Computer Networks 108 (2016) 210–222

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Imposter detection for replication attacks in mobile sensor networks

Tassos Dimitriou[a], Ebrahim A. Alrashed[b,*], Mehmet Hakan Karaata[b], Ali Hamdan[b]

[a] Computer Technology Institute, Greece
[b] Department of Computer Engineering, Kuwait University, Kuwait

ARTICLE INFO

Article history:
Received 17 December 2015
Revised 13 July 2016
Accepted 22 August 2016
Available online 30 August 2016

Keywords:
Mobile sensor networks
Imposter detection
Node replication attack
Wireless network security
Node revocation
Soundness & completeness

ABSTRACT

In a node replication attack, an adversary creates replicas of captured sensor nodes in an attempt to control information that is reaching the base station or, more generally, compromise the functionality of the network. In this work, we develop fully distributed and completely decentralized schemes to detect and evict multiple imposters in mobile wireless sensor networks (MWSNs). The proposed schemes not only quarantines these malicious nodes but also withstand collusion against collaborating imposters trying to blacklist legitimate nodes of the network. Hence the completeness and soundness of the protocols is guaranteed. Our protocols are coupled with extensive mathematical and experimental results, proving the viability of our proposals, thus making them fit for realistic mobile sensor network deployments.

1. Introduction

A *Wireless Sensor Network* (WSN) is a wireless network of small sensors deployed in a specific area to sense various aspects of the environment. A *Mobile Wireless Sensor Network* (MWSN) is a special type of WSN in which sensors are mobile. MWSNs convey the sensed data to base stations or sink nodes, which can be either static or mobile, thus trying to cope with rapid topology changes that make sensing problematic in ordinary sensor networks. As a result, they extend the number of applications for which static (WSNs) are used [1]. Sensors can be attached to people for health and physiological monitoring, to animals in order to track their movements and their feeding habits, or to unmanned aerial vehicles (UAVs) for surveillance, environmental mapping and control [2,3].

In a typical WSN, where the sensor nodes are stationary, the sink or other nodes can ascertain the authenticity of a sensor node by tying its identity to its *claimed* geographic location [4]; through the help of witness nodes, location claims coming from conflicting areas in the network indicate the existence of a replication attack.

In a MWSN, however, the constant movement of nodes makes location-based detection a nearly impossible task. As a result, an adversary can assume the identity of a legitimate node and use it to communicate with the rest of the network. As sensor nodes are not tamper-resistant devices [5], the adversary can create *replicas* of nodes after compromising a node and replicating its cryptographic or other material. We refer to such replicas as *imposters* if they use the identity of existing sensor nodes to communicate with the sink or other nodes of the network.

Since the credentials of replicated nodes do not differ from those of legitimate ones, there is no easy way to distinguish between the two, thus making imposter detection a very difficult process. This type of attack, which is known as *node replication attack* in the literature, has important repercussions in wireless sensor networks security: by assuming a false identity, an imposter can send misleading information, replay old packets which could bias aggregation results or enable other types of attacks in the network, like selective forwarding, sinkhole attacks, etc. [6–8].

Contributions. In this work, we address the problem of node replication attacks by proposing a number of lightweight, *decentralized* protocols to detect imposters in MWSNs. Contrary to prior work that focuses only on imposters that can replicate only a *single* node ID, our schemes work even in those cases where imposters have assumed the identities of *different* nodes. This case is more challenging as it poses another problem: imposters can *frame* legitimate nodes, thus resulting in their dismissal from their network.

In this work, we show not only how to detect these powerful imposters but also maintain the number of false-positives (evictions of legitimate nodes) to a bare minimum. Eventually, when a sensor node is identified to be an imposter, it is prevented from communicating with other nodes in the network by means of an effective quarantining mechanism. Hence our protocols are both sound and complete. Finally, through extensive simulations, we

* Corresponding author.
E-mail address: dr_ebrahim@mac.com (E.A. Alrashed).

http://dx.doi.org/10.1016/j.comnet.2016.08.019
1389-1286/

demonstrate the practicality and viability of our approach in detecting and mitigating the node replication attack.

Organization. The rest of the paper is organized as follows. In Section 2, we review related work on imposter identification in wireless sensor networks. In Section 3, the threat model and assumptions are discussed, while in Section 4, a number of schemes are presented and analyzed. Experimental results are discussed and evaluated in Section 5. Finally, Section 6 concludes the paper.

2. Related work

In this section we review prior work on imposter identification which also comes under the name of *node replication* detection. Initial work [9–11] focused on the study of radio-based detection which attempts to authenticate nodes, and eventually detect imposters, based on signal strength or other physical characteristic of radio communication.

Network-based detection typically relies on the use of a *claimer-reporter-witness* framework, originally proposed by Parno et al. in [4]. These techniques, which mostly work for *stationary* networks, store information about the location of a sensor node in one or more witnesses in the network which can then detect and report replicas once they receive more than one location claim from nodes interacting with a particular sensor node. A more detailed review of works addressing the problem in stationary sensor networks can be found in [12].

For mobile sensor networks, one line of research involves the study of properties possessed by the network as a whole in order to trigger the existence of imposters [13,14]. In [13], a centralized scheme is proposed where a base station is used to calculate the speed of nodes based on location information received by neighbors of that node. If the speed exceeds a predefined threshold, an alarm is raised and the replica is detected. In a similar manner, the basic idea in the work of [14] is to differentiate between the time a node u encounters another node v when there are no replicas in the network (during initial deployment) as opposed to the case when replicas exist. The authors come up with a scheme based on the difference of the distributions of these two cases, hence replica identification is possible with certain probability. These approaches, however, rely on the existence of an all-powerful base station that maintains a complete picture of the network, thus requiring heavy localization and synchronization primitives by the nodes.

A different line of research involves the use of *tokens*, to authenticate the genuineness of a mobile node [15–17]. Once two sensor nodes encounter each other, they exchange random, unpredictable numbers. If the two nodes meet again, both of them request the other for the random number they exchanged at earlier time. If the other cannot reply or replies with a wrong number, the node is treated as an imposter and an alarm is raised. In this work we build upon this technique as it uses lighter cryptography and leads to simpler protocols. Our work, however, differs from these past results in three important aspects.

- First, our scheme can effectively neutralize *multiple* imposters that are copying *different* legitimate IDs. In contrast, past works ([13–17]) only consider imposters that are copies of a *single* node which makes detection easier; once the replicated ID is found, all imposters can be evicted from the network.
- Second, we develop protocols that are completely *decentralized* and nodes themselves, without the need of a powerful base station ([13–16]) or mobile sinks [17], succeed in quarantining these imposters.
- Finally, as in this more challenging case imposters can collaborate to blacklist legitimate nodes, we show how to avoid false positives by coming up with appropriate mitigation strategies.

3. Threat model and assumptions

We consider a mobile wireless sensor network (MWSN) consisting of N mobile sensor nodes deployed in a certain area of interest. Sensor nodes route their sensed data to a stationary base station or to a mobile sink that acts as a gateway to some external network using appropriate routing protocols ([25–27]). We assume all network nodes have limited resources and they are similar in terms of energy, memory and computational capabilities. In particular, sensor nodes have limited wireless communication radius and only the base station can broadcast messages to all nodes, if necessary. Thus, typically, nodes have a small number of neighbors which can utilize in forwarding data or exchange tokens that can be used in detecting imposters. They also move randomly within the specified coverage area but not necessarily with the same speed. As a result, the time and the location of node encounters, as well as the IDs of the meeting nodes are generally unpredictable.

We define an *imposter* to be a malicious node which uses the identity of a legitimate node to communicate with other nodes in the network. In our model, the imposter has obtained the cryptographic credentials of a genuine node u after compromising that node. It then uses these keys to communicate with the sink or other nodes, using u's identity and claiming to be node u. Messages received by either u or its imposter are indistinguishable, so it is not possible to differentiate between the two by virtue of messages sent. The only way that the presence of an imposter can be detected is if a third node encounters both u and its replica, one after the other, and one of them replies with the wrong nonce.

We assume the base station is well protected, hence the adversary cannot generate new IDs by obtaining the corresponding base station credentials. Following [4,16], this is possible by assuming the existence of an ID-based cryptography scheme. Thus a node u is deployed with a private key K_u^{-1} and any other node can derive u's public key K_u by applying an appropriate function F to u's ID, i.e. $K_u = F(u)$. Such dynamic generation of public keys is a more preferable solution over a traditional public key infrastructure (PKI) system which would require every node to prove the validity of its public key by transmitting an appropriate certificate signed by the base station; the other alternative which requires every node to be preloaded with *all* nodes'public keys is clearly an impractical task for large scale sensor networks.

While key management schemes in WSNs are mainly based on symmetric cryptography, recent works [18–21] have demonstrated the feasibility of public key cryptography on resource-constrained sensor nodes. TinyPK [18] utilizes the RSA cryptosystem to provide authentication and key exchange between an external party and a sensor network. The use of Elliptic Curve Cryptography (ECC) [19] constitutes a much better alternative to traditional public key (PK) cryptography algorithms as it is possible to generate short 160-bit keys in resource-constrained devices. Identity-based solutions based on pairings have also been implemented [20,21] for sensor nodes based on 8-bit microprocessors (e.g., MICA 2 and MICAz motes or the Tmote Sky sensors), showing that pairing-based cryptography is indeed a practical alternative for sensor networks.

In the protocols of the next section, we follow the ID-based approach to *authentication* that can be achieved by tying the identity of a node to its public key so that any other node can verify the authenticity of a signed message by deriving the public key of the node from its unique ID. Since the only requirement in our protocols is the ability to generate and verify signatures, Shamir's original Identity-based signature scheme [22] can also be used as we don't need the full set of capabilities provided by pairings. This approach can lead to even lighter implementations when combined with ECC as discussed above. In Section 4.1.1, however, we also suggest a symmetric cryptography alternative to signing that requires less computation but more communication overhead.

Table 1
Summary of terms used.

Term used	Definition
Imposter	A malicious entity that uses the identity of a legitimate sensor node to communicate with other nodes in the network.
Nonce value	A random number used once; a temporary value that is exchanged between nodes and is used in imposter detection.
Nonce List	A list maintained individually by each node which contains nonce values expected from other nodes and nonce values to be sent to other nodes for successful authentication.
Direct Detection	The process through which a sensor node identifies an imposter ID through direct communication with the imposter.
Referred Detection	The process through which a sensor node identifies an imposter ID based on information (claims) received from other nodes in the network.
Quarantining	A process where all the nodes in the network stop communicating with the quarantined node.
Quarantine List	A list maintained individually by each node which contains IDs believed to be used by imposters.
Claim	A claim has the form $\langle detector_{id}, imposter_{id} \rangle$. It bears the signature of the detector node and is used to convey information to other nodes about potential imposters.
Claims List	A list which contains claims from other nodes against certain node IDs that are believed to be used by imposters.
False-positive	Signifies an error in detection by which a legitimate node is accused as an imposter. Typically, imposters can make use of false claims to blacklist genuine nodes.
Detection Time	Time required for all legitimate nodes to detect all imposters in the WSN.

The use of ID-based cryptography also explains why all replicas of a node bear the same network ID and credentials with the original node; as an adversary cannot generate a new ID without guessing the appropriate keys used by the base station (an act which is considered infeasible), the adversary must capture and clone only legitimate nodes. Once an attacker compromises such a node, it can then replicate it as many times as she likes using the node's exposed credentials: ID, software and cryptographic keys. The inserted replicas can then be used by the adversary to orchestrate other attacks, targeting upper-layer applications. This makes imposter detection a very important problem as it can be used to cripple the MWSN with very little effort and cost.

The number of imposters in the network is denoted by M. As we explained in the introduction, we deviate from prior work in the area which assumes that all imposters bear the same node ID (i.e. they are replicas of a single node). Thus, in our model, an attacker can compromise and clone different nodes IDs. When all legitimate nodes in the network stop communicating with a compromised node ID, the associated node (or its imposter) is described as *quarantined*. However, imposters can use the identity of different legitimate nodes and they may *collaborate* with each other to quarantine (evict) other nodes in the network. Hence an important aspect of our work is to keep the number of legitimate, blacklisted nodes (*false-positives*) as small as possible.

A summary of the terminology we will be using throughout the paper is shown in Table 1.

4. Imposter detection framework

In this section, we describe the proposed imposter detection schemes. Our schemes use *nonces* (unpredictable random numbers) to detect imposters in the network. Detected imposters are then prevented from communicating with other sensor nodes by means of the following simple, yet effective *quarantining* mechanism.

Each node maintains a *quarantineList* which contains node IDs identified to be used by an imposter. A sensor node will not send or receive any data from any node whose ID is in that list, hence, effectively keeping those nodes quarantined. In the following sections, we will develop distributed schemes which, through local collaboration and total absence of coordination, manage to detect imposters while keeping the number of false-positives as low as possible.

4.1. Detection mechanism

We consider a mobile sensor network where nodes relay information among themselves and, if necessary, sensed data can reach the base station through the use of appropriate MWSN routing algorithms ([25–27]).

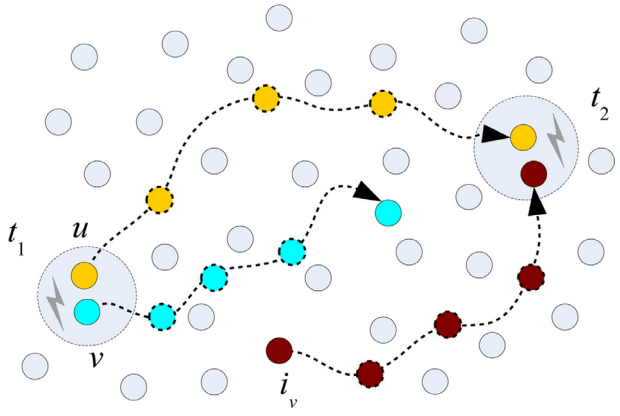

Fig. 1. Detection mechanism. Node u was able to detect the presence of an imposter of v at time t_2.

To detect an imposter the following simple mechanism is used:

"*When two sensor nodes meet for the first time, each node generates a random nonce, stores it in its memory, and sends it to the other node. The next time these nodes meet again, they request each other for the values they exchanged in their previous meeting. If a node cannot reply or replies with the wrong number then it is treated as a imposter and the ID of the node is considered compromised.*"

Hence the nonce values are used to detect existence of imposters and they are changed after each successful communication and maintained in a *nonceList*. The *nonceList* is maintained individually by each node and contains the nonce values expected from other nodes as well as the values to be sent to other nodes for successful authentication. To exemplify this authentication process further, consider Fig. 1 in which two nodes u and v meet for the first time at time t_1, exchange their nonces and then each one follows its random path. At some earlier time, an adversary was able to compromise node v and create an imposter with the same ID (in the figure this is indicated by node i_v). Node u, unaware of this event, meets again with a node bearing the ID of v at time t_2, thus it expects to receive the nonce it sent to v during the previous encounter at time t_1. The imposter i_v is unable to provide this information, thus u knows that ID of v has been compromised.

It should be clear from this discussion, that for imposter detection to take place, a sensor node needs to encounter both the legitimate node and its imposter one after the other (not necessarily in that order) as the node encountered second will be unable to reply with the correct nonce exchanged with first node. Once this

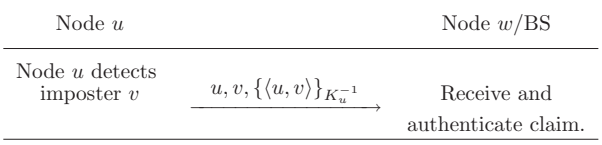

Fig. 2. A claim about an imposter propagated to either the base station or other nodes.

happens, it marks the node's ID as a compromized one and adds it to its *quarantineList*. We call this type of detection *direct detection* as the node is certain it has spotted the existence of an imposter carrying a specific ID. Information about the imposter can now be relayed to other nodes or the base station (BS) in the form of a *claim* (Fig. 2).

4.1.1. Authenticating claims

A claim is a pair $\langle detector_{id}, imposter_{id}\rangle$ that bears the signature of the node making the claim. Using an identity based signature scheme, if node u sends a claim that node with ID v is being used by an imposter, the claim is signed using u's private key K_u^{-1} and has the form $\{\langle u, v\rangle\}_{K_u^{-1}}$. The receiver authenticates the claim by generating u's ID-based public key $K_u = F(u)$ and verifying the signature.

While radio communication consumes most of the energy in sensor network protocols, and thus protocols with unnecessary communication overhead should be avoided, in the following we describe an authentication alternative that relies only symmetric cryptography primitives such as hash functions. While this approach requires less computation to generate and verify signatures, signature size grows considerably, thus requiring more communication overhead than the ID-based solution.

To introduce the asymmetry in signing provided by public key cryptography, r-times signatures ([23,24]) can be used. An r-time signature scheme is similar to a public-key scheme in that it can be used to sign messages that can be verified using publicly known information. These signatures decrease dramatically the signing and verification time compared to public-key signatures, however, one can only sign up to r messages with a given key pair. Then one must generate a new signing key to sign further messages, otherwise security degrades. The new public key that is needed to verify new messages, can be chained to the previous one, thus trustworthiness of public keys is still ensured.

In [24], such a scheme was developed that is tuned for use in sensor networks. The secret key consists of t random l-bit values arranged in a predefined number of Merkle trees whose leaves correspond to these secret values while intermediate nodes contain hashes of their children values. The roots of these trees are treated as the public key of the scheme. To sign a message m, a subset of the secret values is released along with their corresponding authentication paths in the trees. To verify a signature, a node simply re-evaluates the authentication values provided with the signature and checks to see if they match the roots of the trees contained in the public key.

Generation and verification of signatures is very efficient as it requires only hash and comparison operations. Using the parameters suggested in the paper, a signature which is 1200 bytes long, can be verified using less than 20 hash computations or about 200 ms in ordinary sensor nodes. However, while signature generation and verification time is negligible, the time to transmit such a signature is prohibitive. Thus the use of ID-based cryptography seems like the best alternative, despite its higher verification cost. As message transmission accounts for the majority of energy consumption in sensor networks, energy can be conserved by sending smaller signed messages but requiring a higher computation overhead for verification.

4.2. Distributed schemes

Depending on how the nodes relay the claims, two distributed schemes can be implemented. For completeness, we start with a simpler one that uses the base station (BS) as a collector of claims. Then we proceed with a fully distributed one.

4.2.1. Base station scheme

In this case the sensor network employs a centralized base station to which all sensed data is relayed. The base station can also broadcast authenticated information to all network nodes whenever necessary. When a node detects an imposter, it generates an imposter claim message and sends it to the base station using any routing protocol appropriate for MWSNs.

If the base station receives a number of claims against a node ID that exceeds a predefined threshold (to be defined shortly), it concludes that node ID is used by an imposter so it broadcasts to all sensor nodes a message to quarantine this imposter. This type of detection, based on claims received by the base station, is called *referred detection*.

4.2.2. Fully distributed scheme

We now consider the case where imposter detection and quarantining is *fully distributed*. In such a case the sensor nodes need to quarantine imposters individually *without* involvement of the base station. To achieve this, each sensor node maintains a *claimsList* which contains claims about node IDs that are being used by imposters. When two sensor nodes meet, they exchange their quarantine lists after they have authenticated each other successfully. The received *quarantineList* is added as claims in the node's *claimsList*. When the number of claims exceeds a predefined threshold against a node ID (to be defined shortly), the sensor node quarantines the imposter by adding it to its quarantine list. We call this mechanism *distributed referred detection*.

An example of this process is depicted in Fig. 3. The claims and quarantines list of each node before the exchange are shown in Fig. 3(a). We can see for example that node u has quarantined nodes 4 and 5, and has received one claim for node 15, two claims for node 3, and one claim for node 23 after exchanges with other nodes in the network. Similarly, node v has quarantined nodes 3 and 6, and has received one claim for node 7. Fig. 3(b) shows what happens after the exchange of the quarantined lists. The quarantined IDs of one node are copied to the claims list of the other node. For example, a new entry is created for node 6 in the claims list of u, while the number of claims for node 3 increases by one.

One may wonder, however, why do we insist that only quarantined nodes are moved to the claims list of the other node and not all IDs in both the claims and quarantined lists? The answer is that in the second case malicious nodes would be able to "infect" the claims lists of legitimate nodes in the network at a faster pace, since wrong accusations would have been propagated not only by malicious nodes as in the first case but by legitimate nodes as well. As encounters among legitimate nodes would be more frequent that encounters will malicious nodes, this would result in genuine nodes being blacklisted at a faster rate.

It should be obvious now that the use of claims in both schemes may lead to *erroneous* conviction of legitimate nodes as the imposters themselves may try to blacklist nodes by propagating claims about them, either towards the base station or to other nodes, depending on the scheme used. Hence we need to argue about the *soundness* and *completeness* of the detection methodologies. Intuitively, a detection method is sound if it never erroneously claims that a valid node is an imposter, i.e., there are no false-positives, and it is complete if it always detects nodes that are imposters, i.e., there are no false negatives. More formally, these

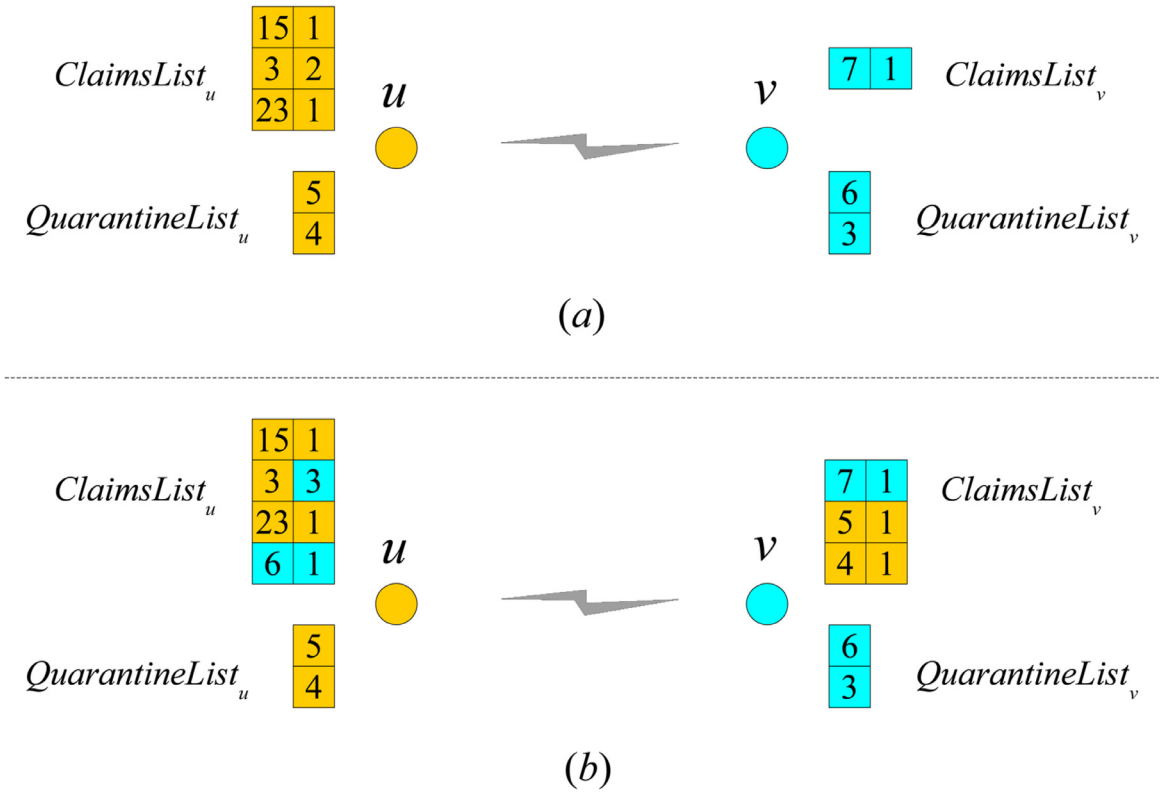

Fig. 3. Exchange of claims between two nodes u and v according to the fully distributed scheme.

properties can be defined as follows:

Definition 1 (Soundness and Completeness)**.** Denote by $D(s) \in \{0, 1\}$ the output of an imposter detection method D on a node $s \in S$, i.e. whether s is an imposter (replica) of a node in the MWSN S. Let also I denote the set of imposters of S, i.e. the set nodes compromised and replicated by an adversary. An imposter detection method D is complete if and only if $\forall i \in S$, $i \in I \Rightarrow D(i) = 1$. An imposter detection method D is sound if and only if $\forall i \in S$, $D(i) = 1 \Rightarrow i \in I$.

In what follows we will prove the completeness and soundness of the distributed schemes that make use of claims to speedup detection. Doing so, we will first assume that the number of imposters M is known in advance to all network nodes. Later on (Section 4.3) we will develop an *adaptive* scheme that eliminates the need to know the number of imposters M. This way the network will respond in a self-healing manner, adapting its level of security as more imposters show up in the network.

So, let's assume that the adversary have compromised M node IDs in the network and that M is known in advance to the network designers. Obviously this is a strong assumption to make, however, we will see how it can be removed in the fully adaptive, distributed scheme. Knowledge of M gives rise to the following simple strategy for treating a node as an imposter:

"*Blacklist a node only if you collect*at least$M + 1$ *claims about it.*"

Lemma 1 (Completeness)**.** *All imposters in the network are detected and quarantined when M is known.*

Proof. Completeness is a natural follow-up of the detection process. As sensor nodes are constantly moving in the coverage area, given sufficient time, each node will encounter the imposters as well the nodes they are cloning, and they will quarantine the imposters through direct detection. Alternatively, some sensors may quarantine imposters through referred detection after receiving at least $M + 1$ claims by other nodes in the fully distributed scheme, or after the base station has issued a quarantining message as described in the base station scheme. In both cases, all imposters will be detected one by one. □

Lemma 2 (Soundness)**.** *If M is known, only the node IDs that have been cloned by an adversary will be quarantined.*

Proof. As the number of imposters is bound by M, no collections of imposters can create more than M accusations (claims) to falsely quarantine a legitimate node u. Recall that a claim is a *signed* message bearing the signature of the *detector* and having the form $\langle detector_{id}, imposter_{id}\rangle$. Hence the set of claims $\langle detector_{id}, u\rangle$ against u can never be more than M. As both the base station (in the base station approach) and the rest of the nodes (in the fully distributed approach) are aware of M and need to see at least $M + 1$ claims in order to quarantine a certain node ID, we conclude that no false accusations can be made. □

Looking at Figs. 4 and 5, in Section 5, we see the behavior of the distributed schemes for a network of 500 nodes and varying node densities or number of imposters, respectively. The base station approach is obviously the fastest, however the fully distributive approach is the preferred one as no claims need to be forwarded to the base station, a process which can quickly drain the energy of nodes. In the next section, we will move one step further, eliminating the need to know M in order to detect imposters.

4.3. Fully adaptive, distributed scheme

Lemmas 1 and 2 above suggest that if the number of imposters M in the network is known, then setting the claims threshold to $M + 1$ is sufficient to neutralize any attempt by imposters to falsely quarantine legitimate nodes. In principle, however, it is not easy to know M in advance. In what follows we will relax this assumption and propose *adaptive* claims threshold schemes to quarantine

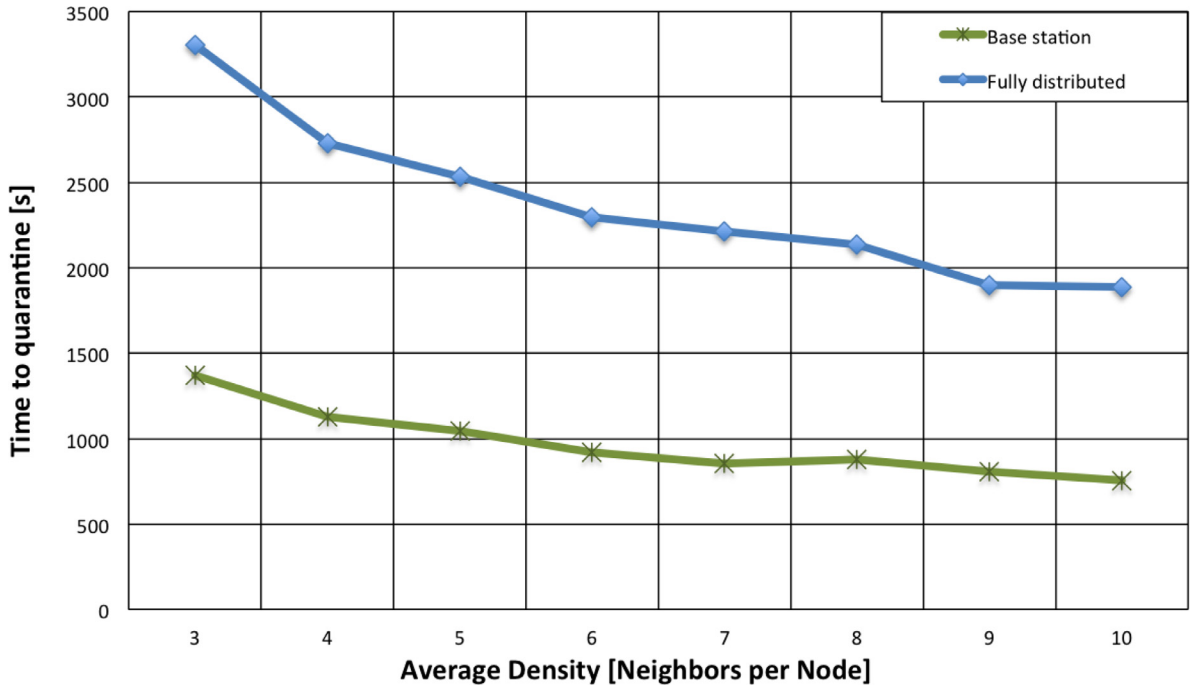

Fig. 4. Time to quarantine an imposter as a function of neighborhood density.

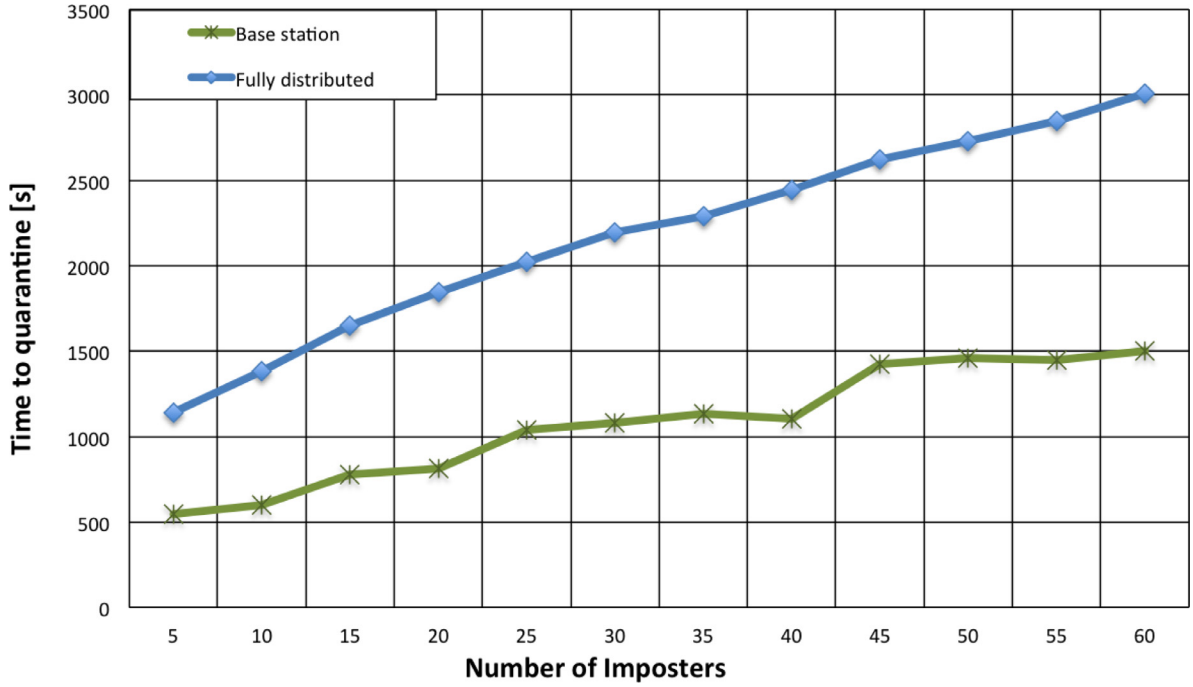

Fig. 5. Time to quarantine as a function of the number of imposters.

imposters efficiently even when their number is not known. For this purpose, we will introduce a new variable M_g, standing for a "guessed" number of imposters, based on which the referred detection scheme will require $M_g + 1$ claims to quarantine a node.

Our next objective is to devise a scheme to *update* the value of M_g in a way that it quickly *converges* to the correct value M, even if this value is not known ahead of time. In the upcoming sections we will only concentrate on the fully distributed scheme (Section 4.2.2) as it is fully decentralized, more cost-effective and does not suffer from a single point of failure as in the base station case (Section 4.2.1). However, the same techniques also apply to the base station scheme.

To gain some intuition on the impact M_g will have on detecting imposters, first consider the case where M_g is greater than the real number of imposters, i.e. $M_g > M$. Given sufficient time, all nodes in the network will meet the imposters and they will collect at least $M_g + 1$ claims for anyone of those. Hence eventually, all imposters will be quarantined. Additionally, imposters cannot accuse other nodes as their number is smaller than M_g and certainly smaller than the claims threshold $M_g + 1$. Hence both completeness and soundness are guaranteed. These observations suggest that if the network designers have a clue about the value M, they can pick a value M_g close to (but larger than) the real M and be assured that all imposters will be detected. The "penalty" here

is *increased* detection time as all nodes must collect at least $M_g + 1$ claims for any imposter.

To reduce detection time, one may start with a conservative value for M_g and gradually increase it as time passes. However, if M_g is selected such that $M_g < M$, it is easy to see that completeness still holds but soundness is no longer achievable as imposters may falsely accuse legitimate nodes; they can easily generate at least $M\ (\geq M_g + 1)$ claims for any particular node. Despite this, the latter method is a more preferable one since it is unlikely that a network will start with a large number of imposters (however, we leave the first one as a viable alternative). Here we envision the case where an adversary will compromise more nodes as time goes on rather than having compromised a large number of nodes initially and running the risk of being detected.

In summary, starting with a small value for M_g helps achieve quick referred detection, while increasing its value limits the false-positives effect incurred by imposters; imposters will only be able to cause other nodes to be falsely quarantined as long as M_g is smaller than M. We now consider two schemes to increase the value of M_g to reach $M_g \geq M + 1$, the *doubling* and the *incremental* schemes.

4.3.1. The doubling scheme

Based on the above discussion, we propose the following strategy executed *individually* by every node:

"*Start with $M_g = 1$. Whenever the number of quarantined nodes reaches M_g, clear the quarantineList and set M_g equal to twice its previous value.*"

A snapshot of these ideas is shown in Algorithm 1 which illustrates how the quarantine list of node i is updated after interacting with node j. Here by "clearing" the quarantine list we mean that node IDs that have been affirmed by *direct* detection to belong to imposters are still quarantined, while *referred* quarantined nodes are released provided the number of claims against them is less than the new threshold.

Algorithm 1 Updating *quarantineList* in Node i.

```
ReceiveQuarantineList (nonce_j, quarantineList_j)
  receivedNonce_j ← nonce_j
  if storedNonce_j == receivedNonce_j
      storedNonce_j ← GenerateNewNonce(i)
      send(j, storedNonce_j)
      AddClaims(quarantineList_j)
      for each k in claimsList
        if ClaimAgainstNode(k)≥M_g
            QuarantineNode(k)
            AdjustClaimsThreshold()
  else
      QuarantineNode(j)
      AdjustClaimsThreshold()

AdjustClaimsThreshold()
  if quarantineList_i.size() ≥ M_g
      M_g ← 2 * M_g
      quarantineList_i.update()
```

The doubling strategy gives quarantined nodes the "benefit of the doubt". A node u no longer interacts with nodes which are in its quarantine list L. However, as M_g doubles and exceeds M, we will see that "forgetting" ensures that the quarantine list will eventually contain only imposters. Note that although the quarantine list is updated each time M_g is doubled, the claims list for the node remains intact. Therefore, each node needs only collect additional claims over the existing claims, limiting the overhead introduced by the scheme.

Looking at Fig. 6 in Section 5, we observe how well the adaptive strategy works in practice. Even if the number of imposters is not known in advance, the behavior of the algorithm clearly matches the one observed when M is known. Hence this is the method to be used in all practical situations.

4.3.2. The incremental scheme

Although the *doubling* scheme, as described above, quickly reaches $M_g \geq M + 1$ by doubling the value of M_g every time the number of quarantined nodes in L is equal to M_g, however, after each doubling, it takes longer time to quarantine the released imposters as M_g increases and even more so with higher number of imposters. There is a tradeoff between reducing the number of iterations, k, for M_g to reach $M + 1$ and how long it takes to quarantine imposters. To reduce the time it takes to quarantine all imposters, we propose the *incremental* scheme. This scheme increases M_g by a constant value each time the number of nodes in the quarantine list is equal to M_g. Based on this, we use the following strategy executed *individually* by every node:

"*Start with $M_g = 1$. Whenever the number of quarantined nodes reaches M_g, clear the quarantineList and increase the value of M_g by a constant value D.*"

An advantage the *incremental* scheme might have over the *doubling* scheme is the shorter time to detect all imposters by each node. To illustrate this, consider the case where $M = 9$. In the *doubling* scheme, at $M_g = 8$, when 8 nodes are quarantined in L, M_g becomes 16 and all quarantined nodes are released. To quarantine all imposters, a node needs to receive 8 *additional* claims for every imposter, even though one additional claim to reach $M + 1$ would be sufficient. The extra claims required here are a result of doubling M_g which may significantly exceed $M + 1$. Now consider the same case but with the *incremental* scheme and $D = 1$. When $M_g = 8$ and 8 nodes are in the quarantine list, M_g will be increased by one to $M_g = 9$ and all quarantined nodes are released. To quarantine all imposters in this case, a node needs to receive only one *additional* claim for every imposter. This will result in a faster imposter detection time than the *doubling* scheme even though it might require $k = M$ iterations.

The above example illustrates that the incremental scheme results in a shorter detection time when compared to the doubling scheme, which is also reflected in the simulation results shown in Fig. 7.

In the next section, we will see how both schemes eventually succeed in minimizing the number of false positives to zero. Hence both soundness and completeness are achieved. However another important issue is the duration a legitimate node is falsely quarantined. When legitimate nodes are placed in quarantine lists, until their legitimacy is proven and they are subsequently released, the network is deprived of their contributions and the effectiveness of the network is reduced. Hence, the duration a falsely quarantined node is placed in the quarantine list is a differentiating aspect among the various imposter detection schemes proposed. Therefore, it is of interest to compare the performances of the *doubling* and *incremental* schemes in this aspect.

Consider the case where *doubling* is used and $M = 33$. When a quarantine list contains 32 nodes, all quarantined nodes are released and the new claims threshold is doubled to $M_g = 64$. If soon after that, a legitimate node i is wrongly added, as a false-positive, to the quarantine list of a node j, node i needs to wait for 63 more nodes to be quarantined to trigger a *doubling* of M_g and the subsequent release of the quarantined nodes, including node i. On the other hand, consider the same scenario but using the *incremental* scheme with $D = 1$. When $M_g = 32$ and the number of quarantined nodes is 32, all quarantined nodes are released and the new claims threshold is now incremented by one to $M_g = 33$. If soon after that a legitimate node i is added as a false-positive to the quarantine

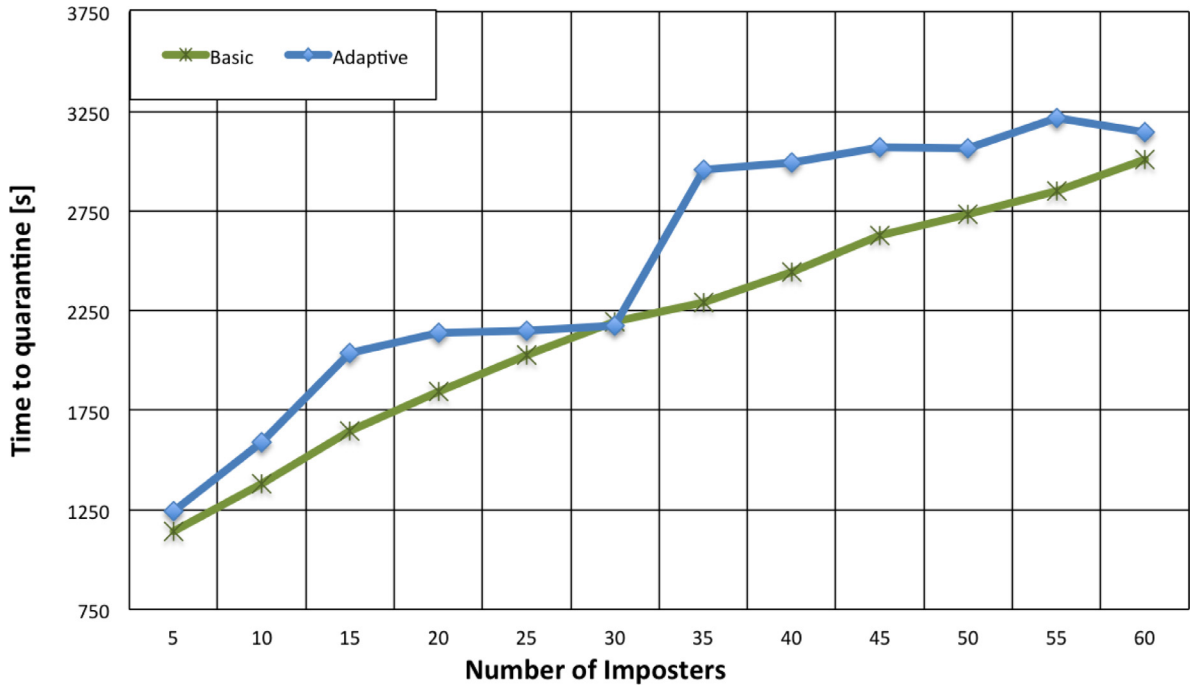

Fig. 6. Time to quarante in the adaptive (*M not* known) and the basic (*M* is known) schemes.

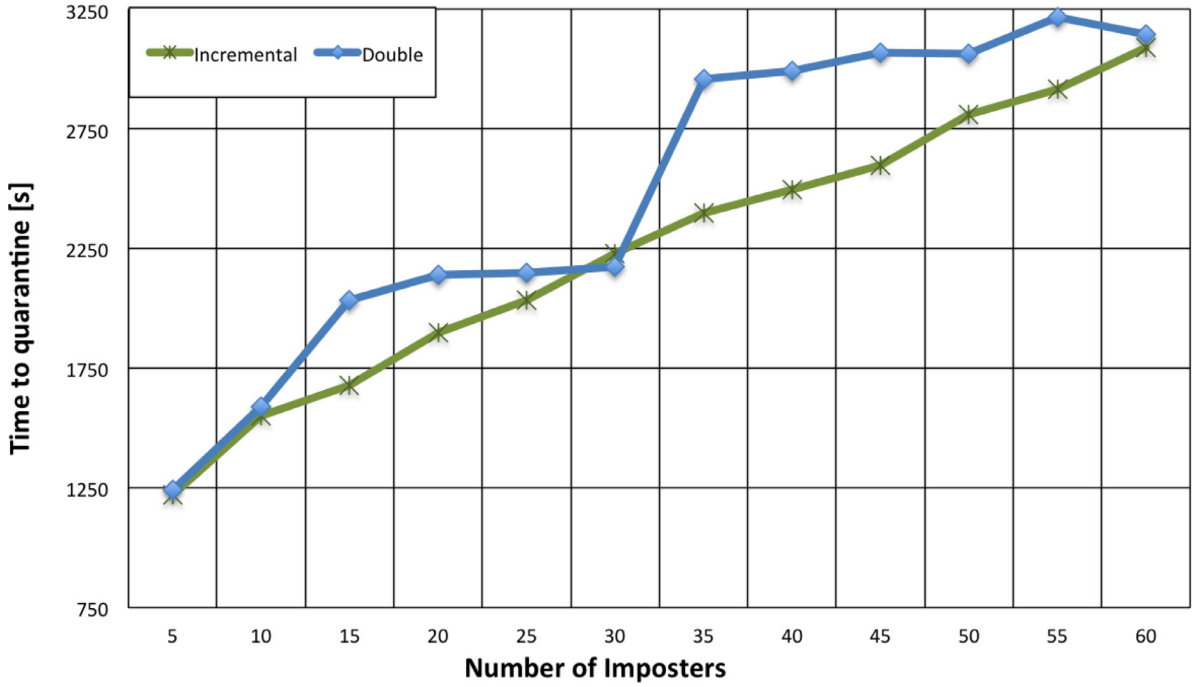

Fig. 7. Time to quarantine in doubling and incremental schemes.

list of a node *j*, node *i* will have to wait for only 32 additional nodes to be quarantined to trigger an increment in M_g and the release of the quarantined nodes including node *i*.

The above example illustrates that the incremental scheme results in a shorter false-positives quarantining time when compared to the doubling scheme, which is also reflected in the simulation results shown in Fig. 8.

4.4. Achieving soundness in the adaptive schemes

In our scheme, false-positives are included in the quarantine list of a legitimate node due to *false claims* received from imposters or other legitimate nodes. A false claim refers to a claim against a legitimate node. A legitimate node that propagates false claims received from imposters to other legitimate nodes is called a *deceived node*. We now discuss as to how legitimate nodes become deceived nodes and increase the number of false-positives in the quarantine lists of legitimate nodes. Then, we present a modification to the proposed scheme to eventually eliminate all deceived nodes and in turn reduce the number of false-positives. Since each node independently increases its claims threshold M_g, it is possible that M_g values of nodes may differ. As a result, some legitimate nodes for which $M_g < M$ may include false-positives in their quarantine lists due to false claims, received from imposters. In turn, these de-

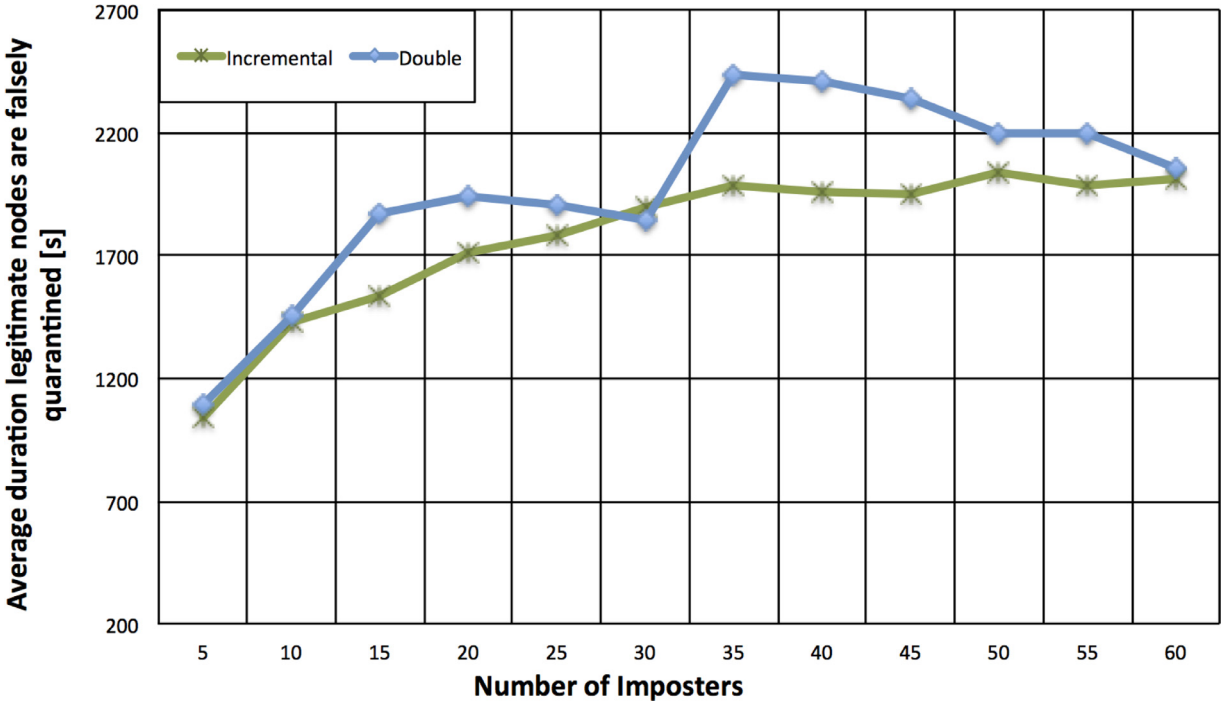

Fig. 8. Total duration legitimate nodes are blacklisted in doubling and incremental schemes.

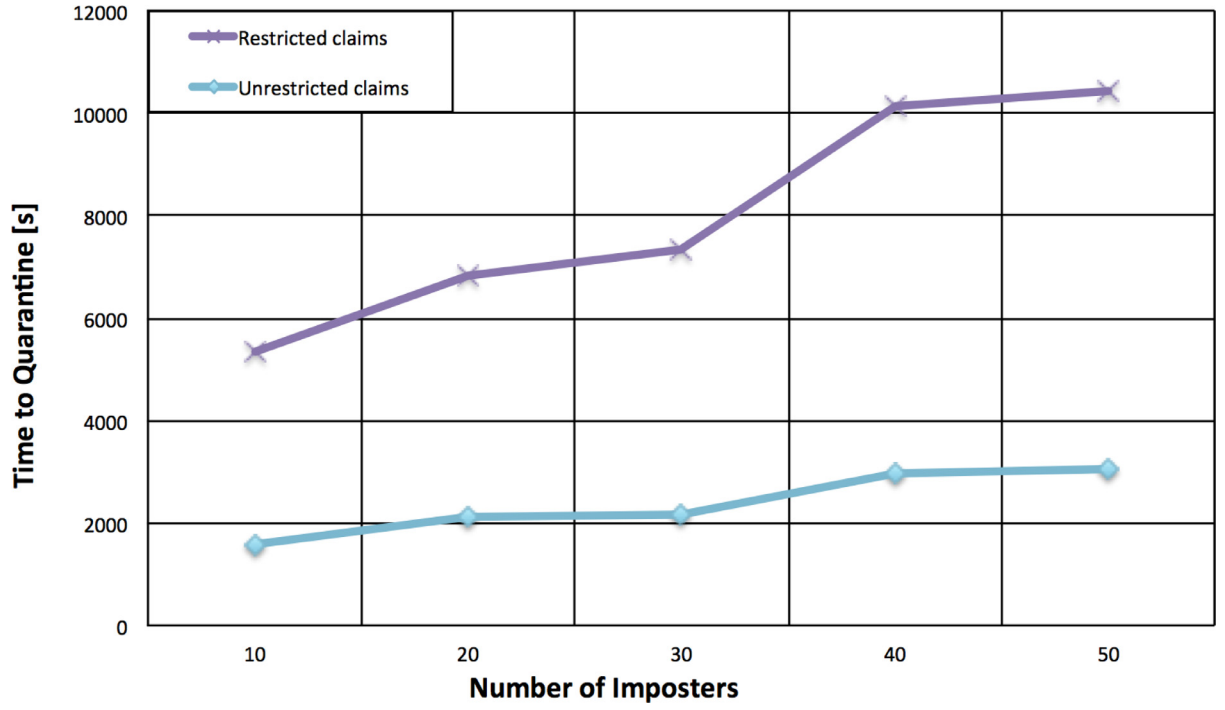

Fig. 9. Time to quarantining in restricted and unrestricted claim schemes.

ceived nodes may also send false claims against false-positives in their quarantine lists to nodes for which $M_g > M$ hold, and cause more than M_g claims to be received leading to quarantining of a legitimate node by a node for which $M_g > M$ holds. As a result, false-positives remain in the quarantine lists of legitimate nodes even after $M_g > M$ holds for them.

Let *direct detection claim* be a claim which is produced after directly detecting an imposter, and let *referred detection claim* be a claim produced due to receiving number of claims greater than M_g. To reduce the number of false-positives in the network, we propose to sent only directly detected claims, while referred detected claims are never sent. Note that this proposal does not stop suspected nodes from being quarantined due to referred detection, but rather if a node is placed in the *quarantineList* due to referred detection, no claims against it are sent to other nodes upon an encounter.

This scheme where some claims are restricted from being sent is referred to as the *restricted claims scheme*, whereas, the one

where no restrictions are applied is referred to as the *unrestricted claims scheme*. The modified scheme effectively reduces the claims sent by imposters against legitimate nodes and the claims against directly detected imposters by legitimate nodes. Observe that now no node remains as a deceived node after $M_g > M$ holds for it. Since $M_g > M$ holds for each node eventually, no deceived node remains in the network in the long run. Subsequently, no false-positives remain after $M_g > M$ holds for each node. This scheme reduces the false-positives until $M_g > M$ holds for each node and eliminates false-positives when $M_g > M$ holds for each node. The following lemma proves the soundness of this modified detection strategy.

Lemma 3. *The doubling strategy ensures that at any point in time the number of false-positives directly caused by the imposters is bounded by M. When M_g exceeds M for a node, its quarantineList will contain no more falsely accused nodes.*

Proof. Consider the *quarantineList* L of a node u. For any value of M_g, this node can have at most M_g nodes in L. In the worst case, all these nodes will be legitimate ones, framed by the real imposters. When L gets filled, M_g and the size of L are doubled, L is cleared, and twice as many claims are required to quarantine a node. This process continues for k iterations, where $k = \lceil \log M \rceil$, at which point M_g becomes larger than M. From then on, imposters can no longer frame legitimate nodes since their number is less than the required threshold $M_g + 1$. Hence soundness is achieved. As L is cleared every time M_g is doubled, it is not difficult to see that the maximum number of framed nodes occurs at stage $k - 1$. Their number is equal to $2^{k-1} = 2^{\lceil \log M \rceil - 1} < M$. It should also be clear that in the next stage when M_g becomes bigger than M, no more false accusations will be possible and the *quarantineList* will contain only imposter IDs. □

It is easy to see that Lemma 3 extends to the incremental scheme as well.

5. Experimental results

We have evaluated the proposed schemes using a network consisting of 500 nodes randomly moving in an area of size 500 m × 500 m with an average speed of 3.0 m/s. Two nodes will be able to exchange messages if they are within communication range of each other. The range was set such that each node has an average of 6 neighbors during each simulation run.

At the beginning of the run, sensor nodes are placed randomly in the simulation area. Similarly, imposters clone different legitimate nodes and are placed at random locations in the network. Each node follows the random waypoint model of movement where it chooses a random point in the network, moves to that point in a straight line according to its speed, then chooses another point and so on. Movement takes place at discrete steps, where each step represents one second in the simulation environment. At each step, when two nodes meet they communicate and exchange nonces according to the algorithm to detect and quarantine imposters. The simulation terminates when every legitimate node (except the cloned ones) quarantines *all* imposters in the network. Time to quarantining is measured as the number of steps required to achieve such full detection. To ensure statistical validity, each experiment was repeated 100 times and the average value is depicted in the graphs.

The effectiveness of the proposed schemes was analyzed mainly according to the following two metrics:

- time taken to quarantine all imposters under a specific scheme, and
- number of false positives in the adaptive schemes.

The performance of the various schemes was studied as a function of the average node density (number of neighbors) and number of imposters in the network. Typically, densities ranged from 3 to 10 and number of imposters from 5 to 50. The density was increased by varying appropriately the communication range of nodes.

Figs. 4 and 5 consider detection time as a function of the average network density and number of imposters, respectively. The graphs clearly demonstrate that the base station scheme (Section 4.2.1) performs better than the fully distributed scheme (Section 4.2.2). This is due to the different imposter quarantining techniques employed by each scheme. When receiving the required number of claims, the base station *broadcasts* a message to quarantine the imposter, while in the distributed scheme this information needs to be forwarded *among* the nodes in a hop-by-hop manner until it becomes available to all of them. However, the loss in performance in the second case is outweighed by the fact that detection is completely decentralized.

Fig. 6 compares the time to quarantining in the basic (Section 4.2) and the adaptive (Section 4.3) scheme (using doubling), where M is known in the first and is approximated with M_g in the second. The adaptive scheme requires slightly more time but no prior knowledge of the number of imposters present in the network; hence, it is more realistic and useful in practice. At some points the two schemes have identical performance which is due to the value of the claims threshold. For example, when there are 16 imposters, M_g increases to 16 which is equal to M and the two schemes require the same number of claims for referred detection. But when $16 < M \leq 32$ (for example $M = 17$), M_g will increase up to 32, hence the adaptive scheme requires 33 claims for referred detection while the basic scheme requires 18 claims only. This explains the increased time to quarantine imposters in the adaptive case.

Fig. 7 shows that the *incremental* scheme results in a faster detection time when compared to the *doubling* scheme as discussed in Section 4.3.2. Similarly, Fig. 8 shows that the *incremental* scheme reduces the average duration legitimate nodes are falsely quarantined as compared to the *doubling* scheme. This is due to the doubling effect on the claims threshold M_g which increases rapidly as number of imposters increases, and hence requires more nodes to be quarantined to reach the doubling point to release the quarantined nodes including any false-positives, as discussed in Section 4.3.2.

Fig. 9 takes a closer look at the effects of restricting claims to be based only on direct detection on the total imposter detection time as compared to the case where claims are based on both direct and referred detections, (recall Section 4.4). As it can be seen and is expected, limiting claims to be based on only direct detection increases the detection time compared to the scheme that employs both referred and direct detection based claims. This is due to the fact that to quarantine an imposter in the restricted claims scheme a legitimate node needs to either directly detect the imposter or receive M_g different claims from those nodes that have directly detected the imposter. Whereas in the unrestricted claims scheme a node needs to either directly detect the imposter or meet M_g different nodes which have the imposter placed in their quarantine lists through direct or referred detection.

On the other hand, Fig. 10 shows that limiting claims to be based on direct detection only yields a very short average quarantining time for false-positives. This is due to limiting claims to be sent by direct detection nodes that effectively eliminates the existence of deceived nodes in the network, and in turn only imposter nodes can cause a false-positive. However, imposters are only effective in causing false-positives while $M_g < M$ holds as proved by Lemma 3. Therefore, after $M_g > M$ holds for a node, it no longer has any false-positives in its quarantine list.

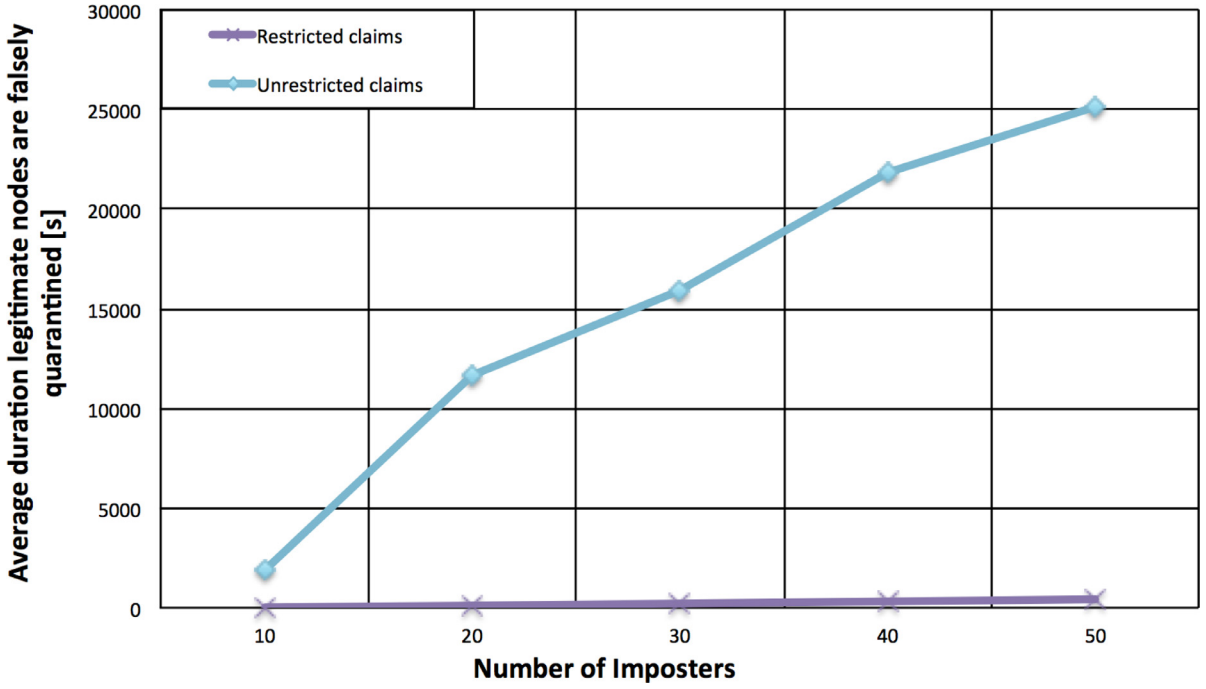

Fig. 10. Total duration legitimate nodes are blacklisted in restricted and unrestricted claim schemes.

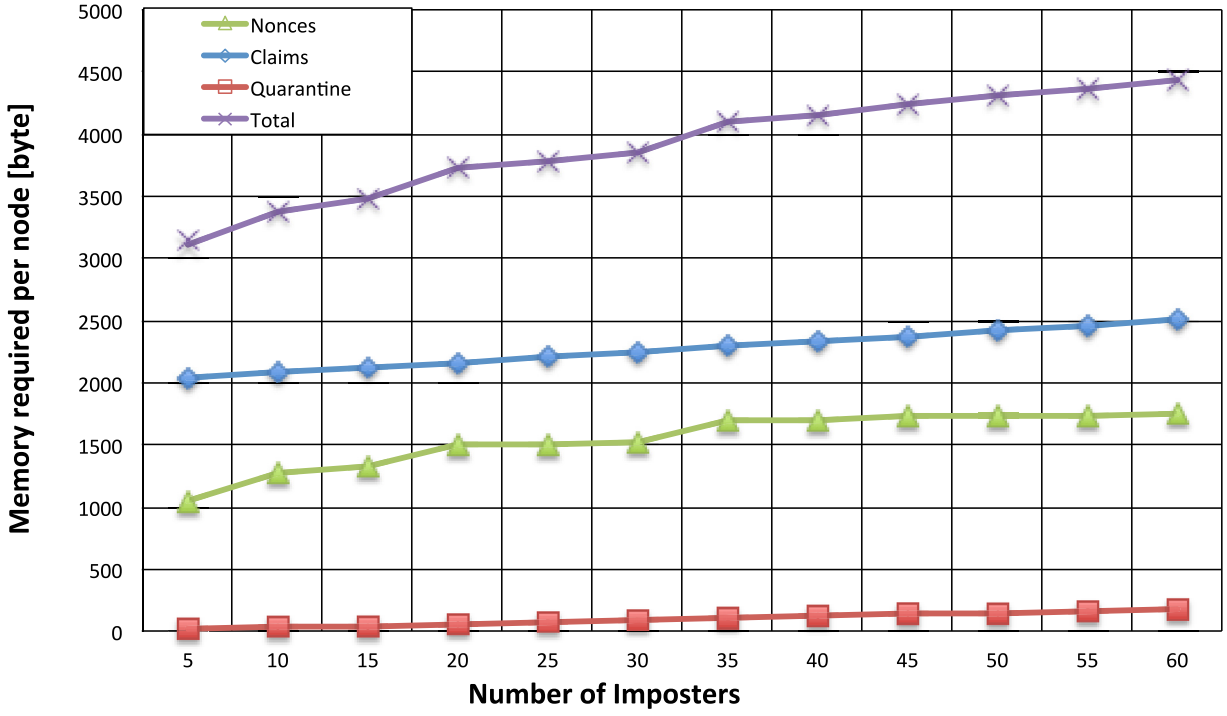

Fig. 11. Average memory requirements per node for imposter detection.

Although simulation results have shown the effectiveness of our proposed imposter detection scheme, a look at the memory requirements of such a scheme is needed due to the limited resources of the sensor nodes in the network. Simulation results in Fig. 11 show that the memory requirements are within the capabilities of the commercially available sensor nodes, with the claim list using the highest percentage of the memory allocation when compared to the nonce values list. This is understandable since the claim list may contain several entries against a single node where the nonce list contains a single entry per node encountered in the network.

6. Conclusions

The node replication attack is one of the most insidious attacks in sensor networks. Although several countermeasures exist, almost all practical schemes assume a stationary network model where sensor nodes are fixed and immobile. In this work, we proposed solutions that can be used to detect imposters in *mobile* sensor networks, where nodes freely and randomly move around in the sensing region. These schemes are *fully distributed* and completely decentralized. Contrary to prior work, the proposed schemes can effectively detect and quarantine the presence of *mul-*

tiple imposters faking the identity of *different* legitimate nodes in the network.

We have proved the completeness and soundness of the proposed detection methods and came up with schemes that are *adaptive* in their operation; even if the number of imposters is unknown, the adaptive schemes not only find all imposters but also respond in a self-healing manner eventually bringing the number of false-positives to zero. Hence they constitute the preferred method for realistic sensor network deployments. Our findings were coupled with both analytical and experimental results, proving the viability of our proposals.

Note that when an imposter is detected, the ID possessed by the imposter is considered compromised. As a result, the remaining nodes stop communicating with the imposters but also with the legitimate nodes bearing the same IDs. In terms of future work, it would be interesting to investigate the possibility of re-instating the legitimate nodes back in the network. This could be done by reprogramming those nodes [24,28], hence installing new software, new cryptographic material, and as a result, new IDs to them.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments that greatly contributed to improving the quality of the paper. This work is supported by Kuwait University, Research grant no. EO 03/12.

References

[1] S.A. Munir, B. Ren, W. Jiao, B. Wang, D. Xie, J. Ma, Mobile wireless sensor network: architecture and enabling technologies for ubiquitous computing, in: Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on, 2, 2007, pp. 113–120. IEEE

[2] S. Ehsan, K. Bradford, M. Brugger, B. Hamdaoui, Y. Kovchegov, D. Johnson, M. Louhaichi, Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals, Wireless Commun. IEEE Trans. 11 (3) (2012) 1220–1227.

[3] B. White, A. Tsourdos, I. Ashokaraj, S. Subchan, R. Żbikowski, et al., Contaminant cloud boundary monitoring using network of uav sensors, Sens. J. IEEE 8 (10) (2008) 1681–1692.

[4] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: Security and Privacy, 2005 IEEE Symposium on, 2005, pp. 49–63. IEEE

[5] T. Giannetsos, T. Dimitriou, Spy-sense: spyware tool for executing stealthy exploits against sensor networks, in: Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy, 2013, pp. 7–12. ACM

[6] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, in: Ad Hoc Networks, 1, 2003, pp. 293–315.

[7] I. Krontiris, T. Giannetsos, T. Dimitriou, Launching a sinkhole attack in wireless sensor networks; the intruder side, in: Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing, 2008a, pp. 526–531. IEEE

[8] I. Krontiris, T. Giannetsos, T. Dimitriou, Lidea: a distributed lightweight intrusion detection architecture for sensor networks, in: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, 2008b, p. 20. ACM

[9] J. Hall, M. Barbeau, E. Kranakis, Detection of transient in radio frequency fingerprinting using signal phase, in: Wireless and Optical Communications, 2003, pp. 13–18.

[10] V. Bhuse, A. Gupta, Anomaly intrusion detection in wireless sensor networks, J. High Speed Netw. 15 (1) (2006) 33–52.

[11] S. Hussain, M.S. Rahman, Using received signal strength indicator to detect node replacement and replication attacks in wireless sensor networks, SPIE Defense, Security, and Sensing, pp. 73440G–73440G, International Society for Optics and Photonics, 2009.

[12] W.T. Zhu, J. Zhou, R.H. Deng, F. Bao, Detecting node replication attacks in wireless sensor networks: a survey, J. Netw. Comput. Appl. 35 (3) (2012) 1022–1034.

[13] J.-W. Ho, M. Wright, S.K. Das, Fast detection of replica node attacks in mobile sensor networks using sequential analysis, in: INFOCOM2009,IEEE, 2009, pp. 1773–1781. IEEE

[14] C.-M. Yu, C.-S. Lu, S.-Y. Kuo, Efficient and distributed detection of node replication attacks in mobile sensor networks, in: Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th, 2009, pp. 1–5. IEEE

[15] C.-M. Yu, C.-S. Lu, S.-Y. Kuo, Mobile sensor network resilient against node replication attacks, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on, 2008, pp. 597–599. IEEE

[16] W.T. Zhu, J. Zhou, R.H. Deng, F. Bao, Detecting node replication attacks in mobile sensor networks: theory and approaches, Secur. Commun. Netw. 5 (5) (2012) 496–507.

[17] E.A. Alrashed, M.H. Karaata, Imposter detection in mobile wireless sensor networks, Int. J. Comput. Commun. Eng. 3 (6) (2014) 434.

[18] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, P. Kruus, TinyPK: securing sensor networks with public key technology, in: The Proceedings of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), 2004.

[19] A. Liu, P. Ning, TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks, in: The Proceedings of the 7th International Conference on Information Processing in Sensor Networks, 2008.

[20] L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, R. Dahab, Tinytate: computing the tate pairing in resource-constrained sensor nodes, in: Proceedings of Sixth IEEE International Symposium on Network Computing and Applications (NCA), 2007.

[21] C.C. Tan, H. Wang, S. Zhong, Q. Li, IBE-Lite: a lightweight identity-based cryptography for body sensor networks, IEEE Trans. Inf. Technol. Biomed. 13 (6) (2009).

[22] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology, 1984, p. 4753.

[23] L. Reyzin, N. Reyzin, Better than BiBa: short one-time signatures with fast signing and verifying, in: Proceedings of the 7th Australian Conference on Information Security and Privacy (ACISP), 2002.

[24] I. Krontiris, T. Dimitriou, Scatter-secure code authentication for efficient reprogramming in wireless sensor networks, Int. J. Sens. Netw. (2010).

[25] C. L. M. Arboleda, N. Nasser, Cluster-based routing protocol for mobile sensor networks, in: Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, 2006, p. 24. ACM

[26] M. Yu, J.H. Li, R. Levy, Mobility resistant clustering in multi-hop wireless networks, J. Netw. 1 (1) (2006) 12–19.

[27] A.R. Khan, S. Ali, S. Mustafa, M. Othman, Impact of mobility models on clustering based routing protocols in mobile wsns, in: Frontiers of Information Technology (FIT), 2012 10th International Conference on, 2012, pp. 366–370. IEEE

[28] L. Mottola, G.P. Picco, Programming wireless sensor networks: fundamental concepts and state of the art, ACM Comput. Surv. (CSUR) 43 (3) (2011).

Dr. Tassos Dimitriou is affiliated with the Department of Computer Engineering at Kuwait University (KU) and the Research and Academic Computer Technology Institute (CTI) - Greece. Prior to that he was an Associate Professor at Athens Information Technology, Greece, where he was leading the Algorithms and Security group, and adjunct Professor in Carnegie Mellon University, USA, and Aalborg University, Denmark. Dr. Dimitriou contacts research in areas spanning from the theoretical foundations of cryptography to the design and implementation of secure communication protocols. Emphasis is given in the authentication and privacy aspects for various types of networks like adhoc and sensor networks, RFID, smart grid, etc. Dr. Dimitriou is a senior member of IEEE, ACM and a Fulbright fellow. He is also a Distinguished Lecturer for ACM.

Ebrahim Alrashed received his Ph.D. and M.S. degree in computer engineering from The University of Southern California, Los Angeles, CA in 1997 and 1993. He is currently working as Assistant Professor in Department of computer engineering at Kuwait University. His research interests includes network security, mobile and wireless networks, cloud computing and sensor networks. Dr. Alrashed has memberships in professional organizations such as ACM Association of Computing Machinery.

Mehmet Hakan Karaata received his Ph.D. degree in Computer Science in 1995 from the University of Iowa. He joined Bilkent University, Ankara, Turkey as an Assistant Professor in 1995. He is currently working as a Professor in the Department of Computer Engineering, Kuwait University. His research interests include mobile computing, distributed computing, fault tolerant and autonomus computing.

Ali Hamdan received his Bachelor degree in Computer Engineering in 2013 from Kuwait University. He is currently working as a Research Assistant in Kuwait University. His research interests include distributed systems, fault tolerant, and network security.