Thwarting the Sybil Attack in Wireless Ad Hoc Networks

Diogo Miguel da Costa e Castro Mónica Oliveira
diogo.monica@ist.utl.pt

Instituto Superior Técnico

(Advisors: Professors Carlos Ribeiro and Luís Rodrigues)

Abstract. The present document is essentially a summarized version of the dissertation that will be developed within the scope of an MSc degree in Communication Networks, in the area of security in wireless ad hoc network. Wireless ad hoc networking is a technology that allows fast, easy, and inexpensive network deployment. Unfortunately, these advantages also make the task of an attacker simpler, as it is also easier to deploy a malicious node in the environment. To make the ad hoc network secure, one has often to use Byzantine fault-tolerance techniques, which typically rely on quorum based security protocols. However, quorums may be easily defeated if a single adversary can participate in the network with multiple identities, a behavior known as the Sybil Attack. This work addresses the problem of preventing the Sybil Attack in wireless ad hoc networks. It studies and compares different techniques that have been previously presented in the literature, and proposes a technique applicable to a broader range of networks.

1 Introduction

Wireless ad hoc networking is a technology that allows for fast, easy and inexpensive network deployment. These networks are usually multi-hop, where each node is able to forward data to other nodes to ensure network connectivity. This means that each node has to make routing decisions, in contrast to wired networks, where the responsibility of making decisions lies on specialized components (the routers). Ad hoc networks are also substantially different from infrastructured wireless networks, where nodes never communicate directly amongst themselves and all communication is performed via specialized nodes known as Access Points. Despite their known limitations in terms of scalability and overall capacity [1, 2], the decentralized nature, minimal configuration and self-healing abilities of wireless ad hoc networks, make them suitable for a variety of situations like search and rescue, recovery from natural disasters, or military conflicts.

Unfortunately, these advantages also make the task of an attacker simpler, as it is also easier to deploy a malicious node in this environment. To start with, the legitimate nodes of an ad hoc network are typically more vulnerable to

tampering than the nodes of a fixed wired network. Also, the membership and topology of a wireless ad hoc network can be very dynamic, making it easy for a malicious node to be inserted in the system. Thus, to make ad hoc networks secure, one has often to use Byzantine fault-tolerance techniques.

Generally, most Byzantine fault-tolerance technics rely on some form of quorum system [3]. In the context of wireless ad hoc networks, Byzantine quorum systems have been used for multiple purposes, including auto-configuration of IP addresses [4], node location [5], power saving protocols [6], mobility management [7] and reliable storage [8].

However, quorums may easily be defeated if a single adversary can participate in the network with multiple identities, a behavior known as the Sybil Attack [9]. Therefore, finding efficient techniques to defeat the Sybil Attack is key to build secure wireless ad hoc networks. This is the problem addressed in this work.

The different techniques that have been proposed in the literature to tackle the Sybil Attack, will be surveyed, analyzed and compared. As will be seen, most techniques require the pre-configuration of the nodes that are part of the network, since they are either based on a PKI or on some kind of pre-shared secret. This work will focus, however, on techniques that allow the auto-configuration of the nodes. It aims at proposing efficient techniques to counter the Sybil Attack in a broader range of ad hoc networks such as multi-hop networks.

The remainder of this document is organized as follows. Section 2 describes the motivation and goals for this work. Section 3 surveys previous work in this area. Section 4 describes the proposed technique. Section 5 analyses the problem of performance evaluation, followed by the work schedule in Section 6 and the concluding remarks in Section 7.

2 Goals

This work aims at analyzing and designing efficient techniques to mitigate the Sybil Attack in wireless ad hoc networks.

One of the open problems in the design of security mechanisms for ad hoc networks is the defense against sybil attacks. Most methods that attempt to mitigate these attacks are based on a centralized authority, needing a pre-shared secret and thus, pre-configuration. However, this requirement makes such solutions unfeasible to implement in many civilian environments, since one cannot assume a common administrative entity with access to every node, and trusted by all. The main goal of this work is, therefore to propose a technique which can reliably test for sybil identities, without requiring the a priori configuration of nodes. The proposed technique must be capable of operation in both 1-hop and multi-hop.

Expected results: The expected outcome of this work is:

- A security mechanism, based on radio resource testing techniques, to mitigate the Sybil Attack within the local radio coverage;

– An extension of this mechanism to multi-hop environments;
– The analysis, evaluation and enhancement of the performance of both mechanisms.

3 Context and Related Work

This work addresses security issues on wireless ad hoc and mesh networks, with a special emphasis on the sybil attack. This section introduces fundamental concepts, starting with a brief overview of the security issues present in this type of networks. Afterwards, the Sybil Attack will be explained, followed by the description of the existing techniques to address this attack.

3.1 Wireless Ad Hoc Networks

Infrastructured wireless networks require a fixed network structure, with centralized administration, for their operation. In contrast, wireless ad hoc networks consist of a collection of wireless nodes, all of which may be mobile. This scenario should allow the dynamical creation of a wireless network among nodes, without using any infrastructure or administrative support. There are essentially three kinds of ad hoc networks: Mobile Ad Hoc Networks (MANET), Sensor Networks and mesh Networks. While the basic principles of these three wireless networks remain the same, they have a few differences between them, making them worth of being separately addressed. See Figure 1 for a conceptual representation.

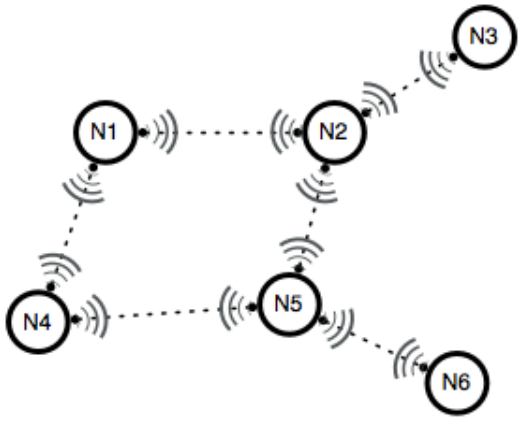


Fig. 1. Conceptual representation of an ad hoc network.

In the case of the MANETs, their main differentiating characteristic is the fact that the nodes are free to move without constraints, and organize themselves arbitrarily. Therefore, MANETs may have a highly dynamic membership

and mobility, which means that the network topology may change rapidly and unpredictably. In sensor networks, nodes are typically static and resource constrained. In order to save resources, nodes may put themselves in sleep mode, which also causes changes in the network topology, since this sleep mode affects the communication patterns among nodes. Finally, mesh networks combine mobile wireless nodes with energy-unconstrained static wireless nodes that support routing in the network. Mesh networks are, for instance, appropriate to expand the wireless network connectivity in regions where there is limited access to an infrastructured network.

3.2 Security

The fact that ad hoc networks do not necessarily rely on a fixed infrastructure, raises many challenges for their security architecture [10]. Issues that further complicate the design of this architecture in ad hoc networks are the vulnerability of the links, limited physical protection of each node, the absence of a certification authority, and the lack of centralized monitoring or management points [11], among others. Unfortunately, the numerous differences between ad hoc and wired networks make the former not eligible for the same security solutions as the latter. While the basic security requirements such as confidentiality, integrity and authenticity remain the same, ad hoc networks restrict the set of feasible security mechanisms that can be used, mostly due to performance issues.

The security requirements of any network depend vastly on the type of application. While there are some networks that operate in a safe and friendly environment, most of them are deployed in hostile environments, subject to constant threats from attackers. An example of such a hostile environment can be found in networks deployed for military communications in direct conflict areas. In this case, all network components are physically vulnerable to tampering and must satisfy very stringent requirements regarding confidentiality and resistance to denial-of-service attacks. While not all applications of ad hoc networks have this kind of strict security requirements, every security solution may have to address the limited power, memory, and CPU available in each node, while still managing to provide strong protection against threats. This trade-off between performance and protection makes the design and implementation of strong protective measures in ad hoc networks a non-trivial problem.

The design of a security scheme requires, in general, the discussion of the vulnerabilities that need to be addressed, the description of the fundamental security components, and finally, must take into consideration the attacks that currently exist to exploit those vulnerabilities. In what follows, several aspects will be addressed: the main vulnerabilities of wireless ad hoc networks; the essential security needs of such networks; finally, the main threats that violate such security needs (generally called *attacks*).

Vulnerabilities of ad hoc Networks From a security point of view, there are several reasons why wireless ad hoc networks are more vulnerable than their

wired counterparts. Many of these reasons derive from vulnerabilities of the wireless medium of communication. Some characteristics that make these networks particularly vulnerable to attacks, will now be discussed. In this context, attacks are defined as procedures launched by unauthorized entities or nodes within the networks, that exploit vulnerabilities with the intent of disrupting the network operation.

Lack of Secure Boundaries In a wired network, attackers need to have physical access to a node or to the medium in order to perform malicious activities. On the other hand, in wireless networks it is not possible to create the same sort of secure *boundary*. Once an attacker is in the radio range of any group of nodes, it can communicate with these nodes and attempt to join the network. As a result, ad hoc networks cannot rely on boundary lines of defense, such as firewalls and gateways, to protect the network from potentially harmful network accesses.

Furthermore, since all communications are performed over the air, ad hoc networks are specially vulnerable to attacks such as passive eavesdropping, active interference, leaking of secret information, data tampering, impersonation, message replay, message distortion, and denial-of-service [12]. These vulnerabilities raise serious concerns, since there are many applications in which confidentiality is one of the major issues. For example, in military applications, confidentiality of the information is one of the most important attributes, as discussed in [13]. Also, without any authenticity and integrity protection (these definitions will be provided further ahead in the text), an attacker is able to destroy, create or even manipulate messages, allowing it to, ultimately, compromise the entire network. Availability is also a central issue in ad hoc networks, that must operate in dynamic and unpredictable conditions. In civilian scenarios, availability has the greatest relevance for the user [14], and is also one of the most difficult properties to preserve, since wireless media are particularly susceptible to denial-of-service attacks, including resource exhaustion and jamming.

Threats from Compromised Nodes An attacker may attempt to compromise the links or the nodes of the ad hoc network. If the attacker gains control of one or more nodes in the network, he can then use these compromised nodes to execute further malicious actions. One of the challenges in face of this attack is to detect accurately the nodes that have been compromised, as discussed in several works of intrusion detection [15, 16]. Usually, compromised nodes are detected by monitoring their behavior. Unfortunately, in wireless environments, it is difficult to distinguish a truly misbehaving node from a node with a poor link quality [12]. In addition, and since the nodes that compose the ad hoc network can have a high behavioral diversity, it is hard to create effective policies that prevent all the possible malicious behaviors from every kind of existent node. Finally, the fact that nodes can join or leave the network with freedom, allows an attacker to frequently change target, and attack a different node in the network. This makes the task of tracking the malicious behavior performed by any compromised node inside the network, even more difficult.

Compromised nodes in the network may deliberately cause Byzantine faults. Byzantine faults are a subset of arbitrary faults, that denote inconsistent semantic faults (e.g., sending different messages to different recipients) [17]. Furthermore, a set of nodes may be compromised in such a way that their malicious incorrect behavior cannot be detected [12], making them a serious threat to the network. While compromised nodes may appear to be operating correctly, they can, for example, create new routing messages or advertise non-existent links, thus inflicting Byzantine faults on the system.

Lack of Centralized Management Facility The lack of centralized management has a significant impact on the design of security mechanisms for ad hoc networks, in particular because attack detection becomes a very hard task. For instance, attack detection based on traffic monitoring is challenging in highly dynamic and large scale ad hoc networks [18], since the malicious activities may be obfuscated by the frequent benign failures in wireless networks, such as path breakages, transmissions impairments and packet dropping (especially when attackers frequently change their targets and attack patterns).

On the other hand, the absence of an authorization facility provided by a pre-existing infrastructure, makes it very hard to distinguish trusted from untrusted nodes. This distinction is a line of defense that may be implemented by requiring trusted nodes to carry credentials that can be validated by the remaining trusted nodes. Unfortunately, in the case of ad hoc networks, no prior security association can be assumed for all network nodes. In consequence, algorithms that rely on the cooperative participation of all nodes can be attacked by adversaries that make use of this vulnerability to undermine decentralized decisions [19].

Restricted Resources Typically, in wireless ad hoc networks, some, or all of the network nodes rely on batteries for power supply. Thus, security mechanisms need to consider the fact that power is a limited resource. For instance, restricted power may be used to launch denial-of-service attacks [20]: the attacker, being aware that his targets are battery-restricted, may, for instance, continuously send packets to his targets, asking them to route those additional packets, or can induce target to make time-consuming computations. This sort of attack will exhaust the battery of the nodes in the network, and prevent them from answering legitimate service requests, since they will be quickly out of service.

Power is not the only scarce resource in wireless networks. For example, in sensor networks, besides power issues, nodes also have a very limited processing capacity [21]. Sometimes, the working memory of a sensor node is insufficient, even to hold the variables required for asymmetric cryptographic algorithms [12]. This restrictions impose the use of other, potentially less secure, security mechanisms.

Scalability The size of an ad hoc network is a variable that changes frequently [12]. This is due to node mobility and network partitions or merges. As a result, protocols and services for ad hoc networks, such as routing protocols, must operate

efficiently regardless of the system size, which can have just dozens, hundreds, or even tens of thousands of nodes.

Lack of Physical Security In wired networks, it is often possible to physically secure the access to nodes (for example, by keeping nodes in rooms with limited controlled access). This is rare in wireless networks. For instance, military nodes in a hostile battlefield scenario cannot use security mechanisms that rely on their physical security, due to the risk of being captured and compromised. Thus, security mechanisms must be able to operate in face of compromised nodes.

Security Criteria Security in networks must address different issues such as: the confidentiality and integrity of information, legitimate use of the network, and availability of services [22]. These issues need also to be addressed in wireless ad hoc networks [12]. The fundamental concepts that will be used to discuss the network security aspects in the remainder of the text, will now be introduced:

- *Availability* is the ability of the network to continuously provide service, irrespective of attacks [10, 18] (including denial-of-service attacks, like radio jamming or battery exhaustion). In [23] *availability* is identified as one of the key attributes related to the security of networks;
- *Integrity* is the guarantee that a delivered message contains exactly the information that was originally sent [12, 18]. This guarantee precludes the possibility of messages being altered in transit. The causes of integrity violation may be accidental or malicious but, in practice, it is impossible to distinguish one from the other;
- *Authenticity* is the guarantee that participants in communication are genuine and not impersonators [12, 18]. To achieve authenticity, participants in the communication are required to prove their identities. Without this authentication, an attacker could impersonate a legitimate node, and obtain access to confidential resources or disturb the normal network operation by propagating fake messages.
  In some cases, it is possible to wave authenticity if end-to-end integrity is assured. For instance, in a wireless sensor network, if the messages arriving at the destination reflect the sensed environment, it does not matter if they were sent by legitimate nodes;
- *Confidentiality* is the guarantee that certain information is only readable by those who have been authorized to do so. This prevents information from being disclosed to unauthorized entities. If authentication is performed properly, confidentiality is a relatively simple process [12, 18];
- *Nonrepudiation* is the guarantee that the sender of a message cannot later deny having sent the information nor the receiver can deny having received it [12, 18]. This can be useful in the detection of compromised nodes, since it makes it possible to prove the malicious behavior of a specific node, by presenting any erroneous message it may have sent;

– *Self-healing* means that a protocol should be able to recover automatically from an erroneous state, in a finite amount of time, without human intervention. For instance, it should not be possible to permanently disable a network by injecting a small number of malicious packets at a given point in time. If a protocol is self-healing, an attacker must remain in the network and inflict continuous damage in order to prevent the protocol from recovering, a behavior that makes the attacker easier to locate [12];
– *Byzantine Robustness* means that a protocol should be able to function correctly, even if some of the nodes participating intentionally, attempt to disrupt its operation. Byzantine robustness can be seen as a stricter version of the self-healing property: the protocol must not only automatically recover from an attack; it should not cease from functioning, even if performance is hampered during the attack.

Attacks Taking into consideration the essential security criteria, the various kinds of possible attacks against ad hoc networks will now be discussed. The discussion will provide the basis for, later in the work, proposing defenses and countermeasures, against these attacks. The attacks can be classified into two broad classes, namely [12]:

– *External attacks* initiated from outside the network, in which the attacker attempts to cause congestion in the network, propagate incorrect routing information, prevent services from working properly, or shut down the network completely;
– *Internal attacks* initiated from within the network, in which the attacker gains normal access to the network by compromising directly or by impersonating an existing legitimate node. The attacker then uses the access to the network to engage in malicious behaviors.

In the two categories shown above, the external attacks are somewhat similar to typical attacks in wired networks, in which the attacker can exchange messages with network nodes but it is not a trusted node. These attacks can, therefore, be prevented and detected by conventional security methods such as membership authentication. On the other hand, internal attacks are far more dangerous, because the compromised nodes are originally legitimate nodes of the network and they can, therefore, pass the authentication and get protection from the security mechanisms [12].

Another way of classifying an attack can be done by the focus of the attack itself. Ad hoc networks are typically subjected to two different levels of attacks:

– *Passive Attacks* which consist on the attacker *eavesdropping* on the data that is being communicated in the network. Examples of passive attacks include covert channels, traffic analysis and sniffing information, allowing an attacker to compromise secrets and keys in the network;
– *Active Attacks* which involve specific actions performed by adversaries, for example, the modification, replication, or deletion of the exchanged data among network nodes.

External attacks are typically active attacks in which an adversary attempts to change the behavior of the operational mechanisms of the network. This is opposed to passive attacks in which the adversary will be subtle on his activities, while gathering information that may be later used to launch an active attack.

Denial-of-Service (DoS) can either be produced by an unintentional failure or by malicious action. The usual way to create a DoS attack is to *flood* a centralized resource with an abnormal number of requests, preventing it from operating correctly, or even cause it to crash. But, the fact that ad hoc networks do not have any centralized resource and distribute responsibilities throughout all the nodes in the network, makes them a difficult target to this kind of attack [23, 13]. However, by using a distributed denial-of-service attack, an attacker that has compromised enough nodes, can congest the network rather easily, rendering it useless. Better yet, a motivate resourceful attacker can completely deny the service to the nodes by using radio jamming, an attack which makes the communication links unusable at the physical level, or by using battery exhaustion [20], taking down the battery of power constrained nodes, one by one.

Eavesdropping The goal of eavesdropping is to obtain confidential information from messages exchanged among legitimate nodes. This attack is facilitated by the use of wireless links, since any node in the radio range of the participants in the communication can eavesdrop the link. To prevent eavesdropping, every critical data passing in the network, including control data, should be encrypted with strong cryptographic mechanisms.

Attacks on Information in Transit Any compromised or malicious node can utilize the information it forwards, for example, by executing routing protocols, to launch attacks. The attacker can maliciously intercept, modify, or fabricate routing messages that pass through him. These attacks can lead to the corruption of information, disclosure of sensitive information, theft of legitimate service from other protocol entities, or denial of network service to protocol entities [12]. Several attacks against routing protocols have been studied and are now well known [24–27].

Node Hijacking It is possible for a malicious node to masquerade as the base station and encourage users to connect to it. That node will be then in a privileged position to collect private data, such as: passwords, secret keys, logon names, etc. This is an example of a node hijacking where a legitimate base station has been hijacked by an attacker. There can be also other kind of node hijacking called "route hijacking", where the attacker modifies the routing information in order to hijack traffic to and from selected nodes [12].

Impersonation attacks pose a serious security risks on ad hoc networking. If the security mechanisms cannot support proper node authentication, compromised nodes may be able to impersonate trusted nodes. This kind of threat can be mitigated by the use of strong authentication mechanisms like digital signatures [10].

Due to the fact that digital signatures are implemented with public-key cryptography, they require high computational power and efficient and secure key management [12], something that most ad hoc network nodes are unable to provide due to the lack of resources. Due to this fact, hybrid encryption mechanisms like Message Authentication Codes (MAC) [28], can be used.

3.3 The Sybil Attack

A Sybil attack is essentially an impersonation attack, in which a malicious device illegitimately fabricates multiple identities, behaving as if it were a larger number of nodes (instead of just one) [9]. A malicious device additional identities are referred to as *Sybil identities* or *Sybil nodes*. According to the taxonomy presented in [29], there are three possible orthogonal dimensions for this attack: direct vs indirect communication; fabricated vs stolen identities; and simultaneity. In the worst case, an attacker can create an unlimited number of Sybil identities, with only one malicious device.

Direct vs. Indirect Communication

- *Direct communication* One way to perform the Sybil attack is for the Sybil nodes to communicate directly with the legitimate nodes. This means that, when a legitimate node sends a radio message to a Sybil node, the malicious device listens to the message. Symmetrically, when any of the sybil nodes sends a message, the messages are actually sent from the malicious node device;
- *Indirect communication* In this version of the attack, the legitimate nodes cannot directly communicate with the Sybil nodes. One or more malicious devices simply claim to be able to reach a number of Sybil nodes. This way, every message sent to a Sybil node is routed through one of these malicious node, which pretends to pass them to the final destination.

Fabricated vs. Stolen Identities There are essentially two different ways in which a Sybil node can get an identity: it can fabricate one (for instance, creating an arbitrary identifier) or it can *steal* an existing valid one from a legitimate node.

- *Fabricated Identities* If there is no network restriction to the allowed identities, or some way of verifying that an identity is legitimate, a malicious node can simply generate an arbitrary identity, and use it to join the network;
- *Stolen Identities* If there are mechanisms to prevent bogus identities from joining the network (for example, a limited namespace to prevent attackers from inserting new identities), the attacker may try to assign legitimate identities to the Sybil nodes. This identity theft may go unnoticed, if the attacker can, somehow, disable the impersonated nodes.

Simultaneity

- *Simultaneous* While a particular hardware entity can only advertise one identity at a time, it can cycle through these identities to make them appear to be present simultaneously. This way the attacker can have all his Sybil identities participating in the network at the same time;
- *Non-simultaneous* Alternately, the attacker can present a large number of identities over a period of time while only acting as a smaller number of identities at a given time. Also, if the attacker has several compromised nodes, he can make the nodes swap identities periodically, making detection even harder.

Attacks There are several known applications of Sybil attacks for wireless ad hoc networks [29, 30].

- *Routing* Sybil attacks have been shown to be effective against routing protocols in ad hoc networks [30]. One specially vulnerable mechanism is *multipath* or *dispersity* routing, where seemingly disjoint paths could, in fact, go through different Sybil identities of the same malicious node. Geographic routing is another vulnerable mechanism, where a Sybil node could appear in more that one place at once [31];
- *Data aggregation* Sensor networks make use of query protocols, which compute aggregates of values, obtained through sensor readings within the network, to conserve energy (rather than return each sensors individual reading)[29]. In scenarios with a small number of malicious nodes reporting erroneous sensor readings, the overall result may not be affected by a wide margin. Still, if the malicious sensors make use of the Sybil Attack, they can fabricate enough Sybil Nodes to alter significantly the outcome of the reading aggregation;
- *Voting* The Sybil attack could be used to alter the outcome of a voting scheme. If, for example, there is a voting scheme to determine node misbehavior in a network, an attacker can create enough false Sybil nodes to be able to expel any target node from the network. Conversely, if there is a vote on whether the attacker's identities are legitimate, the attacker could use his Sybil nodes to *vouch* for each other;
- *Misbehavior detection* If there is a mechanism in the ad hoc network to detect malicious behavior, an attacker launching a Sybil attack can escape detection by "spreading the blame" throughout all the Sybil nodes. If the mechanism requires several observations of this behavior to take action, by using different nodes, the attacker can escape detection completely. Even if, somehow, some Sybil Nodes are expelled from the network for malicious behavior, the attacker can always create more identities, and avoid being caught;
- *Fair resource allocation* Some network resources may be allocated on a per node basis. If, for example, the radio channel allocation is done by using time slots (TDMA MAC, for example), with the use of a Sybil Attack, the

attacker can gain access to more radio resources. This both denies service to legitimate nodes by reducing their share of the resource, and gives the attacker more resources to perform other attacks.

3.4 Countermeasures

As depicted before, the Sybil attack is a fundamental problem in many systems, for which no universally applicable solution has been devised. The most common solution relies on a central authority, in charge of ensuring that each node has a single identity, represented by one key. In practice, this is very difficult to achieve on large scale systems, and would require costly manual configuration, as well as limit the scalability of the whole system. Formal analysis of the Sybil attack have been done in the context of peer-to-peer applications [32, 9]. A number of approaches that can be used to protect from, or detect, this attack, as surveyed in [33], are summarized below.

Trusted Certification Trusted certification is the most common solution, mainly due to its potential to completely eliminate Sybil attacks [9]. However, trusted certification relies on a centralized authority, that must guarantee that each node is assigned exactly one identity, as indicated by possession of a certificate. In fact, Douceur [9] offers no method for ensuring such uniqueness, and in practice, it has to be performed by a manual configuration. This manual procedure can be costly, and create a performance bottleneck in large-scale systems. Additionally, and in order to be effective, the certifying authority must guarantee the existence of a mechanism to detect and revoke lost or stolen identities. These requirements make trusted certification very difficult to implement in ad hoc networks, which lack, by definition, a centralized authority that can provide the certification service.

While there are some solutions that reduce the network dependency on a centralized authority, for instance, requiring the presence of a certification authority only in the bootstrap of the network [34], there is still an additional problem with the use of a centralized authority: the possible existence of multiple administrative entities. If, there is only one common administrative entity managing the whole network, the implementation of a trusted certification, while having the problems stated above, can be a viable, if not perfect, solution. However, different administrative entities usually have different certification authorities. For example, consider an ad hoc network composed of nodes that do not belong to the same entity and, perhaps, have never met before. In this scenario, even if a node possesses a legitimate certificate from some certificate authority, it wouldn't be recognized as legitimate by any other node in the network, since they are not under the administration of that entity. This limits the environments in which the implementation of a trusted certification mechanism is possible.

Trusted Devices The use of trusted devices can be combined with trusted certification, binding one hardware device to one network entity. While this can effectively mitigate the Sybil attack, the main issue with this approach is that

there is no efficient way to prevent one entity from obtaining multiple hardware devices other than manual intervention [34].

Domain Specific There are some countermeasures that are application-domain specific. For example, in [35], a detection mechanism for ad hoc networks is proposed, based on the location of each node. For an attacker with a single device, all Sybil identities will always appear to move together. However, the defense is not applicable beyond mobile networks, and does not protect against malicious nodes with multiple devices.

Another possible way of thwarting the Sybil attack is auditing the correctness of identity behavior. If the audit is cheap, the Sybil attack has little benefit: for instance, a large number of seemingly independent nodes cannot successfully convince another node that they have factored a large number unless they have actually done so. For many peer-to-peer systems, including ad hoc networks, there has been a significant amount of work in using reputation systems as a possible solution for mitigating the damages caused by malicious peers. Using the classification described in [32], *symmetric* and *asymmetric* reputation systems can be distinguished; A symmetric reputation system is one in which an identity's reputation only depends on the topology of the trust graph, and not on the identity of the nodes. In an asymmetric reputation system, each entity computes a trust value along their unique paths to every other identity in the system. It is proven formally in [32] that symmetric reputation systems are susceptible to Sybil attacks. In regard to asymmetric systems, these can be effective in raising the cost of Sybil attacks, due to the fact that attackers have to gain trust before they can effectively launch attacks. Examples of asymmetric systems include the ones proposed by Feldman el al. [36], Guha et al. [37] and, Richardson et al. [38].

Resource Testing The main goal of resource testing is to attempt to determine if a number of identities possess fewer aggregated resources than would be expected if they were independent. In resource testing, it is assumed that each physical entity has a bounded amount of a given resource (e.g., limited bandwidth). The verifier then tests whether identities correspond to different physical entities by verifying that each identity has as much resources as an independent physical device should have. These tests include checks for computing power, storage ability and network bandwidth [9]. A type of resource test is employed by the SybilGuard technique [39], which relies on the limited availability of real-world *friendship* edges between nodes.

Recurring Costs and Fees There are several works in the literature that describe mechanisms in which identities are periodically re-validated using resource tests [40, 41]. This technique is a variation of the normal resource testing, and can limit the number of Sybil nodes an attacker, with constrained resources, can introduce in a period of time. Recently, it was shown [42], that charging a recurring fee for each participating identity is more effective as a disincentive against Sybil attacks. For many applications, recurring fees can incur a cost to

the Sybil attack that increases with the total number of identities participating; whereas one-time fees incur only a constant cost.

Radio Resource Testing In this context, radio resource testing, is a specific type of resource testing, which relies on the assumption that the device radios are incapable of simultaneously sending or receiving on two different frequencies. This idea has been used in [29], to counteract the sybil attack. However, the authors do not address the details that would allow them to build a protocol capable of operating in real world scenarios. Therefore, they do not present a comprehensive study on the cost and complexity of solutions based on this technique.

4 Architecture

This section starts by providing an informal description of the environment being targeted. Then, a more detailed model for this environment will be presented, capturing the assumptions made about the network architecture and radio resource limitations. Finally, a sketch of a technique to address the Sybil attack in multi-hop wireless networks will be provided.

4.1 Environment

This work will address the problem of defeating the Sybil attack in wireless mesh networks without a single administrative entity. The chosen environment has the following characteristics:

- The nodes are less resource-constrained than mobile devices. This fact will allow the use of asymmetric cryptography, increasing the universe of possible, and eventually more secure, solutions. Furthermore, the lack of node mobility in mesh networks creates the opportunity to focus on the problem at hand without having to deal with the additional complexities of highly dynamic environments;
- The fact that nodes do not share a single administrative entity excludes a solution where all legitimate nodes are pre-configured with a shared secret key. This poses a challenging problem of auto-configuration, which must be addressed.

This environment has been selected because it corresponds to a relevant scenario, with several practical and useful applications. Some situations where it would be interesting to deploy mesh networks without a single administrative entity are envisaged:

- In rescue operations, after major natural disasters, such as an earthquake or a tsunami, wireless mesh networks allow to quickly deploy a communication infrastructure. In such scenarios, first responders are often from different organizations or even different nationalities, and bring their own equipment. Therefore, auto-configuration mechanisms would be the right choice, since manual configuration is an expensive, prone to error, and lengthy procedure;

– In domestic applications, people living in the same residential area could cooperate to build a wireless mesh network to increase the coverage and bandwidth of their Internet access. This application is exemplified in Figure 2.
  In this example, there are a few key characteristics that make it hard to implement any solution requiring a shared secret, or pre-configuration of any sort. In the first place, not all users would have the technical skills to configure the equipment correctly; secondly, even between neighbors, it is hard to elect a single administrative identity that everybody trusts; thirdly, even if there was such an entity, there would be a need to physically configure each equipment added to the network, something which would constitute a severe logistic impairment. Mesh networks not requiring a single administrative entity are, thus, a natural choice.

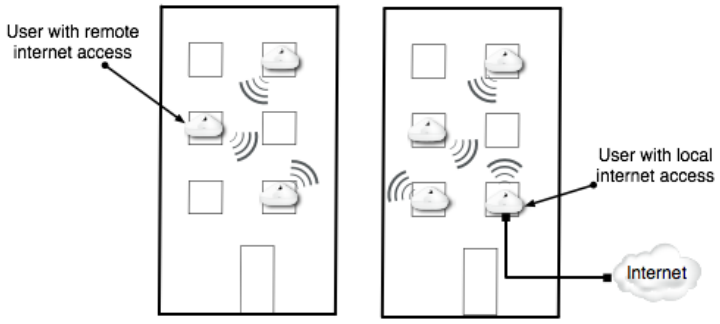


Fig. 2. Example of wireless mesh network for domestic applications.

4.2 The model

The mesh network will be modeled as a set of *identities* that communicate among each other via a shared wireless medium (simply ilustrated in Figure 3). The model is inspired in the one proposed by Douceur [9]. Each identity is assumed to be controlled by at least an *entity*. Physical resources are associated with entities.

The set E of entities is partitioned into two disjoint subsets, C and F. Each entity c in subset C is called a *correct* entity, and follows the rules of defined protocols. Similarly, each entity f in subset F is called a *faulty* entity, and may exhibit arbitrary (possibly Byzantine) behavior, limited only by explicit resource constraints (for example, computing power, network bandwidth and the number radio channels that an entity can simultaneously use). The maximum number of faulty entities in the system is s. There is no way to know, *a priori*, which

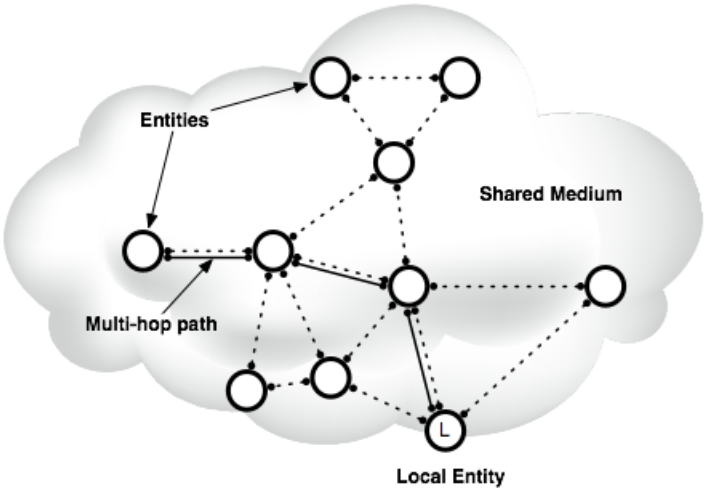


Fig. 3. Conceptual representation of a mobile ad hoc network.

entities are correct and which entities are faulty. In particular a faulty entity may comply to the protocol for an indefinite period of time, and thus behave like a correct entity, before it starts exhibiting malicious behavior.

Each correct entity c will associate itself with *a single* identity, that it will use to communicate with other entities. This identity is referred to as the entities's *legitimate* identity. Correct entities never communicate using identities other than their own legitimate identity. On the contrary, faulty entities may attempt to use multiple identities, including identities of correct entities, if the defined protocols allow such situation to happen. In this context, a Sybil attack consists of having a single faulty entity secretly assuming multiple identities, thus simulating the existence of false entities.

Entities are assumed to have the computational resources required to execute public-key cryptographic protocols (for example, establish private and authenticated virtual point-to-point communication paths among themselves). Using these protocols, it is possible to associate a public/private key pair to each identity. It should be noted that the use of public keys does not imply the existence of a PKI, because the keys do not need to be associated *a priori* with a specific entity.

The communication medium is a shared wireless multi-hop network. In this network, when an entity sends a message (using a given identity), this message is received by a set of other entities in physical range. These entities are said to be 1-hop neighbors. Entities that are not 1-hop neighbors cannot communicate directly. In order to exchange messages, these messages need to be forwarded by other entities in the network.

A message is essentially an uninterrupted finite-length bit string, with a common meaning for every entity (defined explicitly by the protocol, or implicitly by an agreement between the participating entities). When a message is received it is impossible to assess the identity of the sender using direct observation: the identity may only be known if included in the message. The wireless medium may omit messages (due to interference, collisions, etc), but does not alter or create messages (message corruption is detected and transformed into an omission error). The network also does not arbitrarily delay messages: when a message is transmitted, if it is delivered, it is delivered within a bounded interval of time.

The focus of this work will be on attacks that may lead a faulty entity to successfully use multiple identities. The case where a faulty entity simply aims at indefinitely postponing the protocol termination will not be addressed. The study of measures to defeat denial-of-service attacks is, therefore, outside the scope of this work.

4.3 The Approach

As previously described, sybil identities can be either fabricated, or stolen from existing entities. In ad hoc networks, it is difficult to avoid fabricated identities to join the network, but the use of stolen identities can be prevented. Since every node has asymmetric cryptography capabilities, every correct entity generates a public/private key pair, and uses the public key as the entity's identity. If a faulty entity wishes to assign an existing identity to one of the sybil nodes, it is required to have the corresponding private key, since it will have to answer a challenge while trying to join the network. This mechanism effectively mitigates the risk of a sybil attack with stolen identities.

Concerning the sybil attacks with fabricated identities, a technique that can be used to defeat the Sybil attack among 1-hop neighbors will be described, and then a strategy to address the problem in multi-hop networks will be sketched.

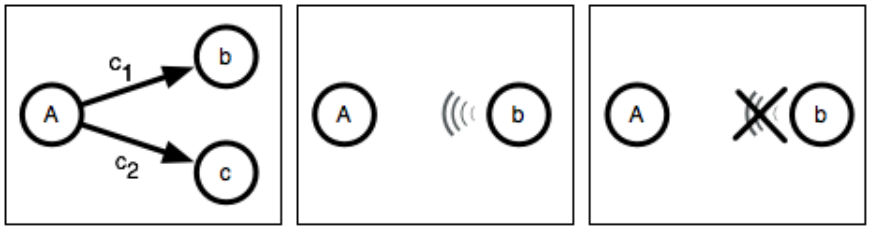
1-hop Sybil Identities To detect 1-hop sybil identities, a radio resource testing technique will be used. In order to apply this technique, it will be assumed that each entity only has access to a single radio device; this is a physical constraint that cannot be violated by faulty entities. Furthermore, it is assumed that radio devices can transmit and receive in different channels but that, at any given time, a device can only transmit or receive in a single channel[1].

Note that this assumption does not prevent physical nodes with multiple radio devices to be part of the network. However, these nodes will be treated as multiple entities, one for each radio device. Therefore, if a node has, for example, two radio devices, it will be modeled as two distinct entities and, consequently, it will be allowed to enter the network with a corresponding number of identities (two, in this case). The defense against sybil attacks can, therefore, be looked at as a scheme to ensure that all the identities currently participating in an ad

[1] This generic technique was previously addressed in [29].

hoc wireless network belong to different radio devices. If a node with multiple radios is faulty, its entities can be seen as colluding Byzantine entities.

The scheme to detect sybil identities uses the previous assumption that radio devices cannot simultaneously operate in more than one channel. This scheme is illustrated in Figure 4: Assume that entity A wants to verify that two identities, b and c, belong to two different entities, and are not, thus, sybil identities belonging to the same entity. To do so, it requests the owners of the identities to simultaneously transmit a message in two different channels, $c1$ and $c2$, respectively (Figure 5(a)). Then, entity A randomly chooses one of the two channels to listen to the message, and verifies if the owner of the identity allocated to that channel did transmit the message as requested[2]. The choice of the channel has to be random, to prevent an attacker from guessing in which channel entity A will listen to. This application of a zero-knowledge proof method[3] is necessary, since entity A cannot verify the transmission in both channels. If the radio devices could switch channels fast enough (frequency hopping), then an absolute test could be made by randomly switching channels during the verification transmission, thus effectively verifying the compliance of both identities. It will be assumed that this capability does not exist.



(a) Entity A requests a message to each identity

(b) Entity A listens on channel c_1, and hears the message

(c) Entity A listens on channel c_1 and does not hear the message

Fig. 4. Sybil identity detection scheme, based on radio resource testing.

For clarity sake, assume that the network is perfect (changes to the basic algorithm to deal with network omissions will also be addressed in the dissertation). Assume, also, that identity b was supposed to transmit in the channel that was selected to be listened to by A. One of two situations can happen: the owner of the identity allocated to that channel did transmit the message (Figure 5(b)),

[2] In order to simplify the examples, it will be assumed that only entity A is testing the identities, while in fact, every neighbor entity will simultaneously be testing that set identities.

[3] A zero-knowledge proof is essentially a method that allows one party to prove another that some statement is true, without revealing anything else than the veracity of the statement [43].

or it did not (Figure 4(c)). In the latter case, entity A is sure that identity b has been created by a faulty entity. In the former case, the test is inconclusive. Unfortunately, due to the inability of listening to both channels simultaneously, entity A cannot guess if a message was sent using identity c on the other channel. However, the test can be repeated enough times to exclude an attacker's lucky guess on which channel the entity A would listen in each round. Let r be the number of tests done by entity A on the same set of identities. Then, r should be large enough to guarantee that, with a chosen, pre-determined high probability, both identities belong to distinct entities. The probability of detecting a sybil identity is $P(detection) = 1 - P(nondetection) = 1 - (\frac{1}{2})^r$. This probability can be easily derived from the interaction model described above.

Note that, if two entities collude, they will be able to fool the test described above, since, if perfectly synchronized and coordinated, they can vouch for any pair of sybil identities they share. The way to deal with collusion is to simply test more than two identities at the same time. More precisely, an entity should test more identities than the number of faulty entities that may collude.

In general, if A wants to verify its neighborhood, where up to f possibly colluding faulty entities may exist, the test should be made with at least $f + 1$ simultaneous identities. Let c be the number of simultaneously tested identities, with $c > f$, and r the number of tests executed by entity A. If all c identities being tested belong to faulty entities, the probability of detection will be given by $P(detection) = 1 - P(nondetection) = 1 - (1 - \frac{1}{c})^r$.

Using this sort of proof, entity A can attest that c different identities correspond to c different entities, if the test is performed for all possible combinations. This eliminates any sybil identities in A 1-hop neighborhood.

Multi-Hop Extension The approach described above ensures that a correct entity only accepts a singe identity from each faulty entity within its 1-hop neighborhood. The problem now is how to guarantee the same, but with entities located two or more hops away. In this section, a sketch of a technique to address this problem is provided.

Consider that entity A desires to send a message to entity B, 2-hops away using identity a. Furthermore, entity B would like to assess that a is not a sybil identity. To solve this problem, it is required that a majority of correct entities exists within range of both A and B. That is: assuming that the objective is to tolerate f faulty entities, $2f + 1$ entities (r_1, r_2, ..., r_{2f+1}) must exist within 1-hop distance from both A and B.

Entity A starts by sending the message m, using identity a, to r_1, r_2, ..., r_{2f+1}. A correct entity with identity r_i forwards m if and only if it has, itself, tested the identity a against all other identities used in its 1-hop neighborhood. Thus, a correct entity never forwards messages from sibling identities created by a faulty entity in its 1-hop neighborhood.

Entity B only accepts (but does not immediately validate) a forwarded message m from an entity with identity r_i if B itself has tested the identity r_i against all other identities used in its own 1-hop neighborhood. This prevents a faulty

entity in the 1-hop neighborhood of B to forward the same message to B using different identities. Once B has accepted the forwarded message m from $f + 1$ 1-hop neighbors, it knows that at least one of these neighbors is correct and it has directly validated the source. It can thus validate m as a message that was sent by an entity that exists and is using a single identity.

This technique can be generalized for multiple hops, as long as there is a majority of correct entities involved in each hop, as illustrated in Figure 5.

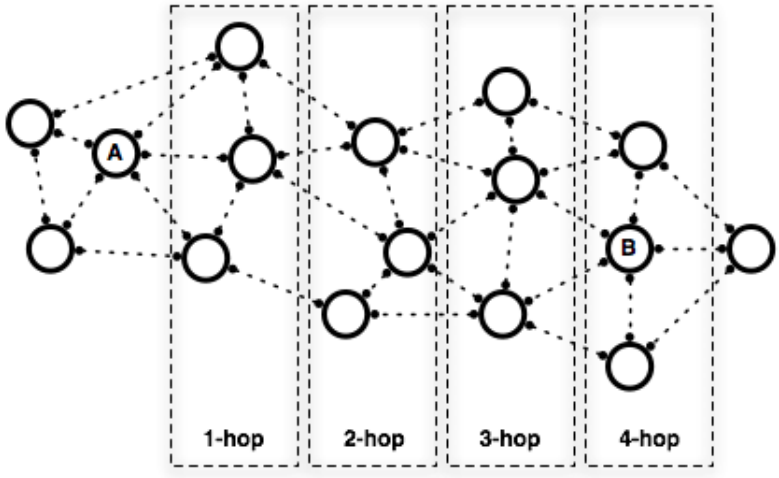
As presented, this approach to the multi-hop case has a potential vulnerability when the number of entities being simultaneously tested is higher than two[4]. The problem lies in the fact that, in this case, an attacker, while unable of assuming multiple identities from the viewpoint of a single correct entity, has, however, the possibility of assuming different identities when tested by different neighborhood entities. This creates the possibility of using this validation inconsistency to create subsets of the neighborhood, each one of which recognizes one different identity as valid for the faulty entity. This means that messages from the faulty entity will be forwarded by different sub-neighborhoods as if it had been originated by different valid entities. If the number of entities in these sub-neighborhoods is large enough, the destination node will accept these messages, and assume that they were originated by multiple valid entities. This would compromise the objective of avoiding the existence of sybil identities in the network, and is, thus, a vulnerability that needs to be addressed.

In the dissertation, all the details concerning the complete algorithm will be discussed, namely addressing the multi-hop scenario vulnerability (or concluding that there is no viable solution for it), the algorithm's impact on the network performance (which will depend on the node density), and the minimum required bounds for operation.

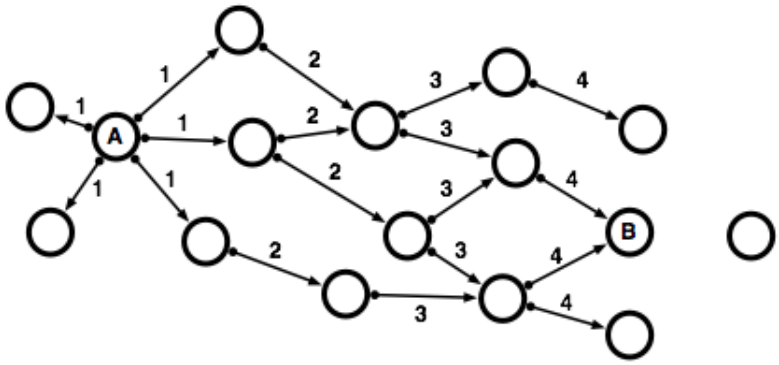
Improving performance Although the radio resource test appears to be a viable mechanism to thwart the sybil attack both in 1-hop networks and in multi-hop environments, one has yet to understand the associated cost of applying this strategy in real world scenarios. A few ideas that may enhance the performance of the proposed techniques will be explored.

For instance, and to illustrate the intuition: in a 1-hop network, the number of required tests to achieve a sybil-free network with a high probability p, is amenable to parameter tuning. In a network with n identities, if all of them are to be tested in pairs, the number of distinct tests is given by ${}_nC_2 = \binom{n}{2} = \frac{n!}{2(n-2)!}$. In itself, this creates an explosive increase in the number of tests, as shown in Figure 6. Additionally, for each pair, the test has to be repeated r times, to achieve the desired probability of detection $p = 1 - (\frac{1}{2})^r$. Figure 7 depicts the detection probability for $0 \leq r \leq 8$. It clearly shows that executing five tests

[4] This would only make sense due to the possibility of collusion between entities, as will be seen below, when addressing the performance issue.



(a) Example of a Multi-hop network, with majority of correct entities, for $f = 1$



(b) Representation of some of the messages propagated in the network, with the respective hop count

Fig. 5. Multi-hop example of a message being sent from Entity A to entity B, assuming $f = 1$

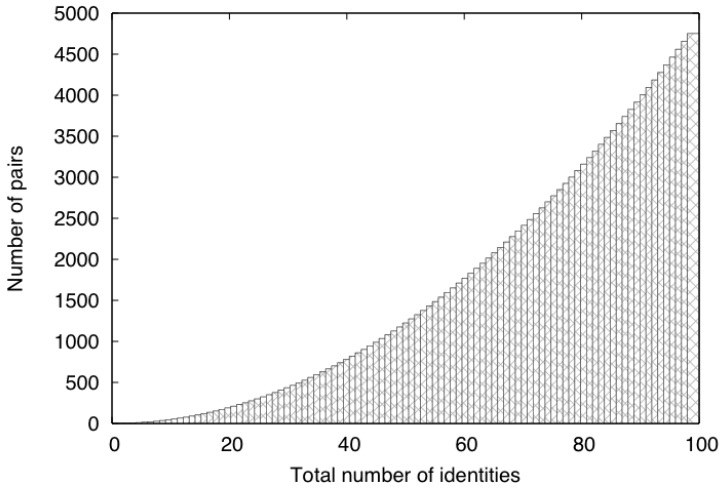


Fig. 6. Number of pairs, in a network with n identities.

to verify each pair of identities in the network is enough to provide a high probability (0.96875) of detecting sybil identities[5].

Considering these results, the total number of tests required in a network with n identities, using five tests per pair, is depicted in Figure 8.

Grouping more identities per test does not decrease the total number of required tests, both because: i) in each combination of identities, the number of tests necessary to achieve the desired probability of detection increases monotonically, and ii) because the number of combinations also increases. The combined effect of these two factors with the increase in the number of entities being simultaneously tested, is depicted in Figure 9, for a network of 10 nodes. In this figure, the number of tests needed for pairwise association is also represented with a dotted line, for easier comparison. However in the case where entities may collude, the problem becomes more complex and parameter tuning may enable performance improvements.

Also, in a multi-hop scenario, there is an expected trade-off between the size of the clusters of entities (1-hop neighborhoods) and the overall performance of the protocol. Artificially lowering the size of each local neighborhood allows to improve the parallelization of the tests, thereby improving the convergence time of the protocol. On the other hand, by decreasing the size of the 1-hop neighborhoods, one may increase the network diameter and therefore, the number of hops required for messages to transverse the network. This can degrade substantially

[5] We are assuming that a probability $p > 0.95$ of sybil identities detection is enough to provide a strong level of security to the network, however the protocol can be tuned to any desired threshold.

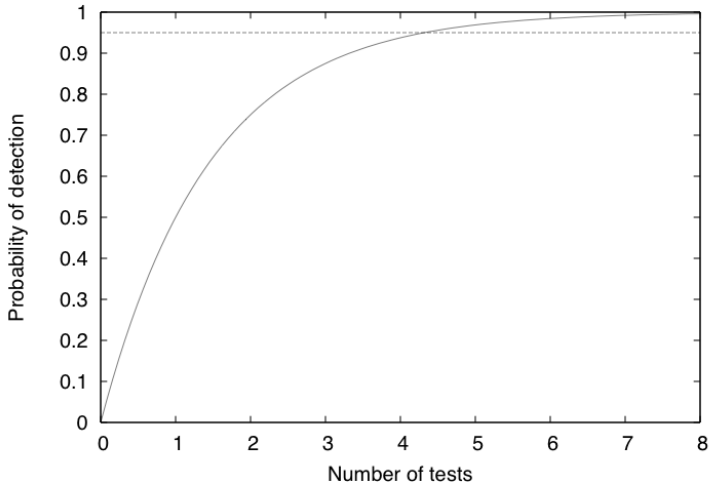


Fig. 7. Probability of detection of a sybil identity.

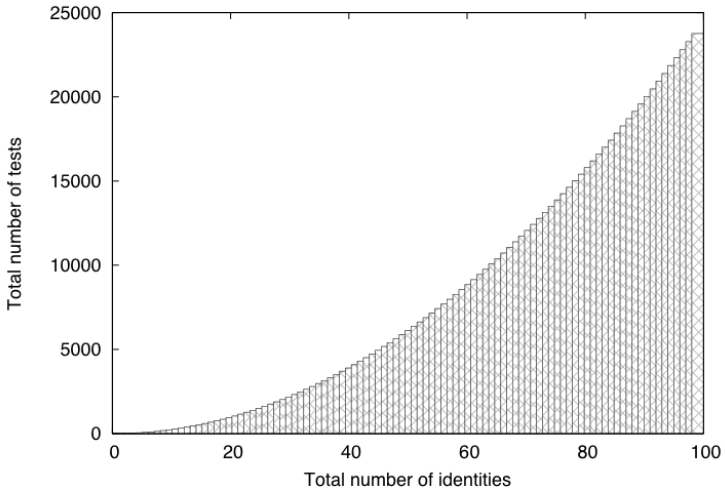


Fig. 8. Number of tests needed in a network with n identities, with 5 tests per pair.

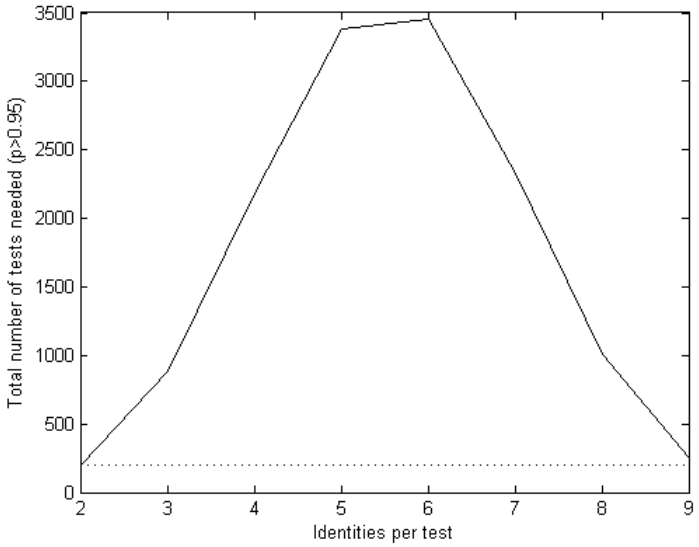


Fig. 9. Number of total tests required for detection probability $p > 0.95$, as a function of the number of entities being simultaneously tested.

the performance of the multi-hop approach. A good balance between the number of possible faulty entities the system is able to cope with, and the performance of the protocol, has yet to be found.

Other issues There are a number of other issues that must be addressed while designing a scheme to thwart the sybil attack. One of these issues is the required synchronization among entities in order to perform the radio resource testing. The radio resource test, which serves as the basis of the proposed approach, relies heavily on the fact that when an entity tests the simultaneous communication of two other entities, both of them are able to exactly determine when to begin transmitting the message. Lack of synchronization among participants may result in the increase of false positives, i.e correct entities that are incorrectly detected as being faulty.

Another challenge related with the proposed approach, is the medium access protocol. While contention-based medium access can prove to be the best solution for a small number of entities, as the size of the network increases, it may become more useful to employ an alternative medium access scheme which avoids collisions that would, otherwise, delay the convergence of the protocol. Therefore, a scheduling algorithm may be required. The lack of an efficient coordination scheme among entities may result in an inefficient usage of the communication medium, increasing the number of collisions, specially in the early phases of the

protocol. This, in turn, will degrade the performance of the system. To address this particular challenge, a scheduling algorithm that every entity can agree on, requiring almost no interaction among themselves, will be essayed. This can be done by having every entity locally sort alphabetically all the identities (since the identities are public keys, the collision probability is very low), and creating a common scheduling algorithm that would allow for all entities to do every test they need, without collisions happening. However such mechanism should also be resilient to faulty entities that may exploit this sub-protocol to generate schedules that facilitate their attacks.

5 Evaluation

The performance of the proposed approach will be obtained both analytically and via simulation. The performance of the protocols will be studied with a set of relevant metrics:

Metrics

- Required number of tests;
- Convergence time of the protocol;
- Probability of non-detection of a sybil identity;
- Maximum number of faulty entities tolerated.

Additionally, an effort will be made to understand how the protocols are affected by a set of environment associated parameters. Namely, the impact of the total number of nodes in the network, and the desired resilience of the protocol to faulty nodes will be studied.

In the specific case of multi-hop scenarios, the problem of neighborhood sizes will be analyzed, in an attempt to derive mathematical relations capable of describing the relation between the neighborhood size, the required number of overall tests, and the level of resilience to faulty entities. These equations will hopefully be amenable to an analysis of extreme values, which may help to develop an optimized scheme to determine the protocol parameters.

6 Schedule

The work is scheduled as follows:

Oct 6 - Nov 12	Preliminary analysis: problem identification, definition of scope of work, feasibility and risk analysis.
15 Dec - Jan 9, 2009	Extended summary/work plan document production.
Jan 15 -	Project evaluation.
Jan 16 - Apr 15	Detailed design of the proposed architecture, including preliminary tests.
Apr 15 - May 3	Final evaluation of performance results.
May 4 - May 23	Article production.
May 24 - Jun 15	Finish the writing of the dissertation.
Jun 15	Deliver the MSc dissertation.

7 Conclusion

This work addresses the design of a mechanism to tackle the Sybil Attack. This attack is a major issue in ad hoc networks, specially for protocols that require the use of quorums as a method of agreement. Existing countermeasures for this attack are surveyed, with the conclusion that the methods based on pre-shared secrets, needing pre-configuration are not adequate in several real-world environments. For 1-hop networks, a technique is proposed, based on the assumption that each entity has a single radio device and cannot communicate simultaneously on more than one channel at a time. This technique does not require any kind of pre-shared secret or central authority, and is, thus, aligned with the auto-configuration objective. The multi-hop case is also addressed, and requires a different technique. Some directions to the algorithm's performance tuning are pointed out, and are left as aspects to be fully analyzed in the dissertation.

Acknowledgments

I am grateful to J. Leitão, J. Mocito and my advisors Prof. Luís Rodrigues and Prof. Carlos Ribeiro, for the fruitful discussions and expert guidance during the preparation of this work.

References

1. Gupta, P., Kumar, P.: The capacity of wireless networks. Information Theory, IEEE Transactions on **46**(2) (2000) 388–404
2. Li, J., Blake, C., De Couto, D.S.J., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: Proceedings of the 7th ACM International Conference on Mobile Computing and Networking, Rome, Italy (July 2001) 61–69
3. Malkhi, D., Reiter, M.: Byzantine quorum systems. Distributed Computing **11** (1998) 569–578
4. Xu, T., Wu, J.: Quorum based ip address autoconfiguration in mobile ad hoc networks. Distributed Computing Systems Workshops, International Conference on **0** (2007) 1
5. Haas, Z., Liang, B.: Ad hoc mobility management with uniform quorum systems. Networking, IEEE/ACM Transactions on **7**(2) (Apr 1999) 228–240
6. Jiang, J.R., Tseng, Y.C., Hsu, C.S., Lai, T.H.: Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks. Mobile Networks and Applications **10**(1) (February 2005) 169–181
7. Haas, Z., Liang, B.: Ad hoc mobility management with uniform quorum systems. Networking, IEEE/ACM Transactions on **7**(2) (1999) 228–240
8. Luo, J., pierre Hubaux, J., Eugster, P.T.: Pan: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In: In Proc. of MobiHoc. (2003) 1–12
9. Douceur, J.R., Donath, J.S.: The sybil attack. In: Proceedings for the 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA (March 2002) 251–260
10. Vesa, K.: Security in ad hoc networks (2000)

11. Hubaux, J.P., Buttyán, L., Capkun, S.: The quest for security in mobile ad hoc networks. In: MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, New York, NY, USA, ACM (2001) 146–155
12. Mishra, A.: Security and Quality of Service in Ad Hoc Wireless Networks. Cambridge University Press (2008)
13. Hubaux, J.P., Le Boudec, J.Y., Giordano, S., Hamdi, M., Blazevic, L., Buttyan, L., Vojnovic, M.: Towards mobile ad-hoc wans: terminodes. Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE **3** (2000) 1052–1059 vol.3
14. Stajano, F., Anderson, R.: The resurrecting duckling: security issues for ubiquitous computing. Computer **35**(4) (Apr 2002) 22–26
15. Zhang, Y., Lee, W.: Intrusion detection in wireless ad-hoc networks. In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, New York, NY, USA, ACM (2000) 275–283
16. Nadkarni, K., Mishra, A.: A novel intrusion detection approach for wireless ad hoc networks. Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE **2** (March 2004) 831–836 Vol.2
17. Verissimo, P., Rodrigues, L.: Distributed Systems for System Architects. Kluwer Academic Publishers, Norwell, MA, USA (2001)
18. Ilyas, M., Dorf, R.C., eds.: The handbook of ad hoc wireless networks. CRC Press, Inc., Boca Raton, FL, USA (2003)
19. Mohapatra, Prasant; Krishnamurthy, S.: Ad Hoc Networks Technologies and Protocols. Springer (2004)
20. D, A., A, J.: Data networks. In: Upper Saddle River, Prentice –Hall, Inc (1992)
21. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4) (2002) 393–422
22. Ford, W.: Computer communications security: principles, standard protocols and techniques. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1994)
23. Zhou, L., Haas, Z.: Securing ad hoc networks. Network, IEEE **13**(6) (Nov/Dec 1999) 24–30
24. Papadimitratos, P., Haas, Z.J.: Secure routing for mobile ad hoc networks. In: in SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002. (2002) 27–31
25. chun Hu, Y.: Abstract ariadne: A secure on-demand routing protocol for ad hoc networks (2002)
26. Sanzgiri, K., Levine, B.N., Shields, C., Dahill, B.: A secure routing protocol for ad hoc networks. (2002)
27. chun Hu, Y., Johnson, D.B., Perrig, A.: Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. (2002) 3–13
28. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication, Springer-Verlag (1996) 1–15
29. Newsome, J., Shi, E., Song, D., Perrig, A.: The sybil attack in sensor networks: analysis & defenses. In: Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on. (2004) 259–268
30. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: In First IEEE International Workshop on Sensor Network Protocols and Applications. (2003) 113–127
31. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Networks **1**(2-3) (2003) 293 – 315 Sensor Network Protocols and Applications.

32. Cheng, A., Friedman, E.: Sybilproof reputation mechanisms. In: P2PECON '05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, New York, NY, USA, ACM (2005) 128–132
33. Levine, B.N., Shields, C., Margolin, N.B.: A Survey of Solutions to the Sybil Attack. Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA (October 2006)
34. Martucci, L.A., Kohlweiss, M., Andersson, C., Panchenko, A.: Self-certified sybil-free pseudonyms. In: WiSec '08: Proceedings of the first ACM conference on Wireless network security, New York, NY, USA, ACM (2008) 154–159
35. Piro, C., Shields, C., Levine, B.N.: Detecting the sybil attack in mobile ad hoc networks. Securecomm and Workshops, 2006 (28 2006-Sept. 1 2006) 1–11
36. Feldman, M., Lai, K., Stoica, I., Chuang, J.: Robust incentive techniques for peer-to-peer networks. In: EC '04: Proceedings of the 5th ACM conference on Electronic commerce, New York, NY, USA, ACM (2004) 102–111
37. Guha, R., Raghavan, P.: Propagation of trust and distrust. In: In WWW, ACM Press (2004) 403–412
38. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: In Proceedings of the Second International Semantic Web Conference. (2003) 351–368
39. Yu, H., Kaminsky, M., Gibbons, P., Flaxman, A.: Sybilguard: Defending against sybil attacks via social networks. Networking, IEEE/ACM Transactions on **16**(3) (June 2008) 576–589
40. Maniatis, P., Rosenthal, D.S.H., Roussopoulos, M., Baker, M., Giuli, T., Muliadi, Y.: Preserving peer replicas by rate-limited sampled voting. SIGOPS Oper. Syst. Rev. **37**(5) (2003) 44–59
41. Maniatis, P., Roussopoulos, M., Giuli, T.J., Rosenthal, D.S.H., Baker, M.: The lockss peer-to-peer digital preservation system. ACM Trans. Comput. Syst. **23**(1) (2005) 2–50
42. Margolin, N.B., Levine, B.N.: Quantifying resistance to the sybil attack. In: Proc. Financial Cryptography (FC). (January 2008)
43. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. Journal of Cryptology **7**(1) (1994) 1–32