

# Security Analysis on Defenses against Sybil Attacks in Wireless Sensor Networks

Karen Hsu  
karenhsu@berkeley.edu  
UC Berkeley

Man-Kit Leung  
jleung@berkeley.edu  
UC Berkeley

Brian Su  
briansu@berkeley.edu  
UC Berkeley

## ABSTRACT

Few security mechanisms in wireless sensor networks (WSNs) have been implemented, and even fewer have been applied in real deployments. The limited resources of each sensor node makes security in WSNs hard, as the tradeoff between security and practicality must be carefully considered. While there are many types of security attacks in WSNs, we have decided to focus our analysis on a particularly harmful one: the Sybil attack. We apply two security metrics (resiliency and connectivity) and three practicality metrics (processing, storage, and communication complexity) to each of the major proposed defenses against the Sybil attack that we are aware of for analysis and evaluation. We conclude that while such metrics are a good indicator of the security and practicality tradeoffs, they alone are not enough to determine a defense's usability. They must be used in addition to other qualitative evaluations as well. We believe that additional research and experimentation with new metrics is necessary for further insight.

## 1. INTRODUCTION

Wireless sensor networks (WSNs) are networks comprised of low-cost, autonomous devices that can be used to gather environmental or motion data such as temperature, sound, vibration, pressure, and pollutants[6].

There are many applications of WSN in which security is a high priority. However, unlike many other wireless networks, WSNs are typically equipped with slower processors, less memory, a shorter battery life, and limited radio transmission ranges. As a result, these resource constraints present additional challenges to any security mechanism for WSN.

One particularly dangerous attack against sensor networks is the Sybil Attack. A Sybil attack succeeds when a malicious node, called the Sybil node, illegitimately claims multiple false identities by either fabricating new identities or impersonating existing ones[15]. The goal of a Sybil attack is to gain a disproportionate amount of influence over the net-

work via its false identities. We argue that Sybil attacks are especially harmful as they are often the gateway to other attacks (such as those on resource exhaustion, voting, etc.)

In this paper, we apply a set of metrics to evaluate and analyze each of the major proposed Sybil attack defenses. Because practicality is so important for usable security, particularly in resource constrained WSNs, we hope to identify useful yet realistic defense heuristics that are worthy of being further researched.

## 2. RELATED WORK

The Sybil attack was first introduced by Douceur as an attack on peer-to-peer(P2P) systems [8], although it is also found in wireless sensor networks. Roosta et al. presented a taxonomy of security attacks and countermeasures in sensor networks and mentioned a variety of defenses against the Sybil attack. However, as it was not their intention, they did not provide details on any of the available defenses[16]. Detailed analysis on one particular Sybil defense, key-predistribution, can be found in work done by Camtepe and Yener [4]. In their evaluation, they use several metrics that we borrow for our analysis.

Newsome et al. presented a security analysis of defenses against Sybil attacks in sensor networks. While their work is most similar to ours, they do not employ consistent metrics in their defense evaluations. Additionally, their analysis concentrates more on the probability of the success of a Sybil attack, given a particular probabilistic defense. The focus of our paper will be more on the *damage* resulting from a Sybil attack (which is a function of resiliency and connectivity) as well as the *practicality* of each scheme. We believe that, especially in a WSN setting, practicality is crucial due to the limited resources of each node. Many security mechanisms have been proposed for WSN, though currently most real, industry deployments employ only a single shared key for the entire WSN as their only security defense. We hope that our work can help bridge the gap between proposed theoretical models and usable security mechanisms for real deployments.

## 3. THREAT MODEL AND ASSUMPTIONS

In our analysis on the various Sybil attack defenses, we construct a threat model and make several relevant assumptions. Aside from the obvious resource limitations of the sensor nodes, we limit the scope of our study to non-hierarchical WSNs. In other words, each node is equally equipped with

the same computational power, storage size, battery life, radio, etc. Furthermore, these WSNs are static and have relatively uniform and dense topologies.

Like any typical network attackers, our adversary can eavesdrop and inject arbitrary packets into the network. The attacker has access to laptop-class devices and are, therefore, not constrained in computational power, storage, or any hardware resources. The attacker is allowed to be selective (smart), meaning that he/she can always pick out the minimal set of nodes to compromise in order to subvert the whole network if such a vulnerability exists. This gives the attacker the edge when breaking a defense algorithm that relies on probabilistic events.

Furthermore, we assume that the WSN is deployed in a hostile environment; any individual node is vulnerable to physical capture by the attacker. Nodes are non-tamper resistant and so once compromised, the attacker will gain control of its stored secrets. As our paper analyzes the resulting damage of the Sybil attack per defense, the attacker is free to replicate any compromised node to gain access to communication within the network. We believed that making realistic assumptions and using a non-conservative threat model will make our research more practical for real deployments.

#### 4. ANALYSIS METRICS

Let  $N$  represent the number of nodes in the WSN. We borrow and modify from [4]’s evaluation metrics:

We employ two security metrics:

- **resiliency:** The number of nodes an attacker needs to compromise in order to compromise the entire WSN. If resiliency has the  $\lambda$ -secure property, the coalition of no more than  $\lambda$  compromised sensor nodes reveals nothing about the pairwise key between any two uncompromised nodes.[2]
- **connectivity:** The probability that a node can directly communicate with another node in the WSN.

and three practicality metrics:

- **storage complexity:** The amount of *extra* storage required for added security mechanism
- **processing complexity:** The number of *extra* unit functions that need to be performed resulting from the security mechanism. These unit functions may include search, Hash, MAC, etc.
- **communication complexity:** The number of *extra* messages sent as well as any increase in size of these messages. This can be written in the format  $a|b$ , where  $a$  is the number of extra messages and  $b$  is the increase in size per message.

### 5. DEFENSE ANALYSIS

#### 5.1 Code attestation

Code attestation validates the code stored in memory by either hardware or software approaches. Clearly, the code

running on a legitimate node will differ from the code running on a malicious node. While this is a promising new approach, few trusted hardware or software systems have been designed for sensor networks. [15]

Seshadri et al. have proposed SCUBA (Secure Code Update By Attestation) for detecting and recovering compromised nodes in sensor networks. They assume a public key infrastructure is set up for authentication between each node and the base station (Here, the base station acts as a trusted authority). Each node stores its node ID and the base station’s public key in ROM. Its node ID is used as input to the ICE (indisputable code execution) verification function which generates a checksum. This is a strong defense against impersonation Sybil attacks, as the attacker is forced to forge the node ID (which can be done by a data substitution attack - a conditional check must be inserted to divert the read to ROM to another place in memory where the attacker stores the node ID of the node it wishes to impersonate). The authors argue that such a data substitution attack would slow down the checksum computation and thus detected. [17] This is in fact a shortcoming of this scheme, because the delay can be caused by many other factors including regular network delay.

Because public key infrastructure (PKI) is used, the resiliency of this scheme is  $N$ . Therefore, a compromised node does not yield a compromised link between any two uncompromised nodes. Connectivity for this scheme is 1, as PKI allows for communication between any two nodes. Please refer to section 5.5.1 on asymmetric key schemes.

The tradeoff for this added security is extra processing for the PKI as well as the ICE function (unfortunately ICE is documented in only one paper that is currently unavailable). The storage requirement is also increased as each node needs to store the base station’s public key, the ICE checksum, and its own private key and public keys for PKI. Communication costs is those occurred from PKI.

**Table 1: Code Attestation**

Metric	Value
resiliency	$N$
connectivity	1
processing	ICE function + PKI
storage	base station public key + PKI + ICE
communication	PKI ICE checksum

#### 5.2 Resource Testing

Resource testing was first proposed by Douceur in [8]. It recognizes the fact that Sybil nodes that claim the same identity still share a single physical node. It detects Sybil nodes through capability challenges in terms of computation, storage, and communication, so that only non-Sybil nodes will have sufficient resources to pass these challenges. However, resource testing was immediately realized as a highly impractical and ineffective defense technique against the Sybil attack. Subsequent research attempted to define new testing challenges that are associated with less overhead. One notable result is the radio resource testing, proposed by Newsome et al. [15], that takes advantage of the temporal characteristics of the typical sensor node’s radio

channel. It assumes that each node has only one radio used for both transmission and reception, and hence a sensor node cannot simultaneously send and receive. Sybil nodes that share the same underlying node and, thus radio, will not be able fulfill the challenge.

Computational and storage testing are mostly unsuitable for WSN applications because the attacker may have devices that are far more superior than a resource-limited sensor node. Moreover, overhead is the biggest concern when performing these challenges often. The global communication overhead of completing one round of the radio testing is  $N*r$  sent messages and  $r*S$  listens, where  $S$  is the number of verifier nodes and  $r$  is the number of times each node will be challenged within a round.  $r$  is usually a dependent parameter on  $S$  because the probability of not detecting a Sybil node is  $(N-S)^r$ . Since radio testing focuses strictly on the presence of an identifiable radio signals, storage and processing complexity are almost negligible (i.e. storage complexity=0, processing complexity=1). For example, nodes can fulfill the challenge by sending their node id's. Connectivity is 1 because radio testing is a detection algorithm that does not produce false positives. Any neighboring nodes can communicate with each other once they pass the challenge.

**Table 2: Radio Resource Testing**

Metric	Value
resiliency	n/a
connectivity	1
processing	negligible (1 math op)
storage	0
communication	$N*r + r*S$ per validation 0b

### 5.3 Location verification

Location verification is a technique used to detect Sybil nodes by mapping node identities to their physical location. Multiple identities residing in the same exact location means they are Sybil nodes that illegitimately use the same hardware. However, finding the absolute location of a node is a non-trivial problem. WSN research have studied node localization using estimation techniques from the received signal strength (RSSI). Besides the general difficulties of overcoming the unreliable and time-varying nature of the RSSI, a malicious node can also deliberately vary its radio transmission power to defeat a naive RSSI-based defense. Demirbas and Song proposed an RSSI-based scheme that implements a multiple observer algorithm [7] so that Sybil nodes cannot fake their physical by simply adjusting their radio power. The scheme uses ratio of the RSSI at several different nodes to estimate the true location of an untrusted node.

Demirbas and Song showed in [7] that using two observing receivers can achieve high precision (i.e. < 5% error) in detecting Sybil nodes. The communication overhead is four for every round of testing because the verifier needs to gather two RSSI values from every observer. Storage required is the product between the maximum number of simultaneous verification and the space for three RSSI ratios. The ratios are computed from four RSSI values sent by the observers and two piggybacked from the messages sent by two potential Sybil nodes. Computational complexity is six division's on the 8-bit RSSI values, three comparison operations, and pos-

sibly three subtraction's that are needed for implementing tolerance. Demirbas and Song claimed that their detection algorithm is complete in that it does not miss detecting a Sybil attacker. However, it may block certain communication channels because some nodes are falsely identified as Sybil nodes. Since the false positives rate is less than 5%, it means the connectivity is over .95, if detection algorithm is performed before the start of every communication channel.

**Table 3: Location Verification**

Metric	Value
resiliency	$N$
connectivity	>0.95
processing	negligible (12 math ops)
storage	3 RSSI ratios
communication	4 8b

### 5.4 Key-based authentication

Key-based authentication, which has been used to secure point-to-point communication, is proposed as a defense mechanism against Sybil nodes from gaining communication channels. Key-based authentication is extensively studied in terms of its storage overhead, computational efficiency and resiliency against fractional compromise of the network. It has a nice property that the amount of overhead can often be modeled as a function of security resiliency. Various keying mechanisms have been proposed to illustrate this tradeoff problem. For instance, a node may have only one master key shared with everyone, a group key shared with a group of nodes, a cluster key shared with all its neighbors, or a pairwise key shared with each immediate neighbor.

#### 5.4.1 Asymmetric key

Practicality for real deployment presents an additional challenge to any security mechanism in WSN, as sensor nodes are constrained by their limited power and memory. While many researchers in the past have rejected asymmetric key cryptography as unsuitable for WSN, work is currently being done to find ways of lowering its computational costs as well as its storage requirement for lengthy keys [1]. Among the current methods, elliptic curve cryptosystem (ECC) is the most promising, as it requires only double the key bits of AES to obtain the equivalent cryptographic strength [18, 1]. Just for comparison, the strength given by 1024 RSA key bits is equivalent to just 163 ECC key bits. [14]

Because public key authentication can communicate with every other node, connectivity is 1. Moreover, because it does not employ shared keys and compromising one node does not compromise the communication between any pair of uncompromised nodes, resiliency is  $N$ . Processing, storage, and communication all depend upon the public key authentication (and the public key length depends on the desired cryptographic strength).

#### 5.4.2 Symmetric key

In the scope of symmetric key authentication techniques, we focus on pair-wise key distribution schemes. For pair-wise key distribution schemes, when two nodes need to communicate to each other they will first establish a shared

**Table 4: Elliptic Curve Cryptosystem**

Metric	Value
resiliency	$N$
connectivity	1
processing	public key authentication
storage	public key + private key
communication	public key authentication public key length

key for the communication and then use the shared key to sign/authenticate the messages in the communication. The pair-wise key distribution scheme consists of three phases:

1. Key Setup phase: Prior to deployment, it is necessary to predistribute keys to each node.
2. Shared Key Discovery phase: After deployment, if a source node wants to communicate to a sink node, it must discover whether or not it shares a key with the sink node. If so, then that shared key is used for the communication.
3. Path-Key Establishment phase: After deployment, if a source node wants to communicate to a sink node, it must discover whether or not it shares a key with the sink node. If not, then they have to establish a session key for the communication through a multi-hop connection path.

In the following, we analyze the symmetric pair-wise key distribution schemes in two categories. The first category is the straight-forward scheme, which includes the most basic and the simplest of symmetric pair-wise key schemes. The second category is the symmetric pair-wise key distribution scheme with guarantees on resilience, which includes some of the key distribution schemes that have  $\lambda$ -secure property [2].

1. Straight-Forward Symmetric Pair-wise Key Distribution Schemes:

(a) Deterministic Key Distribution Schemes:

i. One Key Scheme:

In the key setup phase, all nodes in the network share one key. There is no shared key discovery phase and no path-key establishment phase in this scheme.

The resilience complexity of this scheme is 1; after compromising one node, an attacker can compromise the entire WSN. Thus, s/he can fabricate new nodes into the network or impersonate himself to be others and perform the Sybil attack. The connectivity of this scheme is also 1, as each pair of nodes can directly communicate. The storage complexity is the size of the shared key, as each node needs to store only one key. For processing complexity, the scheme needs to perform authentication. Communication complexity depends on the key discovery algorithm employed.

ii. All Pair Scheme:

This scheme uses distinct keys for each pair of nodes in the network. Therefore, the scheme will have  $N(N - 1)/2$  keys in total. In the key setup phase, each node has to store  $N - 1$  keys for the communication with all  $N - 1$  other nodes. In the shared key discovery phase, a source node needs to search through all stored keys to find the unique key it can use for communication with the sink node. There is no path-key establishment phase in this scheme.

The resilience complexity of this scheme is  $N$ . No matter how many nodes are compromised, an attacker cannot impersonate himself to be any uncompromised nodes because he has no information about the keys stored in those uncompromised nodes. Therefore, this scheme is a good defense against the sybil attack. The connectivity complexity of this scheme is 1, as each pair of nodes can directly communicate. The storage complexity is  $N - 1$ , as each node needs to store a unique key for the communication with the remaining  $N - 1$  nodes. For processing complexity, a source node needs to find out the corresponding key to the sink node first and then sign/authenticate each message. Communication complexity depends on the key discovery algorithm employed.

(b) Probabilistic Key Distribution Scheme:

Eschenauer and Gligor first proposed the probabilistic key distribution scheme in [10]. In their work, they describe  $K$  different keys in a key pool. Each key is accompanied with a key ID. In the key setup phase, each node will store  $k$  randomly selected keys and their respective IDs from the key pool. In the shared key discovery phase, the source node broadcast all the key IDs it has to the sink node so that they can discover whether or not they share at least one key. In the path-key establishment phase, because the source node and the sink node do not share any key, they must trust the path from source node to the sink node to exchange a session key for communication.

The resilience of the complexity of this scheme is 1. Assuming that each key is stored in each node uniformly, when an attacker compromises one node, s/he learns  $k$  keys in the key pool, and s/he can compromise  $k/K$  communication channels in the network. Thus, this scheme is under the Sybil attack. The connectivity of the scheme is  $\frac{((K-k)!)^2}{(K-2k)!K!}$ . Because the connectivity of this scheme is less than one, the scheme is under the routing attack.

**DEFINITION 1.** Routing Attack is an attack on path-key establishment phase. When a source node establishing the session key with a sink node through a path, if any intermediate node on the path is compromised by an attacker, the attacker can catch the session key and compromise all the communications between the source node and the sink node.

The storage complexity of this scheme is  $k$ , as each node has to store  $k$  different keys. For processing complexity, a source node and a sink node has to find out whether they share a key or not. If not, they need to generate a random session key and exchange that session key. After figuring out the key for communication, the scheme needs to sign/authenticate each message. For communication complexity, a source node has to broadcast all the key IDs it has to verify whether it shares one key with the sink node or not. If not, they will need to find a communication path and exchange a session key through that path.

2. Symmetric Pair-wise Key Distribution Scheme with Guarantees on resiliency: There are mainly two approaches to achieve the  $\lambda$ -secure property. The first is the matrix approach, and the second is the polynomial approach.

(a) Matrix Approach:

- i. Single Space Key Distribution Scheme:

Blom first proposed this scheme in [2]. During the key setup phase, the scheme generates  $(\lambda+1) \times N$  public matrix  $G$  and a private symmetric  $(\lambda+1) \times (\lambda+1)$  matrix  $D$ . Each node will store one public column in the matrix  $G$  and one private row in matrix  $(D \cdot G)^T$ . In the shared key discovery phase, the source node  $u$  and the sink node  $v$  broadcast their public columns  $u_c$  and  $v_c$  to each other. Let  $u_r$  and  $v_r$  be the private row for source  $u$  and sink  $v$ , respectively. Then, the source node can compute the shared key by  $u_r \cdot v_c$ , and the sink node can compute the shared key by  $v_r \cdot u_c$ . There is no path-key establishment phase in this scheme.

The resilience complexity of this scheme is  $\lambda$ . Before compromising more than  $\lambda$  nodes in the network, an attacker cannot reconstruct the private matrix  $D$ . Thus an attacker cannot fabricate new nodes or impersonate existing nodes in the network. Thus, this scheme is a good defense against the Sybil attack. The connectivity of this scheme is 1, as every pair of nodes can come up with an unique shared key for communication. The storage complexity of this scheme is  $2(\lambda+1)$ . Each node needs to store a public column with  $\lambda+1$  elements and a private row with  $\lambda+1$  elements. For processing complexity, both the source node and the sink node need the inner product operation between two vectors with length  $\lambda+1$  to find out the shared key. Next, they need to sign/authenticate all messages they exchange. For communication complexity, both source node and sink node have to broadcast their public columns.

- ii. Multiple Space Key Distribution Scheme:

Du et al. first proposed the multiple space key distribution scheme in [9]. A  $(\lambda+1) \times N$  public matrix  $G$  accompanied with a  $(\lambda+1) \times (\lambda+1)$  private matrix  $D$  form a space.

In the key setup phase, the scheme generates  $\omega$  independent spaces. For each node, the scheme randomly selects  $\tau$  distinct spaces from the  $\omega$  spaces, and stores a public column and a private row for each space in the node. In the shared key discovery phase, a source node will broadcast its space IDs. If the sink node shares at least one space with the source node, then they can pick up any of the shared space to communicate. After that, they can follow the single space shared key discovery phase to discover the shared key. For the path-key establishment phase, because the source node and the sink node do not share any space, they do not share any keys. Thus, they have to trust the path from source node to the sink node to exchange a session key for communication.

The resilience complexity for this scheme is  $\lambda$  for each space. However, if an attacker compromises nodes at random, then the probability that he can compromise a space by capturing  $\lambda+1$  nodes is only  $\omega(\frac{\tau}{\omega})^{\lambda+1}$ . The connectivity for this scheme is  $\frac{((\omega-\tau)!)^2}{(\omega-2\tau)! \omega!}$ . Because the connectivity of this scheme is less than one, the scheme is under *therouting attack*. Moreover, because there are several connections share the same space, the scheme is under the connectivity induced sybil attack.

**DEFINITION 2.** Connectivity Induced Sybil Attack is an attack on path-key establishment phase. When an attacker compromise a node  $u$  with space  $s_1, s_2, \dots, s_\tau$ , he can have the full power of that node. Therefore, the attacker can communicate to any node inside the space  $s_i, \forall 1 \leq i \leq \tau$ . By fabricating and deploying nodes that can communicate to other nodes in these spaces, the attacker can have more influence during the path-key establishment phase. For any two nodes that does not share any space with each other but both share at least one space with  $u$ , they will have higher probability to establish their session key through the fabricated nodes due to the large number of the fabricated nodes.

The storage complexity of this scheme is  $2\tau(\lambda+1)$  matrix elements and  $\tau$  space IDs. For processing complexity, both the source node and the sink node need the inner product operation between two vectors with length  $\lambda+1$  to discover the shared key. If they do not share any space, they have to generate a random session key and exchange that session key. For communication complexity, the source node has to broadcast the space IDs it has to the sink node. If they share at least one space, they need to broadcast their public columns to each other. Otherwise, they have to use a path to exchange a session key.

- iii. Location-Based Multiple Space Key Distribution Scheme:

Huang et al. proposed a location-based dis-

tribution scheme in [11]. In this scheme, the deployment region is divided into  $L \times W$  rectangular bins. The nodes are divided into  $L \times W$  groups also. Group  $(i, j)$  will be deployed into regions with coordinate  $(i, j)$ . For communication inside the bins, the scheme use distinct  $\lambda$ -secure space for each bin. For communication across bins, the scheme defines 8 communication directions as shown in Figure ???. Each node will pick one neighbor node in each direction randomly, then store a pair of (key, node ID) in both the node and the neighbor. Thus a node can communicate to 8 neighbors in each direction by the assigned pair-wise key. If the source node needs to communicate with an external sink node that they do not share any key with, they have to trust the path from source node to the sink node in order to exchange a session key for communication.

The resilience complexity of the internal communication is  $\lambda$  because it applies distinct spaces for each bin. The resilience complexity of the external communication is  $N$ . Because all the pair-wise keys used for external communication are distinct, the compromised keys will not reveal any information about the uncompromised keys. The connectivity of the internal communication is 1. The connectivity of the external communication is  $8/(L \times W)$ . Because the connectivity of external communication is less than one, it is under the *routing attack*. The storage complexity of each node is  $2(\lambda + 1)$  vector elements for internal communication, 8 pair-wise keys, and 8 node IDs. For processing complexity, the operations required by internal communications are just the same as the operations required by the single space key distribution scheme. For external communications, if the source node shares one key with the sink node, the only extra necessary operation is authentication. However, if they do not share a key, then they have to generate a random session key and exchange that key. The communication complexity for internal communications is just the same as that for the single space key distribution scheme. The communication complexity for external communications are similar to that of the probabilistic key distribution scheme: if the source node and the sink node share one key, then they can communicate directly. If they do not share one key, they have to exchange a session key first, then start communication by that session key.

(b) Polynomial Approach:

- i. Single Polynomial Key Distribution Scheme: Blundo et al. first proposed the single polynomial key distribution scheme in [3]. The scheme uses a symmetric polynomial  $P(x, y)$  with a degree equal to  $\lambda$ . In the key setup phase, each node  $u$  stores a unique node ID

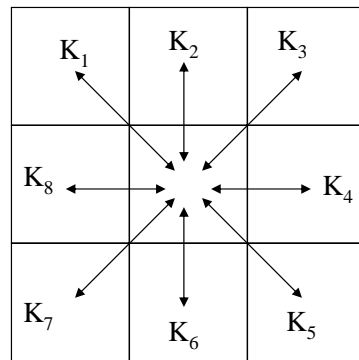


Figure 1: The eight possible communication directions across bins.

$u_d$  and the  $\lambda + 1$  coefficients of polynomial  $P(u_d, y)$ . In the shared key discovery phase, both the source node  $u$  and the sink node  $v$  broadcast their IDs  $u_d$  and  $v_d$  to each other. Then they can establish the shared key by calculating the value of  $P(u_d, v_d)$  and  $P(v_d, u_d)$ , respectively. There is no path-key establishment phase in this scheme.

The resilience complexity of this scheme is  $\lambda$ . Before compromising more than  $\lambda$  nodes in the network, an attacker cannot reconstruct the original polynomial  $P(x, y)$ . Thus an attacker cannot fabricate new nodes or impersonate other nodes in the network. Thus, this scheme is a good defense against sybil attack. The connectivity of this scheme is 1. Every pair of nodes can come up with a unique shared key for communication. For storage complexity, each node needs to store its node ID and  $\lambda + 1$  polynomial coefficients. For processing complexity, both the source node and the sink node need to compute the value of the polynomial to find out their shared key. Then, they need to sign/authenticate all messages they exchange. For communication complexity, both source node and sink node have to broadcast their node IDs.

- ii. Polynomial Pool Key Distribution Scheme: Liu et al. proposed the polynomial pool key distribution scheme in [12]. This scheme is similar to the multiple space key distribution scheme. The only difference is that the key pools are changed from multiple spaces to multiple polynomials.

The resilience complexity and connectivity of this scheme is the same as those for the multiple space key distribution scheme. The storage complexity of this scheme is  $\tau(\lambda+1)$  polynomial coefficients,  $\tau$  polynomial IDs, and a node ID. For processing complexity, both the source node and the sink node need to calculate the polynomial value to find out the shared key. If they do not share any polynomial, they have to generate a random session key and exchange that session key. For com-

munication complexity, the source node has to broadcast the polynomial IDs it has to the sink node. If they share at least one polynomial, they need to broadcast their IDs to each other. Otherwise, they have to use a path to exchange a session key.

iii. Location-Based Polynomial Pool Key Distribution Scheme:

Liu et al. proposed the location-based polynomial pool key distribution scheme in [13]. The deployment method is the same as that of the location-based multiple space key distribution scheme. For internal communication, the nodes inside a bin share one polynomial for communication. For external communication, each node shares four polynomials with nodes located at its north, east, south, and west neighbor bins, respectively. If the source node does not share any polynomial with the sink node, they have to trust the path from source node to the sink node to exchange a session key for communication. The resilience complexity of the internal communication is  $\lambda$  because the scheme applies distinct polynomials to each bin. The resilience complexity of the external communication is still  $\lambda$  because nodes in neighboring bins share one polynomial. The connectivity of the internal communication is 1. The connectivity of the external communication is  $4/(L \times W)$ . Because the connectivity of external communication is less than one, it is under the *routing attack*. The storage complexity of each node is  $4(\lambda + 1)$  polynomial coefficients and a node ID. For processing complexity, the operations required by internal communications are the same as the operations required by the single polynomial key distribution scheme. For external communications, if the source node shares one polynomial with the sink node, all the necessary operations are the same as the operations required by the single polynomial key distribution scheme. However, if they do not share any keys, then they have to generate a random session key and exchange that key. The communication complexity for internal communications is the same as that for the single polynomial key distribution scheme. The communication complexity for external communications is the same as that for the single polynomial key distribution scheme if the source node and the sink node shares one polynomial. If they do not share one polynomial, they have to exchange a session key first, then start communication by that session key.

For the analysis of all the aforementioned pairwise key distribution schemes, we know that those with connectivity less than one are under the routing attack. For example, the probabilistic key distribution scheme, the multiple space key distribution scheme, the external communication in the

location-based multiple space key distribution scheme, the polynomial pool key distribution scheme, and the location-based polynomial key distribution scheme are all vulnerable to such an attack. Moreover, because the multiple space key distribution scheme (the polynomial pool key distribution scheme) shares the same space (polynomial) among connections, it is under the *connectivity-induced Sybil attack*. In order to alleviate these two vulnerabilities, we can apply the multi-path key reinforcement scheme proposed by Chan et al. [5]. Instead of establishing a session key using only one path, the scheme establishes a session key using multiple disjoint paths. When a source node needs to establish a session key with a sink node, it first generates  $m$  random keys  $k_1, k_2, \dots, k_m$ . Next, it selects  $m$  disjoint paths  $p_1, p_2, \dots, p_m$  from the source node to the sink node and then sends each random key  $k_i$  to the sink node through path  $p_i$ ,  $\forall 1 \leq i \leq m$ . After the sink node receives all  $m$  keys, they will perform an exclusive-or on all  $m$  keys as the session key  $k_s$ .  $k_s = k_1 \oplus k_2 \oplus \dots \oplus k_m$ .

With this multi-path reinforcement scheme, an attacker cannot compromise the session key without compromising at least one node on each path. Moreover, an attacker does not know the paths chosen to establish the session key. This scheme can be applied to any other key distribution schemes with connectivity less than one. Moreover, the scheme has no overhead on the resilience complexity, connectivity, and storage complexity. However, it has some overhead on processing complexity and communication complexity. For the processing complexity, instead of generating only one session key, the source node needs to generate  $m$  different random keys, and both the source node and the sink node have to use exclusive-or operations on all the keys to establish the session key. For the communication complexity, instead of sending a message containing the session key, the source node needs to send  $m$  messages containing different random keys.

The analysis among the above pair-wise key distribution schemes are summarized in table 5. For the scheme row, OK means the one key scheme; AP means the all pair scheme; PK means the probabilistic key distribution scheme; SS means the single space key distribution scheme; MS means the multiple space key distribution scheme; LMS means the location-based multiple space key distribution scheme; SP means the single polynomial key distribution scheme; PP means the polynomial pool key distribution scheme; LPP means the location-based polynomial pool key distribution scheme. The I column and E column state for internal communication and external communication. In the storage row, ME means matrix element; PC means polynomial coefficient; SID means space ID; NID means node ID; PID means polynomial ID. In the Processing row, VIP means the vector inner product operation; PE means the polynomial evaluation operation; SKG means the session key generation operation for path-key establishment phase. In the communication row, KD means the key discovery communication; SD means the space discovery communication; PD means the polynomial discovery communication; PCB means the public column broadcast operation; NIDB means the node ID broadcast operation; SKE means the session key exchange operation for path-key establishment phase.

Scheme	OK	AP	PK	SS	MS	LMS		SP	PP	LPP	
						I	E			I	E
Resilience	1	N	1	$\lambda$	$\lambda$	$\lambda$	N	$\lambda$	$\lambda$	$\lambda$	$\lambda$
Connectivity	1	1	$\frac{((K-k)!)^2}{(K-2k)!K!}$	1	$\frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!}$	1	$\frac{8}{(L \times W)}$	1	$\frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!}$	1	$\frac{4}{(L \times W)}$
Storage	1 key	N-1 key	k key	$2(\lambda+1)$ ME	$2\tau(\lambda+1)$ ME	$2(\lambda+1)$ ME	8 key 8 NID	$\lambda+1$ PC	$\tau(\lambda+1)$ PC	$\lambda+1$ PC	$4(\lambda+1)$ PC
Processing		Search	Search SKG	VIP	Search VIP SKG	VIP	SKG	PE	Search PE SKG	PE	PE SKG
Communication			KD SKE	PCB	SD PCB SKE	PCB	SKE	NIDB	PD NIDB SKE	NIDB	NIDB SKE

Table 5: Evaluation among the 9 pair-wise key distribution schemes.

## 6. CONCLUSION

We have evaluated a wide range of defenses against the Sybil attack in wireless sensor networks using five metrics: resiliency, connectivity, processing complexity, storage complexity, and communication complexity. Often times, these metrics alone are not enough to determine the usability of a particular defense, although they do aid in gauging the security and practicality tradeoffs. For example, radio resource testing has reasonable metric values. However, it is deemed unusable due to its requirement of each node having only one radio. Similarly with code attestation, it requires the attacker to go through great lengths to forge multiple identities in order to perform a Sybil attack. However, its current proposed use to detect a Sybil attack (i.e. computation time) is not a very reliable method as many factors can contribute to this (i.e. regular network delays). On the other hand, location verification has perfect resiliency and is, of all the defenses, most likely to handle dynamic network topologies. However, extra communication overhead is required in order to perform location validations. Moreover, each node is required to rely on its observers; if a Sybil attack had already succeeded, the Sybil node can lie about the location of other Sybil nodes to thwart detection.

Key-based methods seemed most suitable to our metrics - ECC is a promising asymmetric key cryptography solution due to its shorter key-length requirement. Additionally, public key authentication offers high connectivity and high resiliency, both very desirable traits. More research in this area should show promising results.

As for pairwise key predistribution schemes, in order to protect sizeable networks, one always has to deal with the key management problem. Multi-space schemes, including location-based schemes, work best for such large networks but require dense node population. Similarly, probabilistic models also require dense, uniform networks. Moreover, it performs poorly in both resiliency and connectivity. We consider polynomial-based schemes to be an improvement over matrix-based schemes as they offer the same resiliency and connectivity, but necessitate less storage and less communication overhead.

## 7. FUTURE WORK

There is an ever increasing demand for mobile sensor networks that can perform maneuvers and respond to environmental stimulus. Future work would be to extend the analysis presented in this paper to mobile sensor networks that have dynamic topology, and possibly derive new metrics for evaluation in this domain. An even greater next step would be to propose new solutions based on our findings.

Another possible dimension for future exploration is to see how the values in our complexity analysis translates concretely at the implementation level. One can accomplish this by simulating the mentioned security algorithms with a model of the hardware.

## 8. REFERENCES

- [1] Erik-Oliver Bläß and Martina Zitterbart, *Towards acceptable public-key encryption in sensor networks*, IWUC, INSTICC Press, 2005, pp. 88–93.
- [2] R. Blom, *An optimal class of symmetric key generation systems*, Eurocrypt 84., 1985.
- [3] Santis A. Herzberg A. Kutten S. Vaccaro U. Blundo, C. and M. Yung, *Perfectly-secure key distribution for dynamic conferences*, Crypto 92, 1992.
- [4] Seyit A. Camtepe and Bülent Yener, *Key distribution mechanisms for wireless sensor networks: a survey*.
- [5] Perrig A. Chan, H. and D. Song, *Random key predistribution schemes for sensor networks*, IEEE Symposium on Research in Security and Privacy, 2003.
- [6] David Culler, Deborah Estrin, and Mani Srivastava, *Overview of sensor networks*, IEEE Computer, Special Issue in Sensor Networks, August 2004.
- [7] Murat Demirbas and Youngwhan Song, *An RSSI-based scheme for sybil attack detection in wireless sensor networks*, WOWMOM '06: Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks (Washington, DC, USA), IEEE Computer Society, 2006, pp. 564–570.
- [8] John R. Douceur, *The sybil attack*, (2002), 251–260.
- [9] Deng J. Han Y. Du, W. and P. Varshney, *A pairwise key pre-distribution scheme for wireless sensor*

- networks*, Proceedings of the 10th ACM conference on Computer and Communication Security, 2003.
- [10] L. Eschenauer and V. D. Glogor, *A key-management scheme for distributed sensor networks*, 9th ACM conference on Computer and Communications Security, 2002.
  - [11] Mehta M. Medhi D. Huang, D. and L. Harn, *Location-aware key management scheme for wireless sensor networks*, 2nd ACM workshop on Security of Ad Hoc and Sensor Networks, 2004.
  - [12] D. Liu and P. Ning, *Establishing pairwise keys in distributed sensor networks*, 10th ACM conference on Computer and Communications Security, 2003.
  - [13] Donggang Liu and Peng Ning, *Location-based pairwise key establishments for static sensor networks*, SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, ACM, 2003, pp. 72–82.
  - [14] D. Malan, M. Welsh, and M. Smith, *A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography*, 2004.
  - [15] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig, *The sybil attack in sensor networks: analysis & defenses*, IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks (New York, NY, USA), ACM, 2004, pp. 259–268.
  - [16] Tanya Roosta, S. P. Shieh, and Shankar Sastry, *Taxonomy of security attacks in sensor networks and countermeasures*, The First IEEE International Conference on System Integration and Reliability Improvements, December 2006.
  - [17] Arvind Seshadri, Mark Luk, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, *Scuba: Secure code update by attestation in sensor networks*, WiSe '06: Proceedings of the 5th ACM workshop on Wireless security (New York, NY, USA), ACM, 2006, pp. 85–94.
  - [18] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz, *Energy analysis of public-key cryptography for wireless sensor networks*, PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications (Washington, DC, USA), IEEE Computer Society, 2005, pp. 324–328.