The Sybil Attack in Sensor Networks: Analysis & Defenses*

James Newsome
Carnegie Mellon University
jnewsome@ece.cmu.edu

Elaine Shi
Carnegie Mellon University
rshi@cmu.edu

Dawn Song
Carnegie Mellon University
dawnsong@cmu.edu

Adrian Perrig
Carnegie Mellon University
adrian@cmu.edu

ABSTRACT

Security is important for many sensor network applications. A particularly harmful attack against sensor and ad hoc networks is known as the *Sybil attack* [6], where a node illegitimately claims multiple identities. This paper systematically analyzes the threat posed by the Sybil attack to wireless sensor networks. We demonstrate that the attack can be exceedingly detrimental to many important functions of the sensor network such as routing, resource allocation, misbehavior detection, etc. We establish a classification of different types of the Sybil attack, which enables us to better understand the threats posed by each type, and better design countermeasures against each type. We then propose several novel techniques to defend against the Sybil attack, and analyze their effectiveness quantitatively.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Security

Keywords

Sybil Attack, Sensor Networks, Security

*This research was supported in part by the Center for Computer and Communications Security at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office, and by gifts from Bosch, Cisco, Intel, and Matsushita Electric Works Ltd. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, Bosch, Carnegie Mellon University, Cisco, Intel, Matsushita Electric Works Ltd., or the U.S. Government or any of its agencies.

IPSN'04, April 26–27, 2004, Berkeley, California, USA.

1. INTRODUCTION

Sensor networks are a promising new technology to enable economically viable solutions to a variety of applications, for example pollution sensing, structural integrity monitoring, and traffic monitoring. A large subset of sensor network applications requires security, especially if the sensor network protects or monitors critical infrastructures.

Security in sensor networks is complicated by the broadcast nature of the wireless communication and the lack of tamper-resistant hardware (to keep per-node costs low). In addition, sensor nodes have limited storage and computational resources, rendering public key cryptography impractical.

In this paper, we investigate the Sybil attack, a particularly harmful attack in sensor networks. In the Sybil attack, a malicious node behaves as if it were a larger number of nodes, for example by impersonating other nodes or simply by claiming false identities. In the worst case, an attacker may generate an arbitrary number of additional node identities, using only one physical device.

Related Work The Sybil attack was first described by Douceur in the context of peer-to-peer networks [6]. He pointed out that it could defeat the redundancy mechanisms of distributed storage systems. Karlof and Wagner noted that the Sybil attack also poses a threat to routing mechanisms in sensor networks [9].

Contributions This is the first paper that systematically analyzes the Sybil attack and its defenses in sensor networks. This paper makes the following contributions. We introduce a taxonomy of the different forms of the Sybil attack as it applies to wireless sensor networks. We analyze how an attacker can use the different types of the Sybil attack to perturb or compromise several sensor network protocols. We propose several new defenses against the Sybil attack, including radio resource testing, key validation for random key predistribution, position verification, and registration. Through quantitative analysis, we show that the radio resource testing method is very effective given the assumption that a malicious node cannot send on multiple channels simultaneously. We also present a quantitative evaluation for the random key predistribution approach showing that it is robust to compromised nodes. In particular, we show that in the multi-space pairwise scheme storing 200 keys at each node, the attacker would have to compromise 400 nodes before having even a 5% chance of being able to fabricate new identities for the Sybil attack.

2. SYBIL ATTACK TAXONOMY

We define the Sybil attack as a malicious device illegitimately taking on multiple identities. We refer to a malicious device's additional identities as *Sybil nodes*. To better understand the implications of the Sybil attack and how to defend against it, we develop a taxonomy of its different forms. We propose three orthogonal dimensions: direct vs indirect communication, fabricated vs stolen identities, and simultaneity.

2.1 Dimension I: Direct vs. Indirect Communication

Direct Communication One way to perform the Sybil attack is for the Sybil nodes to communicate directly with legitimate nodes. When a legitimate node sends a radio message to a Sybil node, one of the malicious devices listens to the message. Likewise, messages sent from Sybil nodes are actually sent from one of the malicious devices.

Indirect Communication In this version of the attack, no legitimate nodes are able to communicate directly with the Sybil nodes. Instead, one or more of the malicious devices claims to be able to reach the Sybil nodes. Messages sent to a Sybil node are routed through one of these malicious nodes, which pretends to pass on the message to a Sybil node.

2.2 Dimension II: Fabricated vs. Stolen Identities

A Sybil node can get an identity in one of two ways. It can fabricate a new identity, or it can *steal* an identity from a legitimate node.

Fabricated Identities In some cases, the attacker can simply create arbitrary new Sybil identities. For instance, if each node is identified by a 32-bit integer, the attacker can simply assign each Sybil node a random 32-bit value.

Stolen Identities Given a mechanism to identify legitimate node identities, an attacker cannot fabricate new identities. For example, suppose the name space is intentionally limited to prevent attackers from inserting new identities. In this case, the attacker needs to assign other legitimate identities to Sybil nodes. This identity theft may go undetected if the attacker destroys or temporarily disables the impersonated nodes.

A related issue is *identity replication*, in which the same identity is used many times and exists in multiple places in the network. The identity replication attack can be performed and defended against independently of the Sybil attack[1]. We do not have space to fully address it, but we believe it is relatively simple to defend against by registering each identity's location. Identities could be registered at a central location, or using a distributed hash table such as GHT [17]. This approach would detect that the same identity exists in multiple locations. Another approach, when using the pairwise-random key approach by Chan et al., is to centrally count the number of connections a node has, and revoke nodes with too many connections, thus countering node replication [5].

[1]For example, an attacker could capture a legitimate node, and "clone" that node on many instances of his own hardware. This would not be the Sybil attack, because each piece of hardware still has just one identity.

2.3 Dimension III: Simultaneity

Simultaneous The attacker may try to have his Sybil identities all participate in the network at once. While a particular hardware entity can only act as one identity at a time, it can cycle through these identities to make it appear that they are all present simultaneously.

Non-Simultaneous Alternately, the attacker might present a large number of identities over a period of time, while only acting as a smaller number of identities at any given time. The attacker can do this by having one identity seem to leave the network, and have another identity join in its place. A particular identity might leave and join multiple times, or the attacker might only use each identity once.

Another possibility is that the attacker could have several physical devices in the network, and could have these devices swap identities. While the number of identities the attacker uses is equal to the number of physical devices, each device presents different identities at different times.

3. ATTACKS

In this section, we examine how the Sybil attack can be used to attack several types of protocols in wireless sensor networks. We first consider attacks on distributed storage algorithms, similar to the ones Douceur [6] describes in the peer-to-peer environment. We then look at attacks on routing algorithms, which Karlof and Wagner discuss [9]. We then look at novel attacks on data aggregation, voting, fair resource allocation, and misbehavior detection algorithms. Table 1 summarizes which of these attacks can be performed by which forms of the Sybil attack.

3.1 Known Attacks

Distributed Storage Douceur observes that the Sybil attack can defeat replication and fragmentation mechanisms in peer-to-peer storage systems [6]. The same problem exists for distributed storage in wireless sensor networks. For instance, the Sybil attack could just as easily defeat replication and fragmentation performed in a distributed hash table such as GHT [17]. While the system may be designed to replicate or fragment data across several nodes, it could actually be storing data on Sybil identities generated by the same malicious node.

Routing Karlof and Wagner point out that the Sybil attack can be used against routing algorithms in sensor networks [9]. One vulnerable mechanism is *multipath* or *dispersity* routing where seemingly disjoint paths could in fact go through a single malicious node presenting several Sybil identities. Another vulnerable mechanism is geographic routing [4, 10, 11] where instead of having one set of coordinates, a Sybil node could appear in more than one place at once. In addition, the more general types of attacks that we shall soon describe may also be used to attack routing algorithms. For instance, while the network may attempt to detect routing attacks such as black holes, in Section 3.2 we show that

	Communication		Identities		Simultaneity	
	Direct	Indirect	Fabricated	Stolen	Simultaneous	Non-Simultaneous
Distributed Storage	√	√	√	√	√	
Routing	√		√	√	√	
Data Aggregation	√	√	√	√	√	
Voting	√	√	√	√	√	√*
Resource Allocation	√	√	√	√	√	√**
Misbehavior Detection	√	√	√	√	√	√

Table 1: Protocols that are vulnerable to the various forms of the Sybil attack. * Votes taking place over a period of time are potentially vulnerable to the non-simultaneous Sybil attack. ** This form of attack is possible if nodes are allowed to join in smaller time intervals than the resource is allocated. e.g. if nodes are allowed to flood the network only once per hour, but new nodes can join the network once per minute.

an attacker could use the Sybil attack to evade such misbehavior detection mechanisms.

3.2 New Attacks

Data Aggregation Efficient query protocols [13] compute aggregates of sensor readings within the network in order to conserve energy rather than returning individual sensor readings. A small number of malicious nodes reporting incorrect sensor readings might be unable to significantly affect the computed aggregate. However, by using the Sybil attack, one malicious node may be able to contribute to the aggregate many times. With enough Sybil nodes, an attacker may be able to completely alter the aggregate reading.

Voting Wireless sensor networks could use voting for a number of tasks. The Sybil attack could be used to "stuff the ballot box" in any such vote. Depending on the number of identities the attacker owns, he may be able to determine the outcome of any vote. For example, this could be used to perform *blackmail* attacks, in which the attacker claims that a legitimate node is misbehaving. 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.

Fair Resource Allocation Some network resources may be allocated on a per node basis. For example, nearby nodes sharing a single radio channel might each be assigned a fraction of time per interval during which they are permitted to transmit. The Sybil attack can be used to allow a malicious node to obtain an unfair share of any resource shared in this manner. This both denies service to legitimate nodes by reducing their share of the resource, and gives the attacker more resources to perform other attacks.

Misbehavior Detection Suppose that the network can potentially detect a particular type of misbehavior. It is likely that any such misbehavior detector has some false positives. As a result, it might not take action until it observes several repeated offenses by the same node. An attacker with many Sybil nodes could "spread the blame", by not having any one Sybil identity misbehave enough for the system to take action. Additionally, if the action taken is to revoke the offending node, the attacker can simply continue using new Sybil identities to misbehave, never getting revoked himself.

4. DEFENSES

To defend against the Sybil attack, we would like to *validate* that each node identity is the only identity presented by the corresponding physical node. There are two types of ways to validate an identity. The first type is *direct validation*, in which a node directly tests whether another node identity is valid. The second type is *indirect validation*, in which nodes that have already been verified are allowed to vouch for or refute other nodes. With the exception of the key pool defense, the mechanisms that we present here are for direct validation. We leave secure methods of indirect validation as future work.

Previous Defenses Douceur proposes resource testing as a method of direct validation. In resource testing, it is assumed that each physical entity is limited in some resource. The verifier tests whether identities correspond to different physical entities by verifying that each identity has as much of the tested resource as a physical device. The resources proposed by Douceur to use for this purpose are computation, storage, and communication. Computation and storage are unsuitable for wireless sensor networks, because the attacker may be using a physical device with several orders of magnitude more computation and storage ability than a resource starved sensor node. The proposed method of testing communication is to broadcast a request for identities and then only accept replies that occur within a given time interval. This method is also unsuitable for wireless sensor networks because all the replies converging at the verifier will result in that part of the network becoming congested.

New Defenses In the remainder of this section, we propose several new defenses against the Sybil attack in sensor networks, including radio resource testing, verification of key sets for random key predistribution, registration and position verification. In addition, we give quantitative analysis for the first two defenses.

4.1 Radio Resource Testing

We present a novel approach to direct validation. As a form of resource testing, this approach relies on the assumption that any physical device has only one radio. We also assume that a radio is incapable of simultaneously sending or receiving on more than one channel

As a concrete example, consider that a node wants to verify that none of its neighbors are Sybil identities. It can assign each of its n neighbors a different channel to broadcast some message on. It can then choose a channel randomly

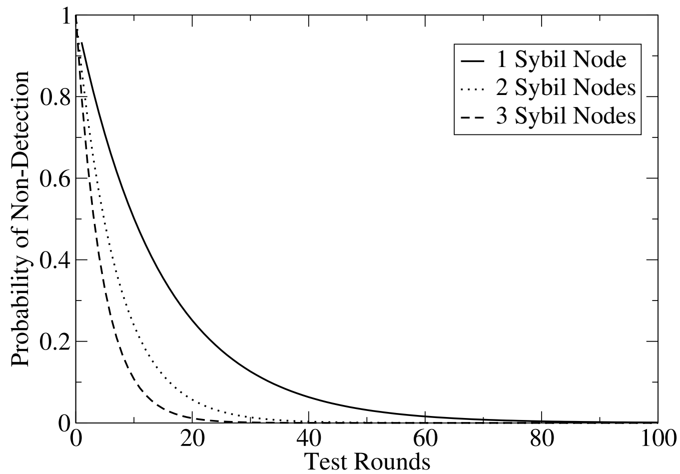

Figure 1: Probability of no Sybil nodes being detected, using the radio defense, with a channel for every neighbor. Assumes 15 neighbors (including Sybil nodes), any number of which could be malicious.

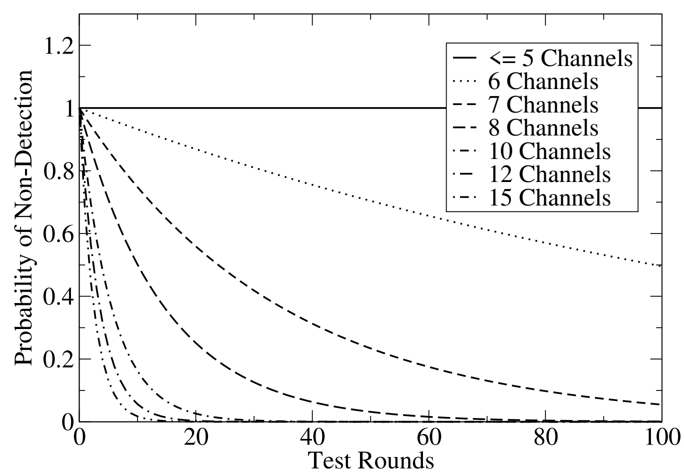

Figure 2: Probability of no Sybil nodes being detected, using the radio defense, with fewer channels than neighbors. Assumes 5 correct neighbors, 5 malicious neighbors, and 5 Sybil neighbors.

on which to listen. If the neighbor that was assigned that channel is legitimate, it should hear the message.[2] Suppose that s of the verifier's n neighbors are actually Sybil nodes. In that case, the probability of choosing to listen to a channel that is not being transmitted on, and thus detecting a Sybil node, is $\frac{s}{n}$. Conversely, the probability of *not* detecting a Sybil node is $\frac{n-s}{n}$. If the test is repeated for r rounds, then the chance of no Sybil nodes being detected is $(\frac{n-s}{n})^r$. Figure 1 shows the probability of not detecting the presence of some Sybil nodes using this method.

A more difficult case is when there are not enough channels to assign each neighbor a different channel. In this case, a node can only test some subset of its neighbors at one time. If there are c channels, then the node can test c neighbors at once. Note that a malicious node not in the subset being tested can *cover for* a Sybil node that is being tested by transmitting on the channel that the Sybil node is supposed to be transmitting on.

Suppose that in a node's set of n neighbors, there are s Sybil nodes, m malicious nodes, and g good (correct) nodes. Of these, a node can only test c neighbors at one time. Of these c neighbors, there are S Sybil nodes, M malicious nodes, and G good (correct) nodes. The probability of a Sybil node being detected is then

$$\begin{aligned} Pr(detection) &= \sum_{all S,M,G} Pr(S,M,G)Pr(detection|S,M,G) \\ &= \sum_{all S,M,G} \frac{\binom{s}{S}\binom{m}{M}\binom{g}{G}}{\binom{n}{c}} \frac{S-(m-M)}{c} \end{aligned}$$

Now suppose that we repeat this test for r rounds, choosing a random subset to test and a random channel to listen to in each round. The probability of a Sybil node being detected is then

$$\begin{aligned} Pr(detection) =& 1 - Pr(nondetection)^r_{1round} \\ =& 1 - (1 - Pr(detection)_{1round})^r \\ =& 1 - \left(1 - \sum_{all S,M,G} \frac{\binom{s}{S}\binom{m}{M}\binom{g}{G}}{\binom{n}{c}} \frac{S-(m-M)}{c}\right)^r \end{aligned}$$

Figure 2 shows the probability of an attacker evading detection when using 5 malicious nodes, and generating 5 additional Sybil identities. This is an effective defense against the simultaneous direct-communication variant of the Sybil attack, if the assumptions hold that an attacker cannot use one device to send on multiple channels simultaneously. However, with the advancement of software radio, we will need to adapt this Sybil node detection technique.

4.2 Random Key Predistribution

Researchers recently proposed a promising technique for key distribution in sensor networks: *random key predistribution* [5, 7, 8, 12]. These techniques allow nodes to establish secure links to other nodes. In this section, we will show how these key distribution schemes can also be used to defend against the Sybil attack.

In random key predistribution, we assign a random set of keys or key-related information to each sensor node, so that in the key set-up phase, each node can discover or compute the common keys it shares with its neighbors; the common keys will be used as a shared secret session key to ensure node-to-node secrecy.

Our key ideas are:

1. Associating the node identity with the keys assigned to the node.

2. Key validation, i.e., the network being able to verify part or all of the keys that an identity claims to have.

Consequently given a limited set of captured keys, there is little probability that an arbitrarily generated identity is going to work, for the keys associated with a random

[2] Note that in order to avoid potential constraints imposed by the MAC layer, such as collision detection and avoidance mechanisms, we must have direct access to the physical layer while performing this test.

identity are not likely to have a significant intersection with the compromised key set, making it hard for the fabricated identity to pass the key validation.

Again, for key validation, we have indirect and direct validation. In the case of direct validation, each node challenges an identity using the limited knowledge it possesses and makes a decision independent of other nodes. Thus nodes may not reach a globally consistent decision. With indirect validation, nodes could collaborate in validating a node, thus it is possible to reach a globally consistent decision. Of course we may also delegate the validation task to a central trusted party such as a base station. Generally speaking, indirect key validation is much more costly in terms of communication overhead than the direct case, because in the former case, if node ID_i tries to validate ID_j, messages only need to be exchanged between ID_i and ID_j; while in the latter, it will also involve exchanging messages between other parties. Also indirect validation, if done improperly, could become the victim of blackmail attacks. However, indirect validation usually provides stronger defense against the Sybil attack, for, due to the memory constraint of sensor nodes, each individual node has limited knowledge that it could use to pose a challenge to an identity.

Different variants of existing random key predistribution techniques include the basic key pool approach [5, 8], the single-space pairwise key distribution approaches [2, 3, 5], and the multi-space pairwise key distribution approaches [7, 12].

So far, researchers have studied these techniques in the context of establishing secret keys between neighboring nodes. However, we shall study them for the purpose of defending against the Sybil attack. We propose an extension to the basic key pool approach to allow it to defend against the Sybil attack. We analyze and compare the effectiveness of several key predistribution schemes in defending against the Sybil attack.

Key Pool Previous research on the key pool scheme focuses on the aspect of key predistribution. We now modify the existing key pool scheme so that it may be used to defend against the Sybil attack; then we evaluate its effectiveness.

The key pool scheme randomly assigns k keys to each node from a pool of m keys. During the initialization phase, if any two neighboring nodes discover that they share q common keys, they can establish a secret link.

To use this scheme to defend against the Sybil attack, suppose that each node's identity is the indices in sorted order of the keys that it holds. The problem with this approach is that if an attacker compromises multiple nodes, he can use every combination of the compromised keys to generate new identities. Let $\Omega(ID) = \{K_{\beta_1}, K_{\beta_2}, \ldots, K_{\beta_k}\}$ be the set of keys assigned to ID, where ID is the identity of node, β_i is the index of its i^{th} key in the key pool. Now suppose that the set of the keys that node ID possesses are determined by $\beta_i = PRF_{H(ID)}(i)$, where H is a hash function, and PRF is a pseudo random function. This means that the index of a node's ith key is determined by a pseudo random function with $H(ID)$ as the function's key, and i as its input. Similar methods of choosing keys have been proposed before as an optimization [15]; We will show that this method helps to defend against the Sybil attack. Here we exploit the property of the PRF function that given its outputs over domain $1 \ldots k$, it is difficult to find a cryptographic key such that the instance of PRF specified by this key will yield exactly these outputs. The one-wayness of the hash function gives us an additional security guarantee in case the PRF should be broken. I.e., if the attacker happens to find a key to the PRF that yields the wanted outputs, he still has a hard time in finding the pre-image of the key, i.e., the Sybil identity.

An attacker may attempt to generate new identities to use in the Sybil attack. To do this, he will need to capture legitimate nodes and read off the keys, thus building up a compromised key pool S. He will then attempt to fabricate usable Sybil identities. If a made-up identity ID' can participate in the sensor network without being detected in the key initialization phase, we call it a *usable Sybil identity*.

A usable Sybil identity must be able to pass the validation by other nodes. To validate an identity, the verifier challenges the identity by requesting it to prove that it possesses one or more keys it claims to have. If $\exists K_i, K_i \in \Omega(ID'), K_i \notin S$, and if some legitimate entity E in the sensor network knows K_i, then E can discover that ID' is cheating by challenging ID' using K_i. To achieve a globally consistent outcome, it is necessary to perform indirect validation, which is the case we shall discuss.

Validation could be done at different granularities. One extreme is the case of full validation where the sensor network tries to verify as many of a node's keys as possible, rendering a Sybil attack more difficult. In practice, however, full validation requires that every node challenge every other node in the network, which could result in excessive communication overhead and the potential of DOS attacks. To avoid these drawbacks we could limit the scope of validation. For instance, we could limit the validation process within the vicinity of the node being validated, such as randomly selecting d nodes out of its k-hop neighborhood to jointly perform the validation. The larger d and k are, an attacker will be less likely to succeed; on the other hand, the validation will be more expensive.

We shall now evaluate the time complexity for an attacker to generate a usable Sybil node ID given a set of compromised nodes. We consider an attacker that performs exhaustive search, i.e., computing $PRF_{H(ID)}(j)(1 \leq j \leq k)$ for each candidate identifier, until he finds one that results in the desired set of key indices. We may provide a rigorous cryptographic proof that this is the best the attacker can do in the full version of this paper. The time complexity of the attacker could be expressed in terms of the probability p that a random identity is a usable Sybil identity. So the expected number of times an adversary has to try to find a usable Sybil identity is $\frac{1}{p}$. In our analysis, we shall compare the security levels of different granularities of validation.

We use the following notation:
ID': a randomly generated identity;
Ψ: key pool;
m: size of key pool, $m = |\Psi|$;
k: size of key ring;
n: size of compromised key pool.

First, consider the full validation case where each identity is challenged by all other nodes in the entire network. Assume that the sensor network is large enough, so that it is possible to verify every key the identity claims to have. Thus to survive the full validation, ID' has to satisfy $\Omega(ID') \subseteq S$.

Therefore

$$Pr(ID' \ is \ a \ usable \ Sybil \ \ ID) = Pr(\Omega(ID') \subseteq S) = \frac{\binom{n}{k}}{\binom{m}{k}}$$

Now consider the case where each identity is challenged by d nodes. To calculate the probability that ID' is a usable Sybil ID, we condition over t, the number of keys in $\Omega(ID')$ that are also in S, i.e., $t = card(\Omega(ID') \cap S)$, where $card(A)$ denotes the cardinality of the set A.

We use the following notation:

$V(ID', ID_0)$: ID' passes validation with a particular verifier ID_0;

$V(ID')$: ID' passes validation with all d verifiers.

Thus

$$Pr(card(\Omega(ID') \cap S) = t) = \frac{\binom{n}{t}\binom{m-n}{k-t}}{\binom{m}{k}}$$

Given $card(\Omega(ID') \cap S) = t$, ID' can survive the validation of a particular verifier ID_0, if and only if $\Omega(ID_0) \cap (\Omega(ID') - S) = \phi$. Put another way, the keys of ID_0 must only be selected out of $\Psi - (\Omega(ID') - S)$, whose cardinality is $m - k + t$. Then

$$\begin{aligned} & Pr(V(ID') \mid card(\Omega(ID') \cap S) = t) \\ =& Pr(V(ID', ID_0) \mid card(\Omega(ID') \cap S) = t)^d \\ =& \left(\frac{\binom{m-k+t}{k}}{\binom{m}{k}}\right)^d \end{aligned}$$

Now we can calculate the probability that a randomly generated Sybil ID is usable:

$$\begin{aligned} & Pr(ID' \ is \ a \ usable \ Sybil \ \ identity) \\ =& \sum_{t=1}^{k} \Big[Pr(card(\Omega(ID') \cap S) = t) \\ & \qquad \cdot Pr(V(ID') \mid card(\Omega(ID') \cap S) = t) \Big] \\ =& \sum_{t=1}^{k} \left[\frac{\binom{n}{t}\binom{m-n}{k-t}}{\binom{m}{k}} \left(\frac{\binom{m-k+t}{k}}{\binom{m}{k}}\right)^d \right] \end{aligned}$$

Now, if we know that the attacker compromised c random nodes, the expected number of compromised keys is $m \cdot (1 - (1 - \frac{k}{m})^c)$. Using this as an estimate of n it is possible to roughly compute the probability that ID' is a usable Sybil identity given c compromised nodes.

Figure 3 plots the estimated probability that a randomly generated Sybil identity is usable against the number of nodes compromised. The three curves represent full validation, partial validation with $d = 50$, and partial validation with $d = 30$. We can see that if the tolerance threshold is

$Pr(a \ random \ Sybil \ \ ID \ is \ usable) = 2^{-64}$,

for the full validation case, the attacker cannot succeed unless he is able to compromise at least 150 nodes, whereas for partial validation with $d = 30$, he only needs to compromise approximately 30 nodes.

Single-space Pairwise Key Distribution

In the random key pool distribution scheme, keys can be issued multiple times out of the key pool, and node-to-node authentication is not possible [5]. Meanwhile, if an attacker succeeds in capturing a sufficient number of nodes, it could compromise a sufficient fraction of keys so that the task of generating a usable Sybil identity will become trivial. By contrast, pairwise key distribution assigns a unique key to each pair of nodes. Single space pairwise key distribution approaches are represented by Blom's scheme [2] and the polynomial-based scheme [3]. Also Chan, Perrig and Song proposed the random pairwise key distribution scheme [5].

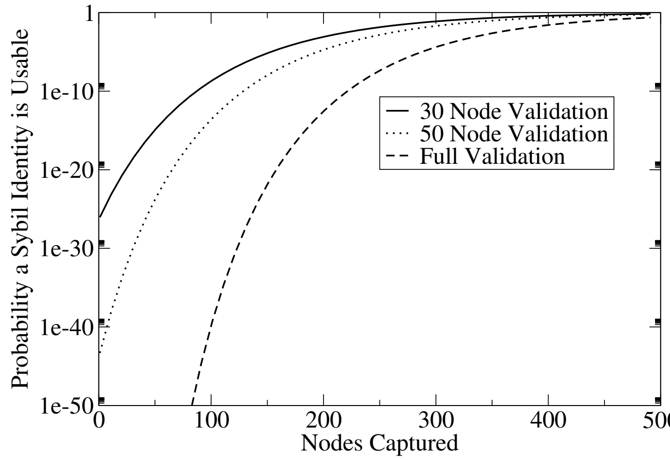

Figure 3: Probability a randomly generated Sybil node is usable in the key pool scheme. The probability p is a direct measure of the time complexity for an adversary to generate a usable Sybil identity, i.e., he expects to try 1/p times to fabricate one usable Sybil identity. Key pool size $m = 20000$, key ring size $k = 200$, memory constraint $l = k = 200$.

Both Blom's and the polynomial scheme require a sensor node i to store unique public information U_i and private information V_i. During the bootstrapping phase, nodes exchange public information, and node i could compute its key with node j with $f(V_i, U_j)$. It is guaranteed that $f(V_i, U_j) = f(V_j, U_i)$. Both approaches ensure the λ-secure property: the coalition of no more than λ compromised sensor nodes reveals nothing about the pairwise key between any two non-compromised nodes. Therefore, given c compromised nodes, if $c \leq \lambda$, a simple direct validation mechanism suffices to prevent both the direct and indirect versions of the Sybil attack. A node validates an identity provided that the identity has the pairwise key between the two nodes. Meanwhile, it is ensured that direct validation yields a globally consistent result. This comes from the fact that the adversary either compromises the entire space such that he could compute the pairwise key between any two identities, or he will know nothing about the key between a new Sybil identity and any other node. On the other hand if $c > \lambda$ it can create an arbitrary number of new identities, and the network will be prone to the Sybil attack.

The random pairwise key distribution scheme proposed by Chan et al. ensures perfect resilience against node capture, i.e., any number of captured nodes reveal no information about the pairwise keys between legitimate nodes. Therefore, an adversary cannot fabricate new identities given any number of captured nodes. The price of this, however, is that the network size will be strictly restricted by each node's memory constraint l, and the probability that any two nodes are connected p. For instance, if $l = \lambda + 1 = 200$

keys, and $p = 0.33$, then the maximum supportable network size is $l/p \approx 600$.

Multi-space Pairwise Key Distribution

Recently, researchers have proposed the idea of multiple key spaces [7, 12] to further enhance the security of single-space approaches. The idea of introducing multiple key spaces can be viewed as the combination of the basic key pool scheme and the above single space approaches. The setup server randomly generates a pool of m key spaces each of which has unique private information. Each sensor node will be assigned k out of the m key spaces. If two neighboring nodes have one or more key spaces in common, they can compute their pairwise secret key using the corresponding single space scheme.

In preventing the Sybil attack, the multi-space scheme exhibits the following properties:

1. Without validation: Given a number of captured nodes, if at least one key space is compromised, the node could make up an arbitrary number of new identities that could directly communicate with the rest of the network. If none of the key spaces are compromised, it is virtually impossible for the adversary to make up any new usable identities to launch a direct-communication Sybil attack. However, the network is still prone to the indirect-communication variant of Sybil attack if no validation scheme is present.

2. With validation: If an adversary claims to have key space T_i which it has not compromised, then a node ID_0 could challenge the adversary if ID_0 has T_i. To do this, ID_0 simply has to verify whether the adversary has the pairwise key of T_i between the two nodes. Similar to the key pool scheme, here indirect validation is necessary to ensure a globally consistent outcome, for it is not guaranteed that any node could successfully challenge an identity given the limited number of spaces it owns. If we could perform full validation, the adversary at least has to compromise k key spaces to fabricate an identity that could pass validation.

We now try to evaluate the probability that at least k spaces are compromised given c compromised nodes. This is a direct measure of the difficulty of a Sybil attack when a validation mechanism is present. Let S_i be the event that space i is compromised.

In previous work [7], it has been proved that given c compromised nodes,

$$Pr(S_i) = \sum_{j=\lambda+1}^{c} \binom{c}{j} (\frac{k}{m})^j (1 - \frac{k}{m})^{c-j}$$

Now if we know that S_1 is true, i.e., space 1 is compromised, then it is less likely that S_2 is compromised. In fact, $\forall i \neq j, Pr(S_i|S_j) \leq Pr(S_i)$. Thus, $Pr(S_1 \cap S_2 \cap \ldots \cap S_k) \leq Pr(S_1) \cdot Pr(S_2) \cdot \ldots \cdot Pr(S_k)$. Now we could derive an upper bound for the probability that at least k spaces are compromised given c compromised nodes.

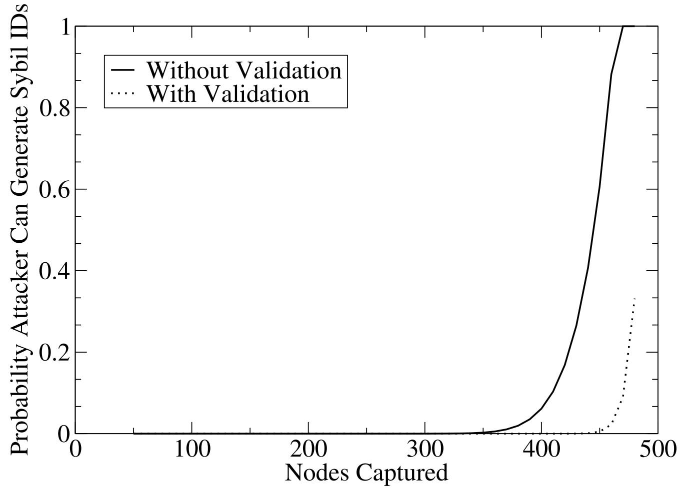

Figure 4: Probability that an attacker can fabricate Sybil identities with the multispace scheme. Number of spaces in pool $m = 50$, number of spaces per node $k = 4$, the λ value as in λ-secure $\lambda = 49$, memory constraint $l = 200$

$$\begin{aligned} & Pr(at\ least\ k\ spaces\ compromised) \\ & \leq \binom{m}{k} \cdot Pr(S_1 \cap S_2 \cap \ldots \cap S_k) \\ & \leq \binom{m}{k} \cdot Pr(S_1) \cdot Pr(S_2) \cdot \ldots \cdot Pr(S_k) \\ & = \binom{m}{k} \cdot Pr(S_1)^k \end{aligned}$$

Figure 4 plots the probability of successfully performing a Sybil attack against the number of nodes captured. The two curves represent the cases with and without validation respectively. In contrast to Figure 3, here the probability is not of a randomly generated identity being usable. Instead it is the probability that the attacker will have the necessary information to succeed given a set of compromised nodes, regardless of the computational effort expended. The figure shows that the attacker has to compromise approximately 400 nodes in the case without validation, and 465 nodes with validation, to successfully perform a Sybil attack with probability $\geq$0.05.

Summary of Random Key Predistribution In this section, we analyzed several random key predistribution techniques in the context of Sybil attack defense.

For the basic key pool approach, by mapping a node's identity to the indices of its keys using a one-way function, and through means of indirect validation, a randomly generated identity has only probability p of being usable. An adversary has to try $\frac{1}{p}$ times on average to obtain a usable Sybil identity, thus for the sensor network to be immune to the Sybil attack, p has to be very small.

Single-space pairwise key distribution, such as Blom's approach and the polynomial-based approach, is intrinsically resistant to the Sybil attack as long as the attacker does not capture more than λ nodes. Here, direct validation ensures a globally consistent validation outcome. However, once the attacker succeeds in capturing more than λ nodes, the entire space is compromised and he can fabricate an arbitrary number of identities.

Multi-space pairwise key distribution is superior to the single-space case in that the attacker has to compromise far more than λ nodes to compromise one space, for each node is randomly assigned k out of m spaces, and he has to capture more than λ instances of each space to compromise it. Besides, he has to compromise at least k spaces to pass full validation, which is even more difficult. To compare it with the key pool approach, we assume the nodes have equal memory constraint. From Figure 3 and Figure 4 we can see that when the attacker compromises approximately 400 nodes, he has a high probability of successfully forging usable Sybil identities in the key pool scheme; whereas in the multi-space pairwise scheme, the attacker will succeed only with a probability of around 0.05 even in the case without validation. We therefore believe the multi-space pairwise approach to be the best among these approaches.

4.3 Other Defenses

Registration One obvious way to prevent the Sybil attack is to perform identity registration. A difference between peer-to-peer networks and wireless sensor networks is that in wireless sensor networks, there may be a trusted central authority managing the network, and thus knowing deployed nodes. The central authority may also be able to disseminate that information securely to the network. To detect Sybil attacks, an entity could poll the network and compare the results to the known deployment. To prevent the Sybil attack, any node could check the list of "known-good" identities to validate another node as legitimate.

Registration is likely to be a good initial defense in many scenarios, with the following drawbacks. The list of known identities must be protected from being maliciously modified. If the attacker is able to add identities to this list, he will be able to add Sybil nodes to the network. Additionally, the deployment information that is checked against must be accurately and securely maintained by the entity that owns and/or manages the sensor network.

Position Verification Another promising approach to defending against the Sybil attack is position verification. Here we assume that the sensor network is immobile once deployed. In this approach, the network verifies the physical position of each node. Sybil nodes can be detected using this approach because they will appear to be at exactly the same position as the malicious node that generates them. While there has been research on automatic location determination [1, 16], it remains an open research question how to securely *verify* a node's exact position. Such a method may be difficult to find, but researchers have proposed methods to securely verify that a node is *within a region* [18]. By placing a limit on the density of the network, in-region verification can be used to tightly bound the number of Sybil identities that a malicious node can create.

Note that a mobile attacker may be able to present several identities by being verified as one identity at one location, and then moving to a different location and being verified as a different identity. To defeat this type of attack, all nodes' positions could be verified simultaneously. Alternatively, given an upper bound on the attacker's mobility, it would only be necessary to test the nodes within a certain range simultaneously.

Code Attestation Remote code verification or attestation is another promising new technique that could be employed to defend against many types of attacks, including the Sybil attack. The basic idea is to exploit the fact that the code running on a malicious node must be different from that on a legitimate node. Therefore, we could validate a node by verifying its memory content. Researchers have already started investigating this idea. Recently, Seshadri et al. proposed SWAtt [19], a new technique to securely verify the code running on a remote embedded device. Though this technique is not readily applicable to a wireless network environment, hopefully in the near future code verification will become possible in wireless sensor networks, helping solve many problems including the Sybil attack.

Future computing devices may be equipped with trusted hardware that provides strong security guarantees, such as a component developed by the Trusted Computing Group (TCG) [20] (formerly known as TCPA), or the Next-Generation Secure Computing Base (NGSCB) [14] (formerly known as Palladium) developed by Microsoft. Both TCG and NGSCB provide an attestation mechanism, which enables an external device to get integrity guarantees about the application state. Through a challenge-response protocol, another device can achieve assurance of the code running on a device. However, the high cost and energy consumption of trusted hardware devices precludes using them in current sensor devices. Dropping costs and increasing efficiency, however, make trusted hardware a promising technique to secure future sensor networks.

5. DISCUSSION

Each of the defenses against the Sybil attack that we have examined has different tradeoffs. As Table 2 shows, most defenses are not capable of defending against every type of Sybil attack.

Additionally, each defense has different costs and relies on different assumptions. The radio resource verification defense may be breakable with custom radio hardware, and validation may be expensive in terms of energy. Position verification can only put a bound on the number of Sybil nodes an attacker can generate unless it is able to very precisely verify node positions. Node registration requires human work in order to securely add nodes to the network, and requires a way to securely maintain and query the current known topology information.

We believe that of the defenses that we have presented, random key predistribution is the most promising. Using random key predistribution will already be desirable in many applications to secure radio communication. We have shown that it can also be used as an effective measure to prevent the Sybil attack with little or no additional cost.

We believe that an important next step in this area will be to examine secure methods of *indirect* validation that do not rely on a trusted central authority. These would allow methods of direct validation that cannot easily be performed by a single device, such as the radio resource defense, to be used for indirect validation. This is a challenging problem, largely

Defense	Who Can Validate	Remaining Sybil Vulnerabilities
Radio	Neighbors	Indirect Com., Non-Simult.
Position Verification	Neighbors	Indirect Com.*
Registration	Anyone	Stolen IDs
Key Predistribution	Anyone w/shared keys	Stolen IDs**
Code Attestation	Anyone	None***

Table 2: Comparison of Sybil defenses. * This is assuming that nodes can only verify the position that they communicate directly with. Future work could find a mechanism without this limitation. ** While the key predistribution defenses will not stop an attacker from using stolen identities, it does make it more difficult for the attacker to steal identities in the first place. An attacker must first compromise a node's key ring before he can steal its identity. * It is not yet known exactly how code attestation may work in wireless sensor networks. If and when it does work, it will be impossible to perform the Sybil attack while attesting correctly without defeating the attestation mechanism. One danger is that an attacker restores the correct state of a node to attest correctly, and then recompromises it.**

because malicious nodes must be prevented from vouching for each-other and from blackmailing legitimate nodes.

6. CONCLUSION

In this paper, we define the Sybil attack and establish a taxonomy of this attack by distinguishing different attack types. The definition and taxonomy are very important in understanding and analyzing the threat and defenses of a Sybil attack.

We present several novel methods by which a node can verify whether other identities are Sybil identities, including radio resource testing, key validation for random key predistribution, position verification and registration. The most promising method among these is the random key predistribution which associates a node's keys with its identity. Random key predistribution will be used in many scenarios for secure communication, and because it relies on well understood cryptographic principles it is easier to analyze than other methods. These methods are robust to compromised nodes. In particular, we have shown that in the multi-space pairwise scheme with each node storing 200 keys, the attacker would need to compromise 400 nodes before having even a 5% chance of being able to fabricate new identities for the Sybil attack.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. We would also like to thank Haowen Chan and Arvind Seshadri for helpful discussions and suggestions; and Rohit Negi, Ben Henty, and Adam Pennington for their feedback on the radio defense mechanism.

7. REFERENCES

[1] P. Bahl and V. Padmanabhan. Radar: an in-building RF-based user location and tracking system. In *Proceedings of IEEE Infocom*, 2000.

[2] R. Blom. Non-public key distribution. In *Advances in Cryptology: Proceedings of Crypto '82*, pages 231–236, 1982.

[3] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *Advances in Cryptology - Crypto '92*, pages 471–486, 1992.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[5] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, May 2003.

[6] J. R. Douceur. The Sybil attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Mar. 2002.

[7] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *ACM CCS 2003*, pages 42–51, Oct. 2003.

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

[9] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, May 2003.

[10] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[11] Y.-B. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 66–75. ACM, Oct. 1998.

[12] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *ACM CCS 2003*, pages 52–61, Oct. 2003.

[13] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad hoc sensor networks. In *Symposium on Operating Systems Design and Implementation*, Nov. 2002.

[14] Next-Generation Secure Computing Base (NGSCB). `http://www.microsoft.com/resources/ngscb/default.mspx`, 2003.

[15] R. D. Pietro, L. V. Mancini, and A. Mei. Random key assignment for secure wireless sensor networks. In

ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of ACM MobiCom*, 2000.

[17] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: a geographic hash table for data-centric storage. In *WSNA 2002*, Sept.

[18] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *Proceedings of the ACM Workshop on Wireless Security (WiSe 2003)*, September 2003.

[19] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWAtt: Software-based attestation for embedded devices. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[20] Trusted Computing Group (TCG). `https://www.trustedcomputinggroup.org/`, 2003.