Active Detection of Node Replication Attacks

Carlos Aguilar Melchor, Boussad Ait-Salem, Philippe Gaborit and karim Tamine

University of Limoges, 83 rue d'Isle
87000 LIMOGES, FRANCE

Summary
Sensor networks allow to deploy large self-organized and adaptable sets of sensors for many applications such as monitoring, detection, tracking etc. Unfortunately, the simplicity and low-cost of the sensors eases replication of nodes by attackers. Node replication attacks are the entry point to a large span of insidious attacks. Using replicas it is possible to capture, alter or suppress traffic and to disrupt protocols through misbehavior.
In 2005 Parno et al. proposed a passive protocol [1] for distributed detection of node replication attacks in sensor networks based on location claims. For an *n* node network, the detection protocol results in $O(\sqrt{n})$ message transmissions per node where the trivial approach would result in $O(n)$ messages per node. It is the first non-centralized protocol providing the emerging property of node replication detection and provides a performance leap from a communication point of view when compared to the trivial approach. On the other hand, each node needs to store $O(\sqrt{n})$ signed location claims which is an important limiting factor as sensor memory is quickly saturated.
In this paper we propose a new distributed protocol in which each node verifies at random a few other nodes in the network. Our protocol results in the same communication complexity than the protocol of Parno et al. but no storage is done on the nodes.

1. Introduction

Sensor nodes (also called motes) are cheap, resource-limited sensing devices which can communicate at short distances, and have a small amount of memory and computing power. Sensor networks are formed of a dense set of such nodes which can be randomly drop on an area and will self-organize without external intervention. Each node usually executes the same simple operations which result in an emerging property or service (detection, monitoring, tracking etc.) that is delivered by the sensor network as a whole. Simple and rapid deployment, scalability, and robustness are usual properties resulting from this approach.
In the 2005 IEEE Symposium on Security and Privacy, Parno et al. presented a paper [1] in which they remarked that, in order to remain cheap, sensor nodes could not be shielded against analysis and replication. This raises many security issues as undetected node replication allows strong insidious attacks. Indeed, data corruption or suppression can be done at a wide scale despite the natural robustness of sensor networks. Moreover, node isolation or revocation and network partition may result from localized outnumbering. More generally, disruption of the basic protocols is difficult to stop if a large percentage of the nodes are misbehaving and cannot be detected as replicas.
Various approaches to solve node replication exist ([2], [3]), but are based on centralized monitoring. Namely, all the nodes transfer their neighbors locations to a central entity which revokes replicated nodes based on cross-comparison of the data received. Of course, such a centralized approach goes against the emergent nature of sensor networks and creates a single point of failure. Thus, after noting the importance of replication detection, Parno et al. proposed two emergent protocols based on the distributed verification of the location claims.
These distributed schemes are based on passive discovery of the replicated nodes by witness nodes storing signed locations claims. This storage can be an issue as memory of sensor network nodes is usually very limited. We present in this paper a protocol in which each node actively tries to learn whether another node is replicated or not eliminating the memory saturation issue, while keeping the communication complexity.
This paper is organized as follows. In Section II we briefly describe the contribution of this paper. The setting and basic notions are described in Section III, and the centralized and trivial distributed protocols are presented in Section IV. In Section V we introduce the proposals of Parno et al., and in Section VI we describe our protocol. Section VI is devoted to the security and performance evaluation. Finally, we conclude in Section VII.

2. Contribution

Our contribution is two-fold. First we propose a new type of approach, active detection. This approach is an alternative to the one of Parno et al. and results in a family of new possible protocols. Second, we consider a particular protocol based on this approach which permits to gain in terms of memory over Parno et al.
More precisely, in our protocol each node tests actively

Manuscript received February 5, 2009
Manuscript revised February 20, 2009

whether some (randomly chosen) nodes are replicated or not. In order to do this it uses a few relays randomly placed on the network to get location claims of the tested nodes. We prove that if enough relays are chosen and a tested node is replicated, it is very unlikely that all the relays will communicate with the same replica. The tester will thus detect through these relays that different replicas exist and ban the replicated node using the conflicting location claims. On the other hand, in the protocols proposed by Parno et al. the nodes detect replication passively. Each node sends to a set of witnesses a location claim (the protocols differ by how these witnesses are chosen). If there is a replicated node, a witness will (passively) receive two conflicting locations claims and use them to ban the replicas.

The work of Parno et al. highlights that passive detection is not itself a protocol, but more a family of protocols. Depending on how witnesses are chosen, different protocols will be obtained, and the performance results will vary. The same can be said about the active discovery approach: depending on how relay nodes are selected protocols and performance will vary. In this paper we present a protocol, but other choices of the relays are possible, and will give very different results.

The second contribution of this paper is, as already noted, the memory usage improvement. The trivial distributed protocol, node to network broadcast, needs $O(n)$ communication per node and $O(1)$ memory on a n node network. The protocol of Parno et al. allows to reduce the communication cost to $O(\sqrt{n})$, but increases memory usage (during the protocol) to $O(\sqrt{n})$. In other words, the (communication, memory) cost passes from $(O(n), O(1))$ to $(O(\sqrt{n}), O(\sqrt{n}))$. With our protocol we obtain the same communication complexity without the memory usage increase, i.e. $(O(\sqrt{n}),\ O(1))$.

Besides the asymptotic improvement, this has an impact in practice. Indeed, to be more precise, in the protocol of Parno et al. each node needs to store in average $3\sqrt{n}$ signed location claims (with the parameters they propose). Supposing that the location can be encoded in just 20 bits (10 for each coordinate) and that the signature is 160 bits long, the amount of memory used in average is $540\sqrt{n}$ bits. Parno et al. use two motes for performance evaluation, the MICA 2 (with 4 Kbytes of RAM) and the new Telos B mote (with 10 Kbytes of RAM). They suppose TinyECC [4] is used for signature, which has a RAM footprint of 1 Kbyte. This leaves for the MICA 2 and the Telos motes 3 and 9 Kbytes respectively for the signed claims. Figure 1 shows the percentage of memory used by the claims depending on the number of nodes in the network.

The MICA 2 cannot handle this protocol for networks of more than 2000 nodes, and the Telos mote is limited to 18000 node networks. In sensor networks it is usual to suppose that the number of nodes goes up to 100000 nodes, so the memory usage is a practical issue for this protocol.

The second contribution of our protocol is thus not only interesting from an asymptotic point of view but also for practical networks, enabling node replication detection for a large span of networks in which the usage of the protocol of Parno et al. is not possible for commonly used motes.

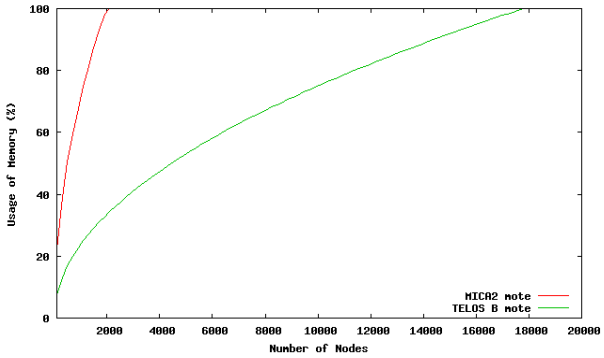

Fig. 1 Memory usage with Parno et al.'s protocol.

3. Bases

To simplify the comparison with the work of Parno et al. we place ourselves in the same context.

3.1 Goal

We aim to detect the presence of replicated nodes in a sensor network with a distributed protocol. We also aim to be able to revoke replicated nodes in such a way that all the other nodes will cease to communicate with them. Both the detection and the revocation should be obtained at the end of the protocol. We want the protocol to be successful with probability exponentially close to 1 in a security parameter K such that the number of messages each node has to send is at most $K\sqrt{n}$. Finally, we want the protocol to require a small fixed amount of memory.

3.2 Nodes and network

We consider that nodes are fixed[1] and that a scheme providing location information to the nodes is available (see [5], [6], [7]). . Each node is supposed to have a

1 None of the motes available nowadays to build up sensor networks are mobile, which justifies this choice.

unique *ID*. We also suppose the existence of an identity-based public key cryptosystem such that from a node's *ID* α every node can obtain through a known function f the associated public key K_α. Each node has the private key associated to his public key (which is included during node production for example).
Public key cryptography is often considered too expensive for sensor networks. However we consider its usage for two reasons. First, if an attacker able to replicate physically nodes can also create new identities it will not be possible to detect the replication. Thus, we need to provide a unique property to each node that can't be simulated by an attacker. Identity-based cryptography is the cheapest technique available. Second, elliptic curve implementations of public key cryptosystems have been proposed for TinyOS [4], [8]. In particular TinyECC allows to sign and verify signatures in less than 100 milliseconds for current motes which is reasonable given the small amount of signatures and verifications needed per node in the presented protocols.

3.3 Adversary model

The adversary is able to clone and modify the nodes in any way he wants but he is unable to create new identities which are linked to the nodes public/private keypairs.
The replicated nodes of the adversary can communicate and collaborate. However, the adversary is supposed to be able to capture a limited number of sensor nodes. Indeed, if he is able to capture most of them he will probably be able to thwart any protocol. Similarly, we suppose the replicated nodes try to remain inconspicuous and that conspicuous attacks will trigger a sweeping protocol (Parno et al. propose SWATT [9]) or human intervention. In particular, we suppose that the adversary cannot intercept or disrupt a significant share of the communications. In other words, the nodes can refuse to play the game, may try to cheat, but cannot prevent the protocol from being executed by the legitimate nodes.

3.4 Notation

Again, we choose the same symbol and notation that Parno et al.

n	Number of nodes in the network
d	Average degree of each node
p	Probability a neighbor will replicate location information
g	Number of witnesses selected by each neighbor
l_α	Location node α claims to occupy
$H(M)$	Hash of M
K_α	α's public key
K_α^{-1}	α's private key
$\{M\}_{K_\alpha^{-1}}$	α's signature on M
S	Set of all possible node IDs

We suppose, to simplify the performance analysis, that the network diameter is in $O(\sqrt{n})$. If this is not the case the performance analysis should be adapted to the given network geometry. In such a model two arbitrary nodes are separated by roughly $\sqrt{n}/2$ hops and we will suppose that sending a message from one node to another generates thus $O(\sqrt{n})$ messages.

4. Trivial Approaches

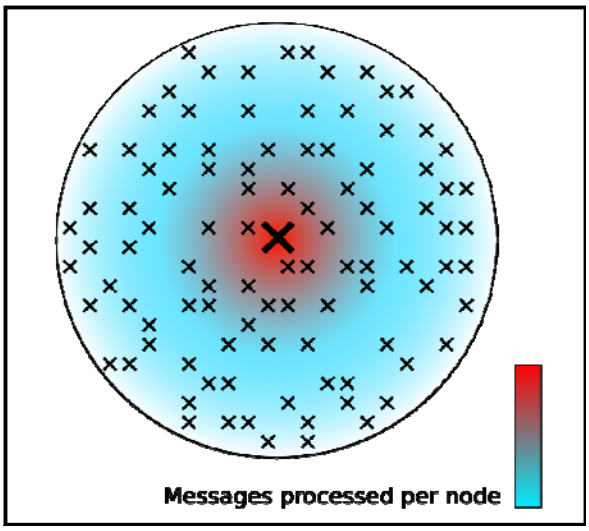

Fig. 2 Concentration of messages in centralized approach

4.1 Centralized approach

In a centralized approach each node will send to a base station a list of its neighbors together with a location. The base verifies that no node is in two (non-adjacent) locations at the same time. The central station receives and processes $O(n)$ messages, but this is not generally an issue as it does not have the memory, communication

and processing constraints of the nodes. This generates $O(n\sqrt{n})$ messages on the network and the node memory usage is null.
Of course, such an approach may lead to node framing as a node could falsely claim the presence of another node. This can easily be solved by a two step protocol. First, each node locally broadcasts a signed location claim. Second, each node gathers all the signed location claims its *d* neighbors have locally broadcasted and sends them to the base station. With such an approach it is not possible to frame a node as its signed location claim will be needed. On the other hand, if a node does not broadcast a signed location claim its neighbors can refuse to communicate with him.
The two main drawbacks of this approach is the existence of a single point of failure and the need to have such a permanently present base station. Both drawbacks disappear in distributed approaches. A third non-negligible drawback is the large unbalance of message processing. Indeed as all the traffic goes to the same destination, nodes close to the base station will be quickly overwhelmed receiving and sending $O(n)$ messages. On the other hand the nodes on the periphery will just send $O(1)$ messages.

4.2 Distributed approaches

1) Node-to-network broadcast: In this approach, each node broadcasts to the whole network a signed location claim and stores the location claims of its *d* neighbors. If it receives a signed location claim conflicting with one of its neighbors it broadcasts to the whole network a revocation proof containing the conflicting claims.
Supposing that there is an efficient duplicate suppression algorithm each broadcast requires $O(n)$ messages, and thus the global communication cost is $O(n^2)$. Each node must store the claims of its neighbors and therefore the memory usage is $O(d)$.
It is not possible to frame a node as the replication proof can only be obtained if two replicas at different locations broadcast a signed claim, and it is easy to force every node to broadcast their claim by refusing local communication with neighbors who haven't done it.

2) Deterministic multicast: In this approach there is a public deterministic function *F* that for each node α outputs a set of witness nodes *F* (α). Each node does a local broadcast of its signed location claim and each neighbor forwards it with probability *p* to a random subset of the node's witnesses. The probability of forwarding the claim, the size of the subset, and the size of the sets returned by *F* are parameters of the protocol. Parno et al. propose to use the coupon collector's problem [10] to choose the parameters such that if two replicas send a location claim at least one witness will receive both and therefore be able to broadcast a revocation proof.
The main issue with this approach is that as *F* is known an adversary needs just to capture a node α and its witnesses *F* (α) to be able to replicate α as many times as desired without being caught. It is possible to choose *F* such that witness set size *g* is large, but the communication cost is on $O(g \times \ln(g) \times n\sqrt{n})$ (when using the coupon collector's problem) which limits the value of *g*.

5. Parno et al. Protocols

5.1 Randomized multicast protocol

In this protocol the function *F* of the deterministic multicast protocol is replaced by a random choice. Each neighbor forwards a node's claim with probability *p* to a set of *g* random witnesses. For $p \times d \times g \cong \sqrt{n}$ each node will have $O(\sqrt{n})$ witnesses and the birthday paradox will ensure that if two replicas send different signed location claims there will be with high probability a witness that will receive both.
The main issue with such a protocol is that sending a message to a node costs $O(\sqrt{n})$ messages, and thus contacting $\sqrt{n}$ witnesses per node costs $O(n)$. The traffic generated is thus $O(n^2)$, the same than with the trivial node to network broadcast. However this protocol shows the path to obtain a communication efficient protocol, the line-selected multicast.

5.2 Line-selected multicast protocol

The idea of the line-selected multicast is to choose witnesses in such a way that there are too many possibilities for an adversary to control them, but in such a way that contacting these witnesses is not costly. If moreover, one wants the witness sets of two replicas to intersect with high probability, the choice of a technique becomes non-trivial.
Parno et al. propose to set the witnesses as the intermediaries routing the messages to a random node. This has two nice advantages, first it creates an almost straight line in the network which has a good probability to be intersected by other random lines. The second advantage is that $O(\sqrt{n})$ nodes witness the location

claim, but the communication cost is not $O\left(\sqrt{n}\times\sqrt{n}\right)$ as for random witnesses.

Each node does a local broadcast of its signed location claim. Then, each of its *d* neighbors forwards the claim with probability *p*. A neighbor deciding to forward the claim will choose a random node and send him the claim. Each of the intermediary nodes routing the claim will store it creating thus a line of witnesses. Parno et al. provide some theoretic formulas showing that in ideal cases the number of witness lines needed to be sure that two replicas will have an intersection is small, and assert that heuristic results show that taking five lines per node lead to a probability of intersection of *95%*.

Setting $p = K/d$ ensures that nodes will have roughly *K* lines of witnesses. For such a value the total communication cost will be $O\left(K\times n\sqrt{n}\right)$ with a probability that two replicas will produce intersecting sets of witnesses exponentially close to 1 as *K* grows. Each node will have to store $O\left(K\times\sqrt{n}\right)$ location claims.

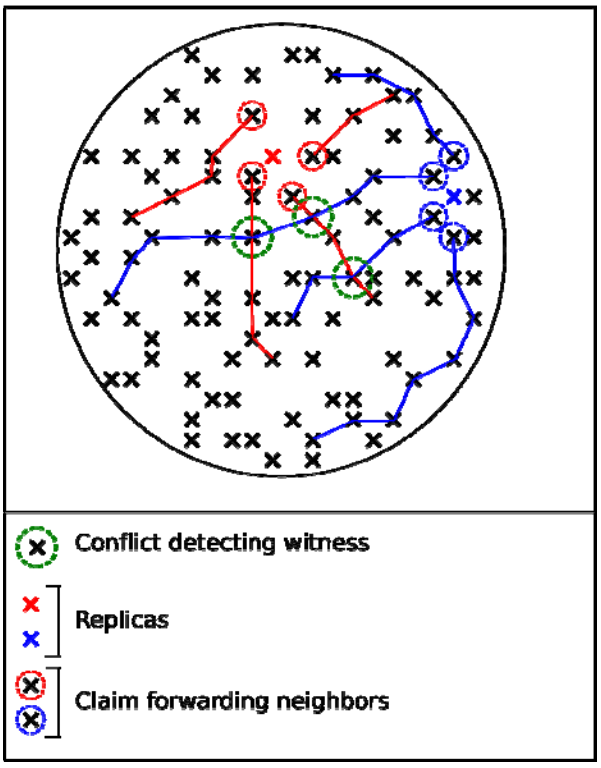

Fig. 3 Intersection of line selected witness sets

6. Active Discovery Protocol

In our protocol, we do not build a distributed database of location claims that will contain local conflicting claims when replicas exist. This is the reason why we do not use up node memory. The idea is that each node will actively test if k_1 other random nodes are replicated or not. We call them the scrutinized nodes. In order to test whether a scrutinized node α is replicated or not k_2 nodes are randomly chosen in the network and asked to forward to α a request for a signed location claim. If two replicas exist, each will probably receive a request, and if both answer two conflicting claims will be obtained by the querier. We will prove that the probability for a given replicated node of being detected is exponentially close to one in $min(k_1, k_2)$.

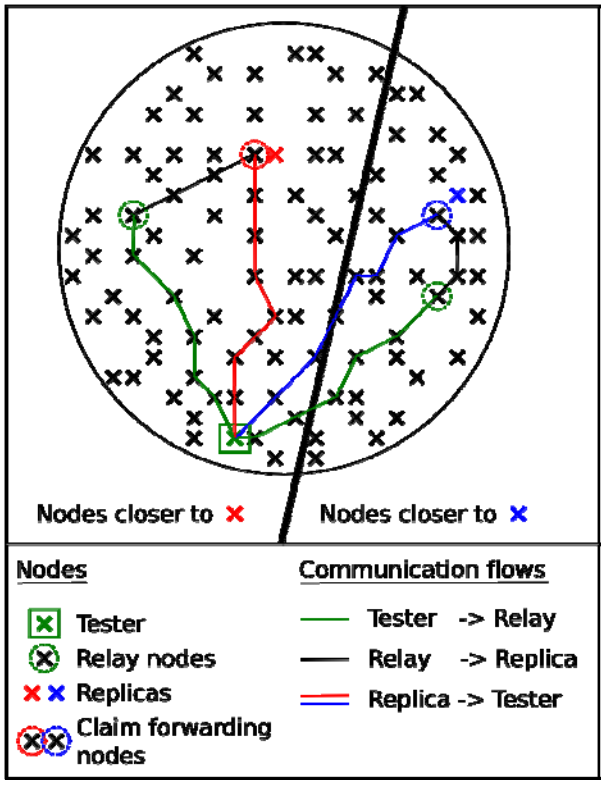

Fig. 4 Location request forwarding.

6.1 Description

The scheme we propose is fully distributed. Each node α executes the following steps.

Active replication discovery protocol

1. Choose randomly k_1 nodes $\alpha_1, \ldots \alpha_{k_1} \in S$ for scrutiny and k_2 relay nodes $\beta_1, \ldots \beta_{k_2} \in S$.

2. For *i* from *1* to k_1 create a location claim request $\{LocRequest, \alpha_i, \alpha\}$

3. For *i* from *1* to k_2 send the location claim requests to β_i

4. Wait until receiving k_2 location claims from each scrutinized node or until the maximum time for the discovery phase is reached.

5. For *i* from *1* to k_1 do

a. If all the location claims from α_i are coherent do nothing

b. Else choose a subset σ of incoherent claims and broadcast to the whole network $\{Revoke, \alpha_i, \sigma\}$ in order to revoke all the replicas of α_i

Whenever a node γ receives a location claim request $\{LocRequest, \alpha_i, \alpha\}$ three situations are possible. First, if γ is the node α_i he creates a signed location claim $\{Position\ P, \alpha_i\}_{K_{\alpha i}^{-1}}$ and sends it to the neighbor who has delivered the location claim request. Second, if γ is not α_i nor one of its direct neighbors it forwards the location claim request to the next node on the route to reach α_i. Third, if γ is a direct neighbor of α_i he follows the next protocol.

Location claim retrieval protocol

1. Ask α_i for a signed location claim

2. If α_i answers verify that the location claim is valid

3. If α_i fails to provide a valid claim

a. Refuse to relay messages to/from α_i

b. Create an isolation request of α_i, $\{Isolate, \gamma, \{LocRequest, \alpha_i, \alpha\}\}$

c. Do a local broadcast of the isolation request

d. Wait for a signed location claim of α_i

e. If a valid location claim is obtained restore the communications with α_i and proceed with the protocol

6. Send the signed location claim to α.

A location claim is said to be valid if the announced location is plausible, the second field is α_i, and the signature is valid. The node α_i answers to the location claim request with a local broadcast of the signed claim. The neighbors record this fact, and ignore further location claim requests from α during this execution of the protocol limiting thus the traffic generated.

When a node γ` receives an isolation request $\{Isolate, \gamma, \{LocRequest, \alpha_i, \alpha\}\}$ he first verifies whether or not he is a neighbor of $\alpha_i$. If he is not, the request is ignored. Else, $\gamma$` does a location claim retrieval protocol except that in the last step, instead of sending the signed claim back to α it sends it to γ.

The isolation procedure is recursive. Indeed if γ` is unable to retrieve a signed location claim he will broadcast a new isolation request that will reach new neighbors. If one of these neighbors gets a location claim he will forward it to $\gamma$` who will forward it to γ. The density of sensor networks ensure that the broadcasts of the isolation requests will, after a few iterations, reach all the neighbors of α_i. If he refuses to collaborate he will therefore become completely isolated.

6.2 Security

In this section we prove that a replicated node has a probability exponentially small in $min(k_1, k_2)$ of avoiding detection. In order to do this we prove two intermediate properties that link k_1 and k_2 to the probabilities for a given node of being scrutinized, and for a replicated node under scrutiny of being discovered. We first present a simple lemma.

Lemma 1: For any $k_1 > 0$, $U_n = (1 - k_1/n)^n$ is an increasing sequence that converges to the limit e^{-k_1}.

Proof: Developing the general term with the binomial formula proves that $U_{n+1} > U_n$ and thus the sequence is increasing. As $(1 - k_1/n)^n = e^{n\ln(1 - k_1/n)}$, using the taylor series of $\ln(1 - x)$ we obtain $U_n = e^{n\left(-k_1/n + O\left((k_1/n)^2\right)\right)} = e^{-k_1 + O\left(k_1^2/n\right)}$ and thus the limit of U_n is e^{-k_1}.

Proposition 1: The probability for a given node of not being scrutinized is exponentially small in k_1.

Proof: A node chooses k_1 nodes among $n - 1$ (it never chooses himself) for scrutiny. As scrutinized nodes are chosen randomly, a given node α will have a probability $k_1/(n - 1)$ of being tested by another given node β. As each node operates independently the

probability that α will not be tested by any node is $U_{n-1} = (1 - k_1/(n-1))^{n-1}$. As U_n is an increasing sequence the probability for a given node of not being scrutinized is smaller than e^{-k1} which proves the proposition.

This value decreases fast even for small values of k_1. For example, for $k_1 = 3$ we have a probability smaller than *5%* that a given node will not be selected for scrutiny and for $k_1 = 5$ this probability drops under *1%*.

Lemma 2: The probability that all the relay nodes address the location claim request to the same replica of a replicated node is exponentially small in k_2.

Proof: When a node α is replicated, we can associate to each replica α_i the set of nodes S_i that are closer (from a routing point of view) to α_i than to any other replica. The sets S_i form a partition of the network. For example if two replicas of α exist (α_1 and α_2), the network will be partitioned in two: the nodes which route the messages for α to α_1 and the nodes which route the messages to α_2. Note n_p the number of nodes in the largest partition. The probability for a node chosen randomly to be in this partition is $n_p / n < 1$ and thus the probability for k_2 random nodes of being in the same partition is smaller than $(n_p / n)^{k_2}$. The probability that all the location claim requests are forwarded to the same replica is thus $(n_p / n)^{k_2}$ which is exponentially small in k_2.

Proposition 2: Two replicas of a scrutinized node have a probability exponentially small in k_2 of avoiding detection.

Proof: Suppose α is a replicated node under scrutiny of a given node β. If β chooses k_2 random relay nodes that route the location claim requests to the same replica, the protocol will have a normal execution and yet the replication will remain undetected. On the other hand, if there are two requests that reach two different replicas, β will obtain a revocation proof or one of the replicas will be locally isolated. Thus to avoid detection, all the relay nodes must address their requests to the same replica which following Lemma 2 happens with probability exponentially small in k_2.

Node replication will therefore be detected (globally through a revocation proof or locally if a node refuses to collaborate) with a probability exponentially close to one in k_2. If a node has a single replica and each partition contains *n/2* nodes, the probability that the replicated node will resist scrutiny is $2 \times 1/2^{k_2}$ (all the requests on one partition or in the other). For $k_2 = 5$ this probability is of approximately *6%* and it drops to *1%* for $k_2 = 7$.

A replicated node will remain undetected if he is not scrutinized, or if he is scrutinized but it avoids detection. Both cases have an exponentially small probability either in k_1 or in k_2. The probability a replicated node remains undetected in our protocol is therefore exponentially small in *min(*k_1 , k_2 *)*.

6.3 Performance

1) Asymptotic results:

In the active recovery protocol, each querying node generates three sort of transmissions; k_2 from the querying node to the relaying nodes; $k_1 \times k_2$ from the relaying nodes to the scrutinized nodes; and k_1 from the scrutinized nodes to the querying node. The expected communication complexity is therefore of $K\sqrt{n}$ messages per node with $K = 1/2(k_2 + k_1 k_2 + k_1)$.

The local claim retrieval protocol together with the isolation procedure will at most generate *O(d)* local messages and therefore the traffic generated is negligible when compared to the one of the discovery protocol.

Table 1: Asymptotic Performance

Protocol	Communication	Memory
Node to network broadcast	$O(n)$	$O(d)$
Deterministic multicast	$\frac{O(g \ln g \times \sqrt{n})}{d}$	$O(g)$
Randomized multicast	$O(n)$	$O(\sqrt{n})$
Lineselected multicast	$O(\sqrt{n})$	$O(\sqrt{n})$
Active discovery	$O(\sqrt{n})$	$O(1)$

From a computational point of view a node just needs to sign a location claim once and use it each time it is requested. On the other hand, the expected value for signed location claim verifications is $2 k_1 k_2$ (once by

each direct neighbor and once by the querier for each query).

Each node must remember the identities of the nodes it tests which needs $O(k_1)$ memory. The expected number of signed claims to process for a given node is $k_1 k_2$. In a worst case scenario a node might end with $O(k_1 k_2)$ signed claims at the same time which is highly unlikely but possible, and would be the most memory consuming situation.

Asymptotic performance of all the protocols is presented in Table 1. Our protocol is the only one with constant memory use and communication cost per node in $O(\sqrt{n})$ without security issues due to a deterministic choice of witnesses.

The second protocol with communication cost in $O(\sqrt{n})$ and random witnesses is the line-selected multicast protocol proposed by Parno et al. On the other hand this protocol has a memory cost in $O(\sqrt{n})$ which is an important drawback.

2) Simulations:

In order to consider practical values and not only asymptotic behavior we have run a set of simulations using JiST the Java in Simulation Time simulator [11]. The comparison to the protocol of Parno et al. being one of our objectives we have followed the same simulation procedure as them. We have tested the detection rates for the same topologies as Parno et al. defined (five hundred tests per topology), for one thousand node networks. We have also tested the average number of packets sent and received per node for our protocol for an *n* node network with $n \in [1000; 10000]$. The nodes are placed on a square network following an uniform distribution and the square size is chosen such that the average number of neighbors *d* is *40*.

In order to reach the same detections rate as Parno et al. We must set $k_1 = 2$ and $k_2 = 3$ with an average detection rate around *75%* (see Figure 5).

Increasing the number of nodes queried to $k_1 = 3$ leads to detection rates around *85%* (see Figure 6). In both sets of tests only one node is replicated and there is just two replicas of the node. If the number of replicas increases the detection rate becomes exponentially close to one hundred percent (the same being true for the protocol of Parno et al.).

The average number of packets sent and received per node in the protocol of Parno et al. is $3\sqrt{n}$. In our protocol, with parameters $k_1 = 2$ and $k_2 = 3$ the average number of packets is $5.5\sqrt{n}$. In Figure 7, our protocol is compared to the one of Parno et al. and to node to randomized multicast. The simulation confirms the theoretical results about communication complexity, our protocol has a $O(\sqrt{n})$ complexity with a communication cost similar to the one of Parno et al.

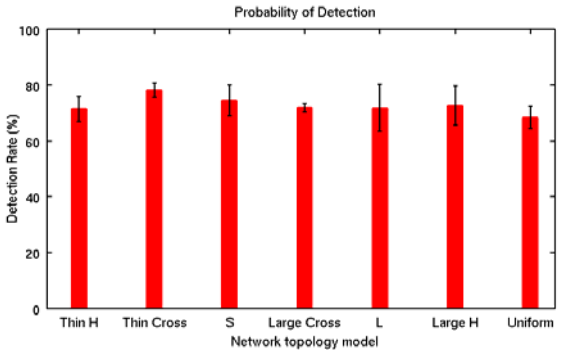

Fig. 5 Detection rates for $k_1 = 2$ and $k_2 = 3$.

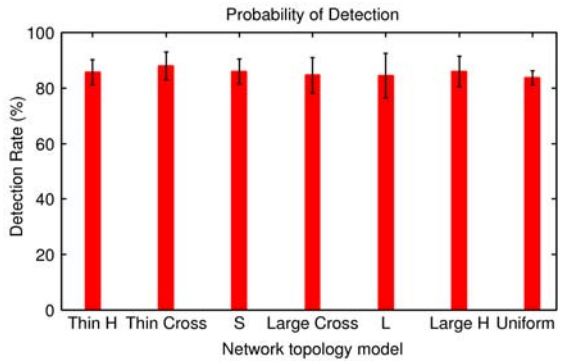

Fig. 6 Detection rates for $k_1 = 3$ and $k_2 = 3$.

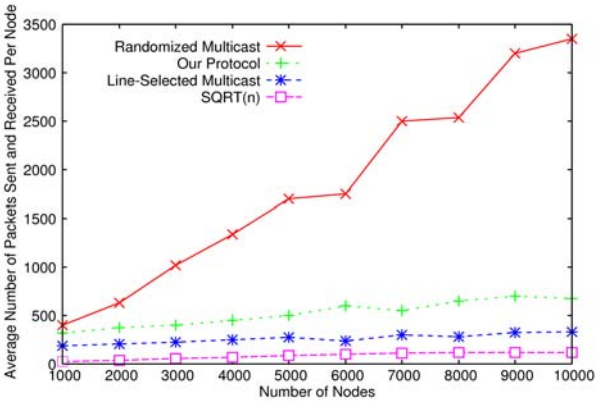

Fig. 7 Communication overhead.

7. Conclusion

Building a replication detection protocol has some constraints. First, node locations must be learnt by other nodes (witnesses). Second, at least one witness must get two conflicting location claims if a node is replicated. Third, witnesses must be randomly associated to the scrutinized nodes to avoid security issues. Fourth, the less witnesses are needed, the best performance we obtain.

The main advantage of an active approach is that the witnesses scrutinize a set of nodes whose size is independent of the number of nodes on the network. On the other hand, the passive approach is based on the intersection of randoms sets of witnesses. The birthday paradox ensures that $O\left(\sqrt{n}\right)$ is enough for such sets and a clever distribution of these witnesses such as the line-selected multicast allows to obtain reasonable communication costs for the distribution of the claims. On the other hand all the proposed protocols fail to solve the other drawback of having so many witnesses: if each node needs $O\left(\sqrt{n}\right)$ witnesses, the total number of stored claims is $O\left(n\sqrt{n}\right)$ which leads to the $O\left(\sqrt{n}\right)$ memory usage per node. The active approach needing a constant number of scrutinized nodes per node, the total number of stored claims is in $O(n)$ and thus the memory usage per node is $O(1)$. As the number of relays per node is also a constant, the communication is automatically in $O\left(\sqrt{n}\right)$ without even needing to chose a clever distribution of the relays.

We hope this paper will motivate research in other active techniques able to lower even more the communication costs and increase detection rates.

References

[1] B. Parno, A. Perrig, and V. D. Gligor, "Distributed detection of node replication attacks in sensor networks," in IEEE Symposium on Security and Privacy. IEEE Computer Society, 2005, pp. 49–63. [Online].Available:http://doi.ieeecomputersociety.org/10.1109/SP.2005.8

[2] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proceedings of the 2003 Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, May 11–14 2003, pp. 197–215.

[3] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in Proceedings of the third international symposium on Information processing in sensor networks (IPSN-04). New York: ACM Press, Apr. 26–27 2004, pp. 259–268.

[4] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," 2004. [Online]. Available: citeseer.ist.psu.edu/malan04publickey.html

[5] J. Newsome and D. Song, "GEM: graph embedding for routing and data-centric storage in sensor networks without geographic information," in Proceedings of the first international conference on Embedded networked sensor systems (SenSys-03). New York: ACM Press, Nov. 5–7 2003, pp. 76–88.

[6] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex optimization methods for sensor node position estimation," in INFOCOM, 2001, pp. 1655–1663. [Online]. Available: http://www.ieee-infocom.org/2001/paper/646.pdf

[7] H.-C. Chu and R.-H. Jan, "A GPS-less, outdoor, self-positioning method for wireless sensor networks," Ad Hoc Networks, vol. 5, no. 5, pp. 547–557, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2006.03.004

[8] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," Department of Computer Science, North Carolina State University, Tech. Rep. TR-2007-36, Nov. 02 2007.

[9] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla, "SWATT: SoftWare-based ATTestation for embedded devices," in Proceedings of the IEEE Symposium on Research in Security and Privacy. Oakland, CA: IEEE Computer Society Press, May 2004.

[10] T. H. Corman, C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms. MIT Press, 1990.

[11] R. Barr, Z. J. Haas, and R. van Renesse, "JiST: an efficient approach to simulation using virtual machines," Softw, Pract. Exper, vol. 35, no. 6, pp. 539–576, 2005. [Online]. Available: http://dx.doi.org/10.1002/spe.647