against Node Compromise

In SDC, witness nodes are chosen randomly from the nodes of a given cell instead of the whole network as in the Ran-

domized Multicast algorithm [9]. Therefore, assuming that the adversary's capability of compromising nodes is limited, intuitively in SDC the probability that an adversary can compromise all the witness nodes storing the location claim of a given identity, i.e. p_{ts}, is higher than that of the Randomized Multicast algorithm. However, we argue that by appropriately choosing the parameters (i.e., s and p_s), we can limit p_{ts} to a very small value, even if the adversaries can compromise a small fraction of the nodes in cell D.

Assuming that the adversary has compromised t nodes in cell D, p_{ts} can be calculated as follows:

$$p_{ts} = \frac{\binom{s-w}{t-w}}{\binom{s}{t}} = \frac{(t-w+1)(t-w+2)\cdots t}{(s-w+1)(s-w+2)\cdots s}, \quad (1)$$

where $t \geq w$.

In Figure 2, we plot the probability that an adversary controls all the witness nodes of a given identity (i.e. p_{ts}) under different settings, when the cell size is 100 (i.e. $s = 100$). Figure 2 shows that when w (in fact s and p_s) is chosen appropriately, p_{ts} is negligible, even if the adversary can compromise a large number of nods in the cell. In particular, when $w = 20$ and $t = 60$, p_{ts} is only 7.82×10^{-6}. Even if w is chosen as a relative small number, e.g. 5, the adversary still needs to compromise around 65 out of 100 nodes in the cell to achieve a success rate of nearly 11%.

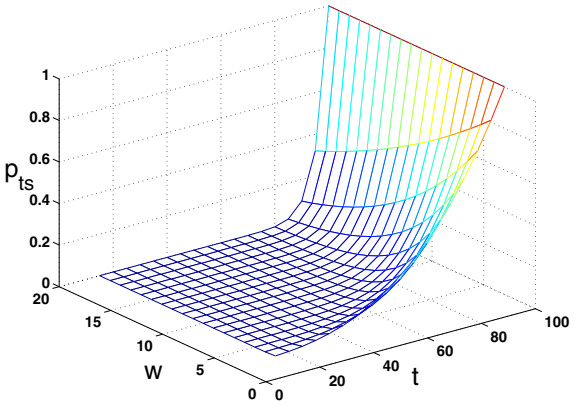

Figure 2. p_{ts} under different w and t ($s = 100$)

4.2 Efficiency Analysis

The metrics used to evaluate the efficiency of the SDC scheme include:

a) the average number of the packets sent and received while propagating the location claim, which is denoted as n_f.

b) the average number of the copies of the location claims stored on a sensor, which is denoted as n_s.

The former is to measure the communication cost, while the latter is to estimate the memory overhead. We do not explicitly consider the computation cost (i.e., verifying that the location claim is generated by an entity which holds the private key corresponding to the identity listed in the claim), since every forwarding node needs to execute such a verification and thus it is proportional to the communication cost. In other words, the higher is the communication cost, the higher is the computation cost.

4.2.1 Communication Cost

The communication cost of the SDC scheme has two components: the cost of forwarding the location claim to the destination cell (denoted as CO_{fw}) and the cost of flooding the location claim within the destination cell (denoted as CO_{fl}). The communication complexities of these two operations are $O(d \cdot p_f \cdot \sqrt{n})$ and $O(s)$ respectively.

4.2.2 Memory Overhead

SDC has the memory overhead of $O(w)$, where $w = s \cdot p_s$. As shown in Section 4.1, a relative small value of w, e.g. between 10 to 15 when $s = 100$, is sufficient to ensure security against node compromise. Therefore, the memory overhead of the SDC scheme is much lower than those of the Random Multicast algorithm and the Line-Selected Multicast algorithm which are of order $O(\sqrt{n})$ or higher. [2].

5 The Parallel Multiple Probabilistic Cells Scheme

In this section, we theoretically 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 [9] is shown at the end of this section.

5.1 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 ID_L and deploys them in the network. We assume that adversaries do not re-position the compromised node, l_1, and the replicas are added in sequence from l_2 to l_x. We 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 as p_{ir}. For analyzing the security of the P-MPC scheme, we use the same metrics employed in Section 4.1, except that we replace the metric p_{dr} with p_{ir}.

[2] Please refer to Section 5.3 for the more detailed comparison.

5.1.1 Detecting Replicas

Let C_{s1} denote the set of all the combinations of choosing 1 to $v-1$ elements from C, i.e., the set of the cells to which ID_L is mapped. If the node replication attack is not detected when the adversary adds replica l_2 to the network, this implies that the location claims for l_2 were forwarded to a set of cells that do not have any nodes that store the previous location claims of l_1.

Let C_{e1} denote a subset of the cells in C that do not store the location claims of l_1. Let $p_{i,1}$ denote the probability that the location claim of l_1 is forwarded to all the cells in C except the cells in C_{e1}, which is an element of C_{s1}. Let $p_{i,2}$ denote the probability that the location claim of l_2 is forwarded to any cell(s) in C_{e1}. Therefore, we have:

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

Now, we consider one step further the case that the adversary adds l_3 to the network. Let C_{s1b} denote the set of all the combinations of choosing 2 to $v-1$ elements from C. For a given $C_{e1} \in C_{s1b}$, let C_{s2} denote all the combinations of choosing 1 to $|C_{e1}|-1$ elements from C_{e1}. We denote C_{e2} as the set of cells that store the location claim from l_2 but not l_1, and $C_{e2} \in C_{s2}$. Let p_i denote the probability that the location claim of l_1 is forwarded to all the cells in C except the cells in C_{e1}, which is an element of C_{s1b}. Let $p_{ij,\,1}$ denote the probability that the location claim of l_2 is forwarded only to all the cells in C_{e2}. Let $p_{ij,2}$ denote the probability that the location claim of l_3 is forwarded to any cell(s) in C_{e1} except those in C_{e2}. Thus, we have:

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

Let $r = 3$ and $v = 3$. In Table 2, we show the estimated successful rate of detecting node replications under different settings of p_{ci} according to Equation (2) and (3). According to Table 2, 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 p_{ci}'s, the higher is p_{ir}.

	p_{c1}	p_{c2}	p_{c3}	p_{2r}	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%

Table 2. Successful Rate of Detecting Node Replications under Different Settings of p_{ci}

5.1.2 Resilience against Node Compromise

Let $p_{ts}^{SDC}(t)$ and $p_{ts}^{P-MPC}(t)$ denote the functions that output the p_{ts} 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 $\sum_{i=1}^{v} t_i = t_\Delta$, where t_i is the number of nodes compromised in cell C_i.

Let C_{t1} denote the set of all the combinations of choosing 1 to v elements from C. For any element in C_{t1} denoted as C_{f1}, the probability that the adversary controls all the witnesses of a given identity, when such a set of cells in C (i.e., C_{f1}) are chosen as the destination cell(s), is the product of all the individual probabilities p_{ts}'s of the cells. Let p_i denote the probability that exactly the cells in C_{f1} are chosen as the destination cells by the r neighbors that forward the location claim. Let $p_{ts}^{SDC}(t_j)$ denote the p_{ts} of the jth cell of C_{f1} when the number of nodes compromised in this cell is t_j. Thus, $p_{ts}^{P-MPC}(t)$ can be calculated as follows:

$$p_{ts}^{P-MPC}(t) = \sum_{i=1}^{|C_{t1}|} (p_i \cdot \prod_{j=1}^{|C_{f1}|} p_{ts}^{SDC}(t_j)) \tag{4}$$

Note that in Equation (4), $|C_{t1}|$ denotes the number of all the combinations of choosing 1 to v elements from C, while $|C_{f1}|$ denotes the number of cells contained in a chosen combination, i.e. C_{f1}. In additional, $p_{ts}^{SDC}(t_j) = 1$ when there is no witness in the jth cell of C_{f1}.

Let $r = 3$ and $v = 3$. In Table 3, we show the estimated successful rate that adversaries control all the witnesses under different compromising strategies (i.e. various distributions of t_i) and probability distributions of the destination cells (i.e. p_{ci}) in the P-MPC scheme, when $s = 100$, $w = 5$, and $t_\Delta = 30$. The settings on t_i and p_{ci} are shown in Table 4 and Table 5, respectively.

	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

Table 3. p_{ts}^{P-MPC} under Different Settings of p_{ci} and t_i ($s = 100$, $w = 5$, $t_\Delta = 30$)

From Table 3, we notice that the best strategy for adversaries is to compromise only nodes in the cell with the highest p_{ci}, i.e. setting A of t_i, 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 p_{ci}'s, the larger is p_{ts}^{P-MPC} and thus the weaker the resilience of the scheme to node compromise.

	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

Table 4. Settings of t_i

	p_{c1}	p_{c2}	p_{c3}
Set. I	0.8	0.15	0.05
Set. II	0.5	0.3	0.2
Set. III	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Table 5. Settings of p_{ci}

Compared to SDC, P-MPC is more robust to node compromise. Assuming that adversaries follow the best strategy just described, i.e. compromising only nodes in the cell with the highest p_{ci}, Equation (4) can be converted into:

$$p_{ts}^{P-MPC}(t) = p_{c1}^{r} \cdot p_{ts}^{SDC}(t) \quad (5)$$

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 $1 - p_{c1}^{r}$.

5.2 Efficiency Analysis

When analyzing the efficiency of the P-MPC scheme, we follow the same metrics employed in Section 4.2.

5.2.1 Communication Cost

Similar to the SDC scheme, the communication cost for P-MPC has two components: the cost for propagating the location claim to the cells chosen and the cost for flooding the claim within these cells, denoted as CO_{fw} and CO_{fl} respectively.

Assuming that in the P-MPC scheme there are on average r neighbors forwarding a location claim, the communication complexity of CO_{fw} is $O(r \cdot \sqrt{n})$ in P-MPC, if we assume that the neighbors 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 CO_{fl} 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., C_i) is chosen by at least one out of r neighbors is:

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

Therefore, the complexity of CO_{fl} 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 p_{ci}'s, when $v = 3$. According to Table 6, the larger the differences between p_{ci}'s, the smaller the extra overhead of flooding the location claim, when compared to the SDC scheme.

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

Table 6. $\sum_{i=1}^{v} p_{si}$ in terms of different settings on p_{ci}'s ($v = 3$)

5.2.2 Memory Overhead

In a similar fashion, we can see that the the memory overhead of the P-MPC scheme is given by $s \cdot p_s \cdot \sum_{i=1}^{v} p_{si}$.

5.3 Summary

Before presenting empirical results in Section 6, in Table 7, we summarize the average communication cost and memory overhead per node of the two variants of the Localized Multicast approach together with the two multicast algorithms proposed in [9], i.e. Randomized Multicast and Line-selected Multicast. In Table 7[3], we denote the number of lines selected in the Line-Selected Multicast algorithm and the number of the witness nodes storing the location claim for a given identity in our approach as f and w, respectively.

According to the analysis in Section 4 and Section 5, we know that r can be set to a value smaller than f while still ensuring higher success rate of detecting replicas. Therefore, the CO_{fw} of either SDC or P-MPC is smaller than the corresponding communication cost of the Line-selected Multicast algorithm. However, our approach has the extra overhead of flooding the location claim within one or more cells, i.e. CO_{fl}.

Note that for both SDC and P-MPC, the lower bound of the cell size is determined by the security requirements. Once the cell size and the flooding algorithm within the cell are chosen, CO_{fl} is fixed and independent of the network size. According to Table 7, we know that the extra overheads of the Random Multicast and the Line-Selected Multicast algorithms over CO_{fw} of our approach can be described as $(\sqrt{n} - r) \cdot S$ and $(f - r) \cdot S$ respectively, where S denotes the average communication cost of forwarding a packet between a randomly chosen pair of nodes in the network. S is tightly related to the network size (i.e., the complexity of S is $O(\sqrt{n})$) and the network topology (i.e.,

[3]By importing a parameter f, compared to [9], we give a more accurate evaluation on the complexity of the communication cost of the two algorithms proposed in [9].

for the same network size, S under an irregular topology is higher than that under a regular uniform topology). Consequently, our schemes are more scalable and less sensitive to irregular topologies, when compared to the two algorithms proposed in [9].

The analysis in Section 4.1 and Section 5.1 shows that it is sufficient to choose a small value for w to resist node compromise, and thus our approaches provide far better memory efficiency, compared to the Randomized Multicast and Line-selected Multicast algorithms, especially when the network size is very large.

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

Table 7. Comparisons of Average Communication Cost and Memory Overhead

6 Evaluation

We evaluated the performance and security of our schemes and those proposed by Parno et al via extensive simulations. To enable a fair comparison, we used the same simulation methodology and simulation code that was used in Parno et al's study [9].

6.1 Metrics

We used the following metrics to compare the schemes:

- **Communication Overhead:** We measured the total number of packets sent and received for running the replica detection algorithm when n nodes are added to the network. We denote this metric as n_f.
- **Success rate in detecting replicas** We measured the probability of detecting a replica, when there are two sensors with the same identity in the network, i.e. p_{2r}.

6.2 System and Network Models

As in the Parno et al study, we considered both uniform and irregular network topologies. In the uniform topology, nodes are randomly distributed within a 500×500 square. The network size (n) varies between 1000 to 10000. We assume a bidirectional communication model, and adjust the transmission range so that the average number of neighbors of a sensor (d) is 40. We also considered six irregular topologies, i.e.,'Thin H', 'Thin Cross', 'S', 'Large Cross', 'L', and 'Large H' with the same density, i.e. $d = 40$. As in Parno et al's study [9], these topologies are generated as the sub-regions of the regular topology ($n = 10000$).

For the Random Multicast and Line-Selected Multicast algorithms, we use the same settings as in [9], expect for p_f in the Line-Selected Multicast algorithm. More specifically, for the former, we set the number of sensors storing a given location claim to $\sqrt{n}$, i.e. $w = \sqrt{n}$; for the latter, we set the number of lines as 6, i.e. $f = 6$. As to p_f in the Line-Selected Multicast algorithm, it is set to $1/d$ in [9], and each forwarding node randomly picks f destinations. In our simulation, we set $p_f = f/d$, and each forwarding node randomly picks only one destination. Given the same density, compared to the former, the latter has a lower probability that there is no neighbor forwarding the location claim. As a result, compared to [9], in our simulation the Line-Selected Multicast algorithm has a higher success rate of detecting node replication, as shown in Section 6.3.2.

For both SDC and P-MPC, we set $p_f = 3/d$. Besides that, for P-MPC, we use Setting I in Table 5 as the setting of p_{ci}'s in the simulation. Namely, $v = 3$, and p_{c1}, p_{c2}, and p_{c3} are 80%, 15%, and 5%, respectively.

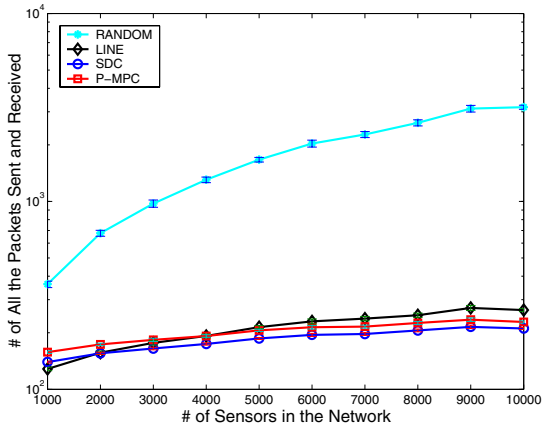

Figure 3. Communication Overhead of SDC, P-MPC, Random Multicast, and Line-Selected Multicast for Uniform Topologies.

6.3 Simulation Results

6.3.1 Communication Overhead

The figures below show the 95% confidence intervals of the reported metric. In Figure 3, we compare the communication costs of our two schemes with the two algorithms proposed in [9] for uniform topologies. As shown in Figure 3, the Random Multicast algorithm has the highest communication costs under all the settings. Among the remaining schemes, SDC has the lowest communication overhead,

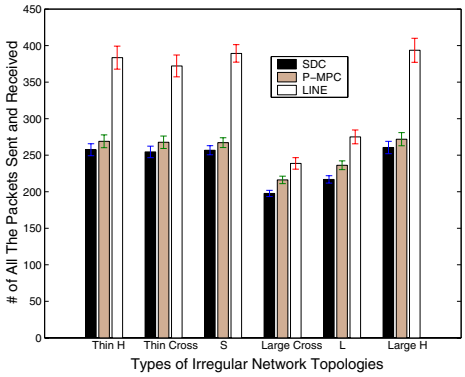

Figure 4. Communication Overhead of SDC, P-MPC, and Line-selected Multicast for Irregular Network Topologies

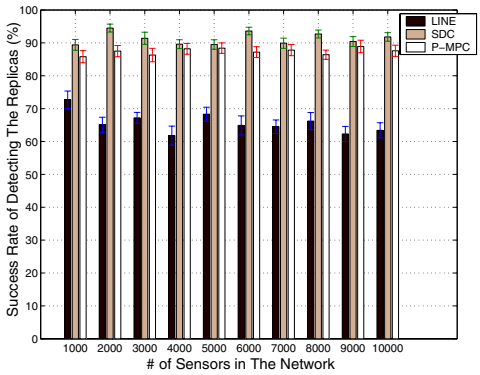

Figure 5. Success Rate of Detecting Replicas in SDC, P-MPC, and Line-Selected Multicast

though the differences between SDC, P-MPC, and Line-selected Multicast are relatively small. As the network size increases, P-MPC and SDC have lower overhead than Line-selected Multicast. Figure 3 shows that SDC and P-MPC have lower communication overheads than Line-Selected Multicast when $n \geq 2000$ and $n \geq 4000$ respectively.

In Figure 4, we compare the communication costs of our two schemes with the two algorithms proposed in [9] for irregular topologies. In comparison to Line-selected Multicast, both SDC and P-MPC show much stronger adaptability for irregular network topologies. Under all the irregular topologies, the n_f's of our two schemes are smaller than that of the Line-selected Multicast algorithm. In particular, under the 'Thin H', 'Thin Cross', 'S', and 'Large H' topologies, the advantage of our two schemes over the Line-selected Multicast algorithm is much higher than that under the regular topology ($n = 10000$). More specifically, under these four topologies, SDC's and P-MPC's advantage over the Line-selected Multicast algorithm (in terms of the communication cost) is 149% to 181% and 238% to 296%, respectively, higher than that under the regular topology ($n = 10000$).

6.3.2 Replica Detection Success Rate

Due to the high cost of the Random Multicast algorithm, we only consider SDC, P-MPC, and the Line-Selected Multicast algorithm while comparing the success rates of detecting node replication (i.e. p_{2r}).

Figure 5 shows that, compared to the Line-Selected Multicast algorithm, both of our algorithms have much higher success rates of detecting node replication. More specifically, on average, the success rates of SDC and P-MPC in detecting node replication are 25.64% and 21.77% higher than that of the Line-Selected Multicast algorithm, respectively.

We notice that, however, the success rates of SDC range from 89.4% to 94.5%, which are lower than the expected value (i.e., 100%) according to the theoretical analysis in Section 4.1. It is due to two reasons. The main reason is that, each neighbor decides whether to forward the location claim independently, and thus there exists a probability that no neighbor forwards the location claim. As a result, SDC fails to detect a node replication attack, if for any of the two replicas no neighbor forwards its location claim. In addition, when p_s is too small, there exists a probability that no node within cell D stores the location claim, which may also result in SDC's failure in detecting node replication. In the simulation, we set $p_s = 0.2$, and thus the second reason only has a negligible effect.

Due to the same reason, the simulation results about P-MPC's success rates of detecting node replication are lower than the expected value according to the theoretical analysis in Section 5.1.

7 Related Work

As discussed in previous sections, Parno et al. [9] were the first to propose distributed algorithms for detecting node replication attacks in sensor networks. In Sections 6, we have compared the performance and effectiveness of our approaches to their schemes.

Once a compromised node is detected in a sensor network, its credentials need to be revoked. The revocation process can be initiated by a central party, e.g., a base station [3], which broadcast a revocation message containing all the information (e.g. identity and pre-distributed keys held) about the revoked node. Alternatively, a group of sensors in the vicinity of the compromised node can cooperate to revoke it. In [1], Chan et al. proposed a distributed

quorum-based revocation mechanism that does not involve a base station. There has also been extensive work on key revocation for broadcast encryption [7] and multicast content distribution [12].

An attack that is superficially similar to node replication is the Sybil attack [2]. In this attack, single physical adversary can generate a number of virtual identities and falsely claim to be a set of non-existent nodes. Douceur [2] proposed the use of a few schemes in which the potential Sybil users are challenged to solve some resource-intensive task that can only be accomplished by multiple real-world users but will be impractical for a Sybil source. In contrast, in node replication attacks, single adversary can generate a number of physical nodes with the same identity and put them at different locations in the network. In other words, each replica is a real physical node, instead of a virtual one. As a result, the detection mechanism proposed in [2] fails to detect node replication. In [8], Newsome et al proposed a few mechanisms for detecting Sybil attacks in sensor networks, among which only the centralized node registration mechanism can be used to detect node replication.

8 Conclusion

We have discussed two variants of the Localized Multicast approach for distributed detection of node replication attacks in wireless sensor networks. 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 our schemes are more efficient than previous distributed approaches in terms of communication and memory costs. Moreover, in our approach, the probability of detecting node replicas is much higher than that achieved in previous distributed protocols.

References

[1] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *Proceedings of 2003 IEEE Symposium on Research in Security and Privacy*, 2003.

[2] John R. Douceur. The sybil attack. In *Proceedings of The First International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pages 251–260, 2002.

[3] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *The 9th ACM Conference on Computer and Communication Security*, pages 41–47, 2002.

[4] Gunnar Gaubatz, Jens-Peter Kaps, and Berk Sunar. Public key cryptography in sensor networks–revisited. In *Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), LNCS 3313*, pages 2–18, 2004.

[5] Florian Hess. Efficient identity based signature schemes based on pairings. In *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography (SAC'02)*, pages 310–324, 2002.

[6] David J. Malan, Matt Welsh, and Michael D. Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *Proceedings of IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 71 – 80, 2004.

[7] Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of Advances in Cryptology - CRYPTO 2001, LNCS 2139*, pages 41–62, 2001.

[8] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The sybil attack in sensor networks: Analysis & defenses. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pages 259–268, 2004.

[9] Bryan Parno, Adrian Perrig, and Virgil Gligor. Distributed detection of node replication attacks in sensor networks. In *Proceedings of The 2005 IEEE Symposim on Security and Privacy (S&P'05)*, pages 49–63, 2005.

[10] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: A geographic hash table for data-centric storage. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 78–87, 2002.

[11] Arvind Seshadri, Adrian Perrig, Leendert Van Doorn, and Pradeep Khosla. SWATT: SoftWare-based ATTestation for embedded devices. In *Proceedings of IEEE Symposium on Security and Privacy (S&P'04)*, pages 272–282, 2004.

[12] Chung Kei Wong, Mohamed Gouda, and Simon S. Lam. Secure group communications using key graphs. In *Proceedings of the ACM SIGCOMM '98*, pages 68–79, 1998.