Security in Wireless Sensor Networks

Manvendra Singh[1], Rohan Verma[2], Sarthak Mittal[3]
{[1]ms234, [2]rv285, [3]sm273}@snu.edu.in

***Abstract*—Wireless Sensor Network is an emerging area that shows great future prospects. Today such networks are used in many industrial and consumer applications, such as military, industrial process, monitoring health and in automated and smart homes. So far, the researchers have only focused on making WSNs useful, feasible, and less emphasis was placed on security. The sensors used are susceptible to different types of attacks, denial of service, physical tampering. In hostile scenarios, it is very important to protect WSNs from malicious attacks. This is the reason we need better security against these challenges, threats and issues in WSN. The intent of this paper is to shed light on the security related issues and challenges in wireless sensor networks investigated by researchers in recent years and that shed light on future directions for WSN security.**

Keywords—Attack, Challenge, Sensor, Security, Wireless Sensor Networks

I. Introduction

The basic idea of sensor network is to disperse tiny sensing devices; which are capable of sensing some changes of incidents/parameters and communicating with other devices, over a specific geographic area for some specific purposes like target tracking, surveillance, environmental monitoring etc. [2]. Today sensors can monitor temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions and what not. Basically the major challenges we face before employing any security scheme in wireless sensor networks is related to the size of sensors, their processing power, memory and type of tasks expected from the sensors. A major benefit of these systems is that they perform in-network processing to reduce large streams of raw data into useful aggregated information [4]. This is the reason protecting all this information is critical. Therefore, the security in WSNs becomes an important and a challenging design task. When working with the security mechanism in sensor networks key areas where one should focus on are limited energy, limited memory, limited computational power, limited communication bandwidth, limited communication range [5]. Here, we outline security issues in these networks, we will discuss challenges and issues in WSN and suggest future directions for research.

II. Requirements For Sensor Network Security

In this section we will discuss the important properties related to security in the field of sensor network, and how they are directly applicable in a typical sensor network [3].

A. Self-organization

Nodes usually in a state of unattended, so when the node is destroyed, or network structure has changed, the nodes need to self-organization and self-repair, which requires the dynamic characteristics of security mechanism, especially for dynamic allocation of key and dynamic maintenance of trust relationship [3].

B. Data confidentiality

A sensor network should not leak sensor readings to the neighboring networks. In many applications (e.g., key distribution) nodes communicate highly sensitive data. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving confidentiality [3].

Given the observed communication patterns, we set up secure channels between nodes and base stations and later bootstrap other secure channels as necessary [3].

C. C. Data authentication

Message authentication is important for many applications in sensor networks (including administrative tasks such as network reprogramming or controlling sensor node duty cycle) [3].

Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender. In the two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender [3].

This style of authentication cannot be applied to a broadcast setting, without placing much stronger trust assumptions on the network nodes. If one sender wants to send authentic data to mutually untrusted receivers, using a symmetric MAC is insecure: anyone of the receivers knows the MAC key, and hence, could impersonate the sender and forge messages to other receivers. Hence, we need an asymmetric mechanism to achieve authenticated broadcast [3].

D. Data integrity

While communicating, data integrity ensures that the data received on the receiver side in not changed in transit because of some reasons. In SPINS, we achieve data integrity through data authentication, which is a stronger property [3].

E. Data freshness

Sensor networks send measurements over time, data confidentiality and authentication it is not enough; we also must ensure each message is fresh. Informally, data freshness means that data is new and not the repeated one. We identify two types of freshness: weak freshness, which provides partial

message ordering, but carries no delay information, and strong freshness, which provides a total order on a request–response pair, and allows for delay estimation. Weak freshness is useful for sensor measurements, while strong freshness is useful for time synchronization within the network. Time synchronization: Many applications of wireless sensors require time synchronization, and the corresponding security mechanism should also be time synchronization [3].

III. Security Challenges, Threats And Issues In WSN

In sensor network application addressing of security concern may arise, WSNs must consider a variety of unique challenges that make them very vulnerable to malicious attacks in hostile environments such as a military battlefield. The first challenges of security in sensor networks lie in the conflicting interest between minimizing resource consumption and maximizing security. Therefore the usefulness of a potential solution depends how good the compromise it achieves is. The resource in this context includes energy as well as computational resource like CPU cycles, memory, and communication bandwidth.

A. Limited Resources

The constrained resources make it very difficult to implement strong security algorithms on a sensor platform due to the complexity of the algorithms. Most of the time, symmetric key cryptography is the first choice when designing a security protocol for WSNs, although public key cryptography is possible under careful optimization in design and implementation. In addition, a WSN can scale up to thousands of sensor nodes. These pose the demand for simple, flexible, and scalable security protocols. However, to design such security protocols is not an easy task. A stronger security protocol costs more resources on sensor nodes, which can lead to the performance degradation of applications. In most cases, a trade-off must be made between security and performance. However, weak security protocols can be broken easily by attackers [1].

B. Unreliable Communication

Ad-hoc networking topology renders a WSN susceptible to link attacks ranging from passive eavesdropping to active interfering. [5] A wireless channel is open to everyone. With a radio interface configured at the same frequency band, anyone can monitor or participate in communications. This provides a convenient way for attackers to break into WSNs. As in the Internet, most protocols for WSNs do not include potential security considerations at the design stage. Due to standard activity, most protocols are known publicly. Therefore, attackers can easily launch attacks by exploiting security holes in those protocols. A WSN is usually deployed in hostile areas without any fixed infrastructure. It is difficult to perform continuous surveillance after network deployment. Therefore, a WSN may face various attacks. [1]

Further, the wireless communication characteristics of WSN render traditional wired-based security schemes impractical. The broadcast nature of the wireless communication is a simple candidate for eavesdropping. In most of the cases various security issues and threats related to those we consider for wireless ad hoc networks are also applicable for wireless sensor networks. [2] While ad hoc networks are self-organizing, dynamic topology, peer to peer networks formed by a collection of mobile nodes and the centralized entity is absent; the wireless sensor networks could have a command node or a base station, therefore applying mechanisms devised for ad hoc networks also will fail to a certain degree.

IV. Key Issues

A. Key Management in WSN

Confidentiality, integrity, and authentication services are critical to preventing an adversary from compromising the security of a WSN. Key management is likewise critical to establishing the keys necessary to provide this protection in WSN. However, providing key management is difficult due to the ad hoc nature, intermittent connectivity, and resource limitations of the sensor network environment. Traditional key management service is based on a trusted entity called a certificate authority (CA) to issue public key certificate of every node. The trusted CA is required to be online in many cases to support public key revocation and renewal. But it is dangerous to set up a key management service using a single CA in a sensor network. The single CA will be the vulnerable point of the network. If the CA is compromised, the security of the entire network is crashed. How to setup a trusted key management service for the WSN is a big issue [6].

B. Securing routing of WSN

There are two kinds of threats to ad-hoc routing protocols:

1) External attackers.

The attacks include injecting erroneous routing information, replaying old routing information, and distorting routing information. Using these ways, the attackers can successfully partition a network or introduce excessive traffic load into the network, therefore cause retransmission and ineffective routing. Using cryptographic schemes, such as encryption and digital signature can defend against the external attacks.

2) Internal compromised nodes.

They might send malicious routing information to other nodes. It is more severe because it is very difficult to detect such malicious information because compromised node can also generate valid signature. Existing routing protocols cope well with the dynamic topology, but usually offer little or no security measures. An extra challenge here is the implementation of the secured routing protocol in a network environment with dynamic topology, vulnerable nodes, limited computational abilities and strict power constraints [6].

C. Prevention of Denial-of-service

Strictly speaking, although we usually use the term Denial-of-service (DoS) to refer to an adversary's attempt to disrupt, subvert, or destroy a network, a DoS attack is any event that

diminishes or eliminates a network's capacity to perform its expected function. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interaction between these factors can cause a DoS. An adversary may possess a broad range of DoS attack capabilities in WSN. For example, a wireless sensor network can be aerially deployed in enemy territory. If the enemy already has a wired network and power grid available and can interact with the newly deployed sensor network, it can apply powerful back-end resources to subvert or disrupt the new network [6].

V. KEY MANAGEMENTS

A. Symmetric key management

Most symmetric key algorithms, such as Data Encryption Standard (DES) [17] or Rivest Cipher 5 (RC5) [18], require simple hash, rotation, or scrambling operations, which can be efficiently implemented in hardware or software. On the other hand, asymmetric key algorithms, such as Diffie-Hellman [22] or Rivest Shamir Adleman (RSA) [19], require exponential operations over a field modulo a large prime number, which are more complex than symmetric key operations. Therefore, the symmetric key technology is more viable on resource constrained low-end devices than the asymmetric key technology.

Most of the security protocols in the literature for WSNs are based on symmetric key technology. A basic problem for applying the symmetric key technology is how to establish a symmetric key between two sensor nodes. A simple approach is to distribute a global key [20] to all the sensor nodes. This approach is secure from external attackers that do not know the key but not from internal attackers because the key can be exposed if a node is compromised. Due to the existence of BSs, centralized key distribution can be used. In particular, each sensor node shares a unique key with a BS, which acts as a key distribution center (KDC).

If two nodes must communicate securely, they can acquire a shared key from the BS, which unicasts the key to each of them. This centralized approach could incur a large amount of communication overhead because two neighboring nodes might be required to do handshakes through a central key server at a distant place. In addition, the key server could become a potential point of failure in that the entire network is disabled if the server is corrupted by an attacker.

Most recent solutions to key establishment in WSNs follow a distributed approach, called key pre-distribution, where every sensor node is preloaded with key material with which to establish shared keys with other nodes after being deployed into the network terrain. There are two components in this approach: one is how to establish a shared key with key materials, and the other is how to distribute key materials

In the design of the distributed approach, several problems must be considered:

• Memory cost: The memory resource of sensor nodes is scarce. We cannot distribute a large amount of key material into each node.

• Resilience to node compromise: Usually, it is impossible to prevent an attacker from compromising some nodes. We can do nothing to rescue those compromised nodes. However, a good scheme should reduce the impact of the node compromise attacks on other normal nodes as much as possible. By compromising a node, an attacker can learn the keys the compromised node uses to communicate with other nodes, but it should not be able to learn keys that the compromised node does not know so that communications between normal nodes are still safe.

• Local secure connectivity: Because each node cannot store much key material, it usually can establish shared keys with a subset of its neighboring nodes. Local secure connectivity is the probability that two neighboring nodes establish a shared key directly, that is, the portion of neighbors with whom a node can establish shared keys in one hop. It is directly related to the communication overhead of key establishments. In WSNs, high local secure connectivity is desirable because it means that each node is not required to spend too much energy on the establishment of indirect keys with neighbors through multi-hop routing, thus saving a lot of communication overhead.

B. Asymmetric Key Management

Though it is much more computationally expensive, asymmetric key technology is easier to manage and more resilient to node compromise than symmetric key technology. Each node can keep secret its private key and only publish its public key; therefore, compromised nodes cannot provide clues to the private keys of non-compromised nodes.

1) Computational Efficiency

Recently, some researchers began to investigate the feasibility of using asymmetric key technology on sensor platforms because of the rapid advances in hardware capability. The most challenging problem here is how to perform asymmetric key algorithms in an efficient way. One approach is to use specific parameters that can speed asymmetric key algorithms without compromising security. For example, Tiny public key (TinyPK) [21] uses RSA-based certificates to authenticate external parties before they can access the network, where the RSA [18] public key is chosen as e = 3, such that the signature verification at the sensor side is simplified. Moreover, the Diffie-Hellman algorithm [22] is used in TinyPK [21] to exchange keys between sensor nodes, where the base of exponentiation is chosen as 2, such that the exponential operation is simplified.

2) Application

In addition to RSA for authentication and Diffie-Hellman for key establishment [21], ECC also is attracting interest for the security design of WSNs due to its efficiency. Huang et al. [23] considered a sensor network consisting of secure managers and several sensor nodes. An ECC-based authenticated key establishment protocol is proposed for the key establishment between secure managers and sensor nodes. To reduce the computational overhead of sensor nodes, most computationally expensive asymmetric key operations are put

on the secure manager side. Zhou, Zhang, and Fang [24] designed an access control protocol based on ECC. In particular, the Elliptic Curve Digital Signature Algorithm (ECDSA) [73] is used to authenticate new sensor nodes when they join the network, and the ECC- based Diffie-Hellman algorithm is used to establish shared keys between sensor nodes.

3) Authenticate Public Keys

Another critical issue of applying asymmetric key technology is the authenticity of public keys. A public key should be owned by the node that claims to have it. Otherwise, attackers can easily impersonate any node by claiming its public key and launch the man-in-the-middle attack. For example, a malicious node C may impersonate node B to node A and also impersonate A to B if A and B cannot verify the public key of each other. In this way, node C can act as an invisible router and learn all the messages between A and B. The conventional solution to public key authentication is to use a certificate signed by a trustful certificate authority (CA). Therefore, node B can send its public key with corresponding certificate to node A such that A can verify the correctness of the certificate with the well-known public key of the CA. Node B can verify the authenticity of A's public key by following the same procedure.

C. Authentication and Integrity

1) One-hop authentication

To support one-hop authentication, a shared link-layer key is required between neighboring nodes. Most symmetric key and asymmetric key management schemes discussed previously can achieve this goal. TinySec [26] is the first fully-implemented link-layer security architecture for WSNs, providing encryption and authentication. It defines two packet types: TinySec-AE and TinySec-Auth. In TinySec- AE packets, the data payload field is encrypted according to Skipjack [27], which is a lightweight block cipher. All the packets of the two types include MACs to provide the packet authentication service. However, TinySec does not discuss how to establish link layer keys; therefore, TinySec can be combined with key establishment protocols discussed previously to provide a link- layer security solution.

2) Multi-hop authentication

Like the case in one-hop authentication, an end-to-end shared key can support multihop authentication. Most symmetric key establishment schemes discussed previously target the link-layer key establishment. Based on the link-layer secure infrastructure, a multi-hop key can be negotiated between two end nodes through a multihop path. However, the multihop key negotiation may fail if one of the intermediate nodes along the path is compromised. To deal with this problem, multipath enhancement combining secret sharing [can be performed. If an asymmetric key infrastructure is available, the establishment of a multihop key is more secure. Because only the two end nodes can encrypt and decrypt the negotiation messages, the compromise of intermediate nodes does not expose the end-to-end shared key. Unlike the authentication based on a shared key, a public key certificate also can support multihop authentication.

Access points, forwarding nodes, and mobile sensor nodes. Packets generated by sensor nodes are forwarded by forwarding nodes to access points, where they are routed to specific applications. Preloaded RSA-based initial certificates are used to authenticate sensor nodes and access points to external applications. During the lifetime of the network, applications continue to renew certificates for access points and sensor nodes. Considering the limited capability of access points and sensor nodes, applications authenticate new certificates using the TESLA [29] protocol. The authentication based on the shared key is more efficient than the certificate-based one. A node at one end can verify the identity of a node at the other end through the challenge-response approach based on the shared key. The authentication involves only symmetric key operations such as hash.

3) Broadcast authentication

Broadcast is a common method to disseminate information in a WSN when a source node intends to spread the same messages, such as commands or queries, to a group of nodes. Each broadcast packet should be authenticated so that attackers cannot inject false information. Currently, most broadcast authentication schemes are based on symmetric key techniques due to their efficiency.

D. Open issues

Most current symmetric key schemes for WSNs aim at link-layer security for one-hop communications, but not the transport layer security for multi hop communications, because usually, it is unlikely for each node to store a transport layer key for each of the other nodes in a network due to the huge number of nodes. Asymmetric key technology is expensive but has flexible manageability. Any pair of nodes can establish a shared key using asymmetric key techniques such as Diffie-Hellman. A more promising approach is to combine both techniques such that each node is equipped with an asymmetric key system and relies on it to establish end-to-end symmetric keys with other nodes. To achieve this goal, a critical issue is to develop more efficient asymmetric key algorithms and/or their implementations so that they can be widely used on sensor platforms. How to prove the authenticity of public keys is another important problem. Identity-based cryptography is a shortcut to avoid the problem. There still is a demand for the development of more efficient symmetric key algorithms because encryption and authentication based on symmetric keys are very frequent in the security operations of sensor nodes. Key revocation is another unaddressed problem. When a node is detected as a malicious one or as a compromised one, its key must be revoked such that it cannot participate in nor-mal communications. Though some issues are discussed in [30], they mainly target RPK [31]. Because there are so many schemes following different approaches, it is very difficult to design a universal key revocation scheme. It is still an open problem for resource constrained WSNs.

VI. SECURE ROUTING

Essentially the purpose of the network is to provide a stable infrastructure to deliver data between source node and destination node. The main problem is data delivery and routing where we need to find path between source node and destination node. This is how routing protocol comes into the picture and this is the most critical component. If the routing protocol fails due to any reason then application layer also fails; this is where the entire network becomes useless. Therefore, secure routing is very important to guarantee the network functionality [1].

A. Problems

Routing techniques for WSN are used to minimize energy consumption. The different types of flat routing protocols like directed diffusion, data aggregation and in-network processing we can reduce the number of transmissions of redundant data. Clustering is a critical process to build up a hierarchical WSN in hierarchical routing protocols such as the low energy adaptive clustering hierarchy (LEACH). Sensor nodes in the local area cooperate to select a cluster head that may be more powerful so that it can perform more complex operations such as data aggregation or long distance routing. In location-based routing protocols such as Geographical and Energy-Aware Routing (GEAR), the location of a sensor node, which can be estimated by global positioning system (GPS) devices or GPS-free methods, is used as the routing metric. Unencrypted packets that carry routing information can be easily subject to eavesdroppers so that attackers can discover the network topology.

Attackers can inject false routing information to launch a Sybil attack or redirect packets to change network topology. Both of the attacks can change the network traffic pattern so that some malicious nodes can receive most of the traffic before it arrives at the BS. A location deterministic operation is vulnerable because it requires cooperation among several nodes and may not be successful if some of them are malicious. A malicious node intentionally may drop some of the passing traffic. Of course, it can drop all the packets to act like a black hole, but this is easy to detect. A malicious node may drop the packets from some selected nodes and forward those from other nodes. A more subtle way is to drop packets intermittently so that it behaves like an unstable channel. Most routing protocols require that each sensor node periodically broadcast routing information to maintain the network topology. If the time synchronization in the maintenance operation is attacked, the whole network fails. Though several proposals tried to secure ad hoc routing protocols, they hardly can be applied in WSNs for three reasons. First, those proposals all target ad hoc networks, which are different from WSNs in terms of resources and communication patterns. Second, those proposals are security extensions of existing ad hoc routing protocols such as dynamic source routing (DSR), ad hoc on-demand distance vector (AODV), or destination-sequenced distance vector (DSDV), which are not suitable for WSNs. Third, those proposals require either asymmetric key cryptography or complicated symmetric key cryptography, which are expensive on sensor platforms [1].

B. Solutions

Link-layer encryption and authentication by using a global key can protect WSNs against external attackers because they do not know the global key. However, this does not secure against node compromise because the global key can be exposed. A trustful BS can detect spoofed node identities if every node shares a unique key with it, which is studied in SPINS. However, the centralized control can introduce too much communication or management overhead. To counter selective forwarding by malicious nodes, multipath routing can be used to increase the probability of data delivery. To support topology maintenance, authentication is required to protect broadcast of routing information in a local area. Though these methods effectively can prevent external attackers from spoofing, modifying, and replaying information and reduce the impact of selective forwarding, they cannot protect the network from internal malicious nodes efficiently [1].

In the INtrusion-tolerant routing protocol for wireless SEnsor NetworkS (INSENS), the authenticated routing information can be collected by the BS so that it can calculate the routing table for every sensor node. The broadcast information from the BS is authenticated by a one-way hash chain. To prevent DoS attacks, individual nodes are not allowed to broadcast information to the entire network. To increase the tolerance to node compromise, redundant multipath routing is used so that traffic can survive even if some paths are compromised. However, INSENS assumes an application scenario where communications can happen only between sensor nodes and the BS. It does not support in-network processing. An LKH is a key tree structure with source nodes as leafs and a sink node as the root. Each leaf node holds keys along the path from it to the root node. In LKHW, an LKH is established before data are fused. Then the LKH is used to provide encryption and authentication for data fusion. A Secure Routing Protocol for Sensor Networks (SRPSN) is developed in. A hierarchical network is constructed with cluster heads and cluster member nodes. Messages from sensor nodes are routed by cluster heads. To protect data, a preloaded symmetric key is shared between all cluster heads and the base BS.

C. Open issues

For routing protocols, security considerations should be considered at the design stage. Considering specific application scenarios, the network administrator should analyze the possible black holes that may corrupt or disrupt the applications and deploy security countermeasure in advance [1].

A general approach to protect routing protocols is to authenticate the routing information exchanged between nodes. This can effectively prevent an external attacker from injecting false routing information. However, authenticated routing metrics may not be correct in that an internal malicious node can claim false routing metrics without being detected because it has correct keys. Therefore, it is necessary for high layers to verify routing metrics. Some metrics such as residual energy are very difficult to verify because each node cannot know the energy consumption of other nodes. The metrics that have local similarity are easy to verify. For example, most geographical

routing protocols have inherited immunity to false routing information because the nodes that are close to each other should have similar geographical distances to the sink. Considering the frequent node failure under malicious attacks, multipath routing is a promising approach to provide robust and secure data transmission services. Based on the secret sharing technique, a message can be decomposed into many shares, and those shares can be spread into multiple paths and collected by the sink to recover the original message. How many shares each path can be assigned depends on the security or the reliability of that path. In this way, the confidentiality of data can be strengthened because attackers must compromise a certain number of routes to capture a message [1].

However, the integration of security measures into routing protocols can introduce additional overhead. Strong security primitives can ensure a high level of security but may not be acceptable because of the resource constraints of WSNs. How to achieve a trade-off between security and routing overhead is still an open problem [1].

VII. Intrusion Detection And Countermeasures

A. *Node compromise*

Usually, a WSN is managed by an authority that can deploy a secure infrastructure to protect the network. At first, all the secrets deployed are unavailable to attackers. An attacker, without knowing any secrets, can eavesdrop on packets but may not be able to discover the content of the packet because most likely, the packet payload is encrypted. Thus, the attacker is an external attacker with limited attack capabilities. However, a WSN usually is deployed in a hostile environment. An attacker may compromise a sensor node to extract all its keying material. Even if tamper-resistant devices are available for a sensor platform, they still cannot guarantee the perfect security of secrets. Hence, node compromise usually is unavoidable in WSNs.

By compromising one node, an external attacker can become an internal attacker and launch more severe attacks. The attacker can use the compromised node to monitor the network traffic. It is very hard to detect this attack because the attacker follows the normal network protocols without showing an anomaly. The attacker can also use the malicious node to launch various active attacks. This situation poses the demand for compromise-tolerant security design. The network should remain highly secure even when a certain number of nodes are compromised. Therefore, using location information can mitigate the impact of node compromise. To further control the impact of node compromise, public key techniques can be used, because the symmetric key techniques used in cannot totally solve the node compromise problem. Proposed a suite of location-based security mechanisms in which each node holds a private key bound to both its ID and its geographic location. Based on location-based keys, they developed a neighborhood authentication protocol that can successfully localize the impact of compromised nodes to their vicinity. In addition, they demonstrated how location-based keys can act as efficient countermeasures against many notorious attacks against WSNs, such as a Sybil attack, a node replication attack, or a Wormhole attack [1].

B. *Node monitoring*

Through active attacks, an attacker displays many anomalies, which are the indications of a malicious attack. Intrusion detection mechanisms attempt to detect an attack based on those anomalies. Usually, the neighbors of a malicious node are the first entities to learn of the abnormal behaviors. Hence, it is convenient to let each node monitor its neighbors such that intrusion detection mechanisms can be triggered as soon as possible. Khalil, Bagchi and Nita-Rotaru proposed the detection, diagnosis, and isolation of control attacks in sensor networks (DICAS) protocol to detect, diagnose, and isolate malicious nodes. Local monitoring capability is utilized in that a neighbor of both the sender and receiver can oversee the communication behaviors of the receiver. If the receiver has any abnormal behavior on the received packets, it can be detected. If the number of abnormal behaviors is larger than a threshold, the neighbors of the detected malicious node refuse to receive packets from and send packets to it so that the malicious node is isolated from the network. Cumulative observations of anomalies can be used to evaluate the integrity of sensor nodes. In addition to monitoring neighbors, Seshadri et al. proposed a physical layer intrusion detection method called secure code update by attestation (SCUBA) in. SCUBA implements a self-monitoring mechanism by a primitive function called indisputable code execution (ICE). ICE can create a correct execution environment for programs. An attacker can be detected by ICE if it tries to fake the execution environment [1].

C. *Secure base station*

A BS is the gateway of a WSN to the external wired world. A WSN must exchange all information with the external world rushing path records by embedding node list in through a BS. Usually, a BS has more powerful capabilities to perform centralized computation and is more resilient to malicious attacks. Hence, the conventional security schemes for WSNs assume that the BS is always secure. However, there is still a possibility that the BS may become a failure point if attackers are powerful enough to break it. Hence, BS security is also important and requires more consideration.

Deng, Han, and Mishra proposed several methods to protect the BS from malicious attacks. The first method is to deploy multiple BSs to provide tolerance against individual BS failure. The second method tries to hide the identity of the BS. Particularly, a pairwise shared key is used to encrypt packets, including the address field in the packet headers, between two neighboring nodes. Instead of using node IDs in the address field, they use a hash function to construct several anonyms for each node. All the nodes use their anonyms as either source addresses or destination addresses. The pairwise shared keys are generated and distributed by the BS during the topology construction phase. In the third method, the BS is allowed to relocate so that its location is hard to track by attackers. If an attacker wants to attack the BS, it must know where to find it.

D. *Open Issues*

Although many intrusion detection techniques were proposed to detect malicious attacks, most of them target only one specific attack by using different approaches and hardware

assumptions. It is very difficult to integrate those techniques into a uniform hardware platform due to cost and implementation considerations. A promising approach is to choose a set of criteria based on the characteristic analysis of different attacks and establish a simple, uniform intrusion detection framework. Reputation-based detection is good in that it is based on the statistics of anomalies but not specific detection techniques. However, it cannot recognize the kind of attack that is occurring. This is because the criteria for determining malicious behavior has not been addressed. It is beneficial to investigate how to detect a specific attack under the general framework [1].

Identifying an intrusion is a requirement for further active network protection measures. The network administrator utilizes the alert messages from the intrusion detection system to analyze the cause and impact of an attack and enforce some countermeasures. In particular, realizing autonomic computing in a WSN requires two tasks. One is that the network is capable of defending itself against malicious attacks and recovering by self-healing measures. The other is that the network must be able to anticipate potential security problems based on historic log reports and take the required steps to avoid or mitigate them, for example, by automatically upgrading its defense systems. Breaking the physical layer is the first step for adversaries to launch attacks. Most recent security solutions target the protocols of high layers, but not the physical layer. However, it would be beneficial if the physical layer can provide resilience to malicious attacks. Spread spectrum can reduce the impact of radio jamming by hiding a signal behind the environmental noise. Improved tamper-resistant techniques can reduce the probability of node compromise such that the possibility of internal attacks also can be reduced. Meanwhile, the changes in physical characteristics of sensor nodes, such as code length, may be used to detect malicious tampering [1].

VIII. Attacks In Wireless Sensor Networks

Attacks against wireless sensor networks could be broadly considered from two different levels of views. One is the attack against the security mechanisms and another is against the basic mechanisms (like routing mechanisms). [2]
Attacks can therefore be classified from different points of view.

A. Attacks on Information in transit

In a sensor network, sensors monitor the changes of specific parameters or values and report to the sink according to the requirement. While sending the report, the information in transit may be altered, spoofed, replayed again or vanished. As wireless communication is vulnerable to eavesdropping, any attacker can monitor the traffic flow and get into action to interrupt, intercept, modify or fabricate [8] packets thus, provide wrong information to the base stations or sinks. As sensor nodes typically have short range of transmission and scarce resource, an attacker with high processing power and larger communication range could attack several sensors at the same time to modify the actual information during transmission.

B. Sybil Attack

In many cases, the sensors in a wireless sensor network might need to work together to accomplish a task, hence they can use distribution of subtasks and redundancy of information. In such a situation, a node can pretend to be more than one node using the identities of other legitimate nodes (Figure 1). This type of attack where a node forges the identities of more than one node is the Sybil attack [9], [10]. Sybil attack tries to degrade the integrity of data, security and resource utilization that the distributed algorithm attempts to achieve. Sybil attack can be performed for attacking the distributed storage, routing mechanism, data aggregation, voting, fair resource allocation and misbehavior detection [10]. Basically, any peer-to-peer network (especially wireless ad hoc networks) is vulnerable to sybil attack. However, as WSNs can have some sort of base stations or gateways, this attack could be prevented using efficient protocols. Douceur [9] showed that, without a logically centralized authority, sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities. However, detection of sybil nodes in a network is not so easy. Newsome et. al. [10] used radio resource testing to detect the presence of sybil node(s) in sensor network and showed that the probability to detect the existence of a sybil node is:

$$Pr(detection) = 1 - \left(1 - \sum_{allS,M,G} \frac{\binom{s}{S}\binom{m}{M}\binom{g}{G}}{\binom{n}{c}} \frac{S-(m-M)}{c}\right)^{r}$$

Where, n is the number of nodes in a neighbor set, s is the number of sybil nodes, m malicious nodes, g number of good nodes, c is the number of nodes that can be tested at a time by a node, of which S are sybil nodes, M are malicious (faulty) nodes, G are good (correct) nodes and r is the number of rounds to iterate the test.
As mentioned in [1], to detect the Sybil attack, two methods were discussed in [11]. One method is radio resource testing in which each node assigns a unique channel to each of its neighbors, including fake neighbors, and tests whether its neighbors can communicate with it through the assigned channels. Because the radio of a sensor platform is usually incapable of simultaneously sending or receiving on more than one channel, the failure of communication through one channel may be a sign of the Sybil attack. The other method is to use the ID-based symmetric keys. For example, each sensor node is preloaded with a set of keys that are selected from a global key pool by its node ID. The ID of a suspected node is challenged by a set of validating nodes based on the keys shared between the suspected node and the validating nodes. Several other methods were suggested in [11], including registration, position verification, and code attestation. Moreover, ID-based public keys also can defeat the Sybil attack because both the ID and location information were taken into the generation of key material during the initialization phase, hence multiple identities need multiple keys, and this is impossible for a malicious node to achieve.

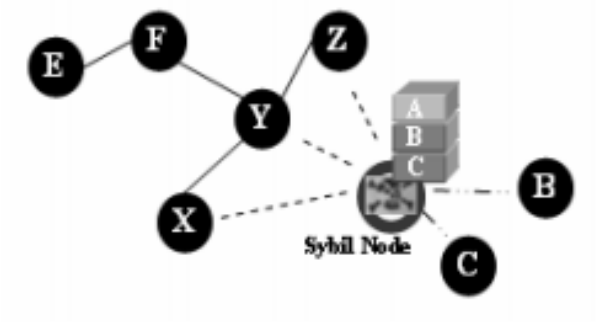

Figure 1: Sybil Attack

C. Blackhole/Sinkhole attack

In this attack, a malicious node acts as a blackhole [12] to attract all the traffic in the sensor network. Especially in a flooding based protocol, the attacker listens to requests for routes then replies to the target nodes that it contains the high quality or shortest path to the base station. Once the malicious device has been able to insert itself between the communicating nodes (for example, sink and sensor node), it is able to do anything with the packets passing between them. In fact, this attack can affect even the nodes those are considerably far from the base stations. Figure 2 shows the conceptual view of a blackhole/sinkhole attack.

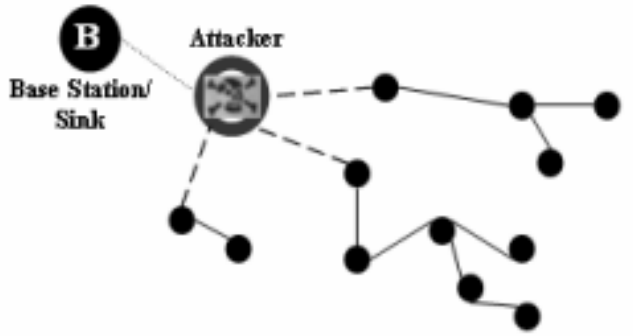

Figure 2: Conceptual view of Blackhole Attack

D. Hello Flood Attack

Hello Flood Attack is introduced in [13]. This attack uses HELLO packets as a weapon to convince the sensors in WSN. In this sort of attack an attacker with a high radio transmission (Termed as a laptop-class attacker in [12]) range and processing power sends HELLO packets to a number of sensor nodes which are dispersed in a large area within a WSN. The sensors are thus persuaded that the adversary is their neighbor. As a consequence, while sending the information to the base station, the victim nodes try to go through the attacker as they know that it is their neighbor and are ultimately spoofed by the attacker.

E. Wormhole Attack

Wormhole attack [14] is a critical attack in which the attacker records the packets (or bits) at one location in the network and tunnels those to another location. The tunneling or retransmitting of bits could be done selectively. Wormhole attack is a significant threat to wireless sensor networks, because; this sort of attack does not require compromising a sensor in the network rather, it could be performed even at the initial phase when the sensors start to discover the neighboring information.

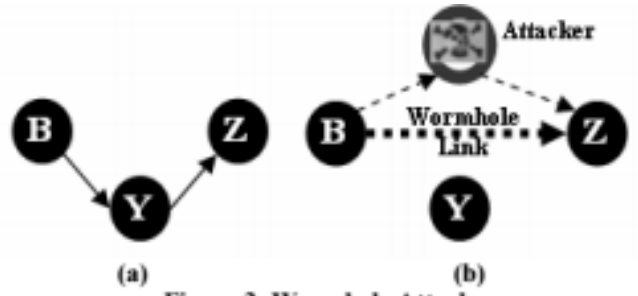

Figure 3: Wormhole Attack

Figure 3 (a and b) shows a situation where a wormhole attack takes place. When a node B (for example, the base station or any other sensor) broadcasts the routing request packet, the attacker receives this packet and replays it in its neighborhood. Each neighboring node receiving this replayed packet will consider itself to be in the range of Node B, and will mark this node as its parent. Hence, even if the victim nodes are multi hop apart from B, attacker in this case convinces them that B is only a single hop away from them, thus creates a wormhole.

To detect the Wormhole attack, Hu et al. proposed to use packet leashes, where location or timing information is embedded in packets, to limit the maximum range over which packets can be tunneled. They require that each node either knows its location or has a tightly synchronized clock so that this information can be used to calculate the maximum distance that a relayed packet could travel. Directional antennas [15] also were used to defend against the Wormhole attack, where some direction information is used to detect the replayed packets. However, these defenses target ad hoc networks and require expensive hardware devices, which may be infeasible for most resource-constrained sensor networks. Wang and Bhargava [16] proposed to use centralized computing to detect the Wormhole attack in sensor networks, in which a controller collects the location information for all nodes to reconstruct the network topology such that any topological distortion can be visualized.

However, the visualization approach incurs too much communication overhead, especially when malicious nodes move around in the entire network because each location change of the Wormhole triggers a new round of execution of the topology reconstruction algorithm. Location-based keys also can effectively address the Wormhole attack because each packet is authenticated by the location-based key.

F. Denial-of-service attacks in WSN

Denial of Service (DoS) is a result of unintentional failure of nodes or malicious action. Simplest DoS attack tries to exhaust the resources available to the victim nodes, by sending

extra unnecessary packets and thus preventing authentic network users from accessing services or resource to which they are entitled [7]. DoS attack is meant not only for the adversary's attempt to subvert, disrupt, or destroy a network, but also for any event that diminishes a network's capability to provide a service [7]. At physical layer the DoS attacks could be jamming and tampering, at link layer, collision, exhaustion, unfairness, at network layer, neglect and greed, homing, misdirection, black holes and at transport layer this attack could be performed by malicious flooding and de-synchronization [4]. For example, in case of jamming an adversary attempts to disrupt the network's operation by broadcasting a high-energy signal. If the transmission is powerful enough, the entire system's communication could be jammed. A more sophisticated attacks are also possible; the adversary might inhibit communication by violating the 802.11 medium access control (MAC) protocol by, say, transmitting while a neighbor is also transmitting or by continuously requesting channel access with a request-to-send signal.

There are two types of DoS attacks:
• Passive attacks: Selfish nodes use the network but do not cooperate, saving battery life for their own communications: they do not intend to directly damage other nodes.
• Active attacks: Malicious nodes damage other nodes by causing network outage by partitioning while saving battery life is not a priority. DoS attacks can happen in multiple WSN protocol layers.
Attempts to add DoS resistance of existing often focus on cryptographic-authentication mechanisms. Limited resources make digital-signature schemes impractical authentication in sensor networks poses serious complications [5].

Feihu et al. [5] highlight four mechanisms that could be helpful to overcome DoS attacks in WSN:
• Watchdog scheme: Based on the above text, we can see that a necessary operation to overcome DoS attacks is to identify and circumvent the misbehaving nodes. Watchdog scheme attempts to achieve this purpose through the using of two concepts: watchdog and path rater.
• Rating scheme: Watchdog scheme was further investigated and extended to rating scheme. In rating scheme the neighbors of any single node collaborate in rating the node, according to how well the node execute the functions requested from it.
• Virtual currency: This scheme conceptualized the motivation for nodes not to be selfish as nuglets, a sort of virtual currency (also called nuglets)
• Route DoS prevention: This scheme attempts to prevent DoS in the routing layer through the cooperation of multiple nodes. In the authors introduce a mechanism to assure routing security, fairness and robustness targeted to mobile ad hoc networks. There can be levels of protection as a negotiable metric in route discovery. In this way, a pair of nodes establishes a certain application-specific level of protection before any security-sensitive traffic begins.

IX. Current Research Challenges

The deployment and several other challenges that are faced in wireless sensor networks make security of these systems more challenging than conventional network. However, several properties of sensor networks may help address the challenge of building secure networks. First, we have the opportunity to architect security solutions into these systems from the outset, since they are still in their early design and research stages. Second, many applications are likely to involve the deployment of sensor networks under a single administrative domain, simplifying the threat model. Third, it may be possible to exploit redundancy, scale, and the physical characteristics of the environment in the solutions. If we build sensor networks so they continue operating even if some fraction of their sensors is compromised, we have an opportunity to use redundant sensors to resist further attack. [4]

Many other problems also need further research. One is how to secure wireless communication links against eavesdropping, tampering, traffic analysis, and denial of service. Others involve resource constraints. Ongoing directions include asymmetric protocols where most of the computational burden falls on the base station and on public-key cryptosystems efficient on low-end devices. Finally, finding ways to tolerate the lack of physical security, perhaps through redundancy or knowledge about the physical environment, will remain a continuing overall challenge. We are optimistic that much progress will be made on all of them. [4]

X. Acknowledgment

The authors are grateful to all Wireless Security Papers that are attributed as references in the References section. This paper contains comprehensive study of papers on issue of security in Wireless Sensor Networks, cited from various contributions from the authors mentioned below. The authors would also like to express their gratitude to the Course Instructor, Ms. Divya Lohani for giving us this opportunity to write this Research Paper on "Issue of Security in Wireless Sensor and Networks".

XI. Conclusion

Security is becoming a major concern for WSN protocol designers because of the wide security-critical applications of WSNs. In this article, we discussed general security problems in WSNs and described corresponding solutions. We also discussed various attacks and their existing solutions, including DoS attacks.

However, there are still many open issues. On the one hand, WSNs are still under development, and many of the protocols designed for WSNs fail to take security into consideration. On the other hand, the salient features of WSNs make it very challenging to design strong security protocols while still maintaining low overheads in terms of computation and power. Hence, wireless security for WSNs is still a very fruitful research area to be explored.

XII. REFERENCES

[1] Zhou, Yun, Yuguang Fang, and Yanchao Zhang. "Securing wireless sensor networks: a survey." IEEE Communications Surveys & Tutorials 10, no. 3 (2008).

[2] Pathan, A.S.K., Lee, H.W. and Hong, C.S., 2006, February. Security in wireless sensor networks: issues and challenges. In Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference (Vol. 2, pp. 6-pp). IEEE.

[3] Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V. and Culler, D.E., 2002. SPINS: Security protocols for sensor networks. Wireless networks, 8(5), pp.521-534.

[4] Perrig, A., Stankovic, J. and Wagner, D., 2004. Security in wireless sensor networks. Communications of the ACM, 47(6), pp.53-57.

[5] Hu, F., Ziobro, J., Tillett, J. and Sharma, N.K., 2004. Secure wireless sensor networks: Problems and solutions. Rochester Institute of Technology, Rochester, New York, USA.

[6] Xiaomei, Y. and Ke, M., 2016, July. Evolution of wireless sensor network security. In World Automation Congress (WAC), 2016 (pp. 1-5). IEEE.

[7] CHELLI, K., 2015. Security Issues in Wireless Sensor Networks: Attacks and Countermeasures. In Proceedings of the World Congress on Engineering (Vol. 1, pp. 1-3).

XIII. BIBLIOGRAPHY

[8] Pfleeger, C. P. and Pfleeger, S. L., “Security in Computing”, 3rd edition, Prentice Hall 2003.

[9] Douceur, J. “The Sybil Attack”, 1st International Workshop on Peer-to-Peer Systems (2002).

[10] Newsome, J., Shi, E., Song, D, and Perrig, A, “The sybil attack in sensor networks: analysis & defenses”, Proc. of the third international symposium on Information processing in sensor networks, ACM, 2004, pp. 259 – 268.

[11] J. Newsome et al., “The Sybil Attack in Sensor Networks: Analysis and Defenses,” Proc. 3rd IEEE Int’l. Symp. Information Processing in Sensor Networks (IPSN’04), Berkeley, CA, Apr. 2004.

[12] Culpepper, B.J. and Tseng, H.C., “Sinkhole intrusion indicators in DSR MANETs”, Proc. First International Conference on Broad band Networks, 2004, pp. 681 – 688.

[13] Karlof, C. and Wagner, D., “Secure routing in wireless sensor networks: Attacks and countermeasures”, Elsevier's Ad Hoc Network Journal, Special Issue on Sensor Network Applications and Protocols, September 2003, pp. 293-315.

[14] Hu, Y.-C., Perrig, A., and Johnson, D.B., “Packet leashes: a defense against wormhole attacks in wireless networks”, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE INFOCOM 2003, Vol. 3, 30 March-3 April 2003, pp. 1976 – 1986.

[15] L. Hu and D. Evans, “Using Directional Antennas to Prevent Wormhole Attacks,” Proc. 11th Annual Network and Distributed System Security Symp. (NDSS’04), San Diego, CA, Feb. 2004.

[16] W. Wang and B. Bhargava, “Visualization of Wormholes in Sensor Networks,” Proc. 2004 ACM Wksp. Wireless Security (Wise’04), Philadelphia, PA, Oct. 2004.

[17] FIPS PUB 46-2, “Data Encryption Standard (DES),” Dec. 1993.

[18] IETF RFC 2040, “The rc5, rc5-cbc, rc5-cbc-pad, and rc5-cts algorithms,” Oct. 1996.

[19] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Commun. ACM, vol. 21, no. 2, Feb. 1978, pp. 120–26.

[20] S. Basagni et al., “Secure Pebblenets,” Proc. 2nd ACM Int’l. Symp. Mobile Ad Hoc Networking and Computing (Mobi-hoc’01), Long Beach, CA, 2001.

[21] R. Watro et al., “TinyPK: Securing Sensor Networks with Pub-lic Key Technology,” Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN’04), Washington, DC, Oct. 2004.

[22] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Trans. Info. Theory, vol. IT-22, no. 6, 1976, pp. 644–54.

[23] Q. Huang et al., “Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks,” Proc. 2nd ACM Int’l. Conf. Wireless Sensor Networks and Applications (WSNA’03), San Diego, CA, Sept. 2003.

[24] Y. Zhou, Y. Zhang, and Y. Fang, “Access Control in Wireless Sensor Networks,” Elsevier Ad Hoc Networks Journal, Special Issue on Security in Ad Hoc and Sensor Networks, vol. 5, 2007, pp. 3–13.

[25] S. Vanstone, “Responses to NIST’s Proposal,” Commun. ACM, vol. 35, July 1992, pp. 50–52.

[26] Chris Karlof, Naveen Sastry, and David Wagner, “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks,”Proc. 2nd Int’l. Conf. Embedded Networked Sensor Systems (SenSys’04), Baltimore, MD, Nov. 2004.

[27] National Security Agency, Skipjack and KEA algorithm specifications. May 1998.

[28] M. Bohge and W. Trappe, “An Authentication Framework for Hierarchical Ad Hoc Sensor Networks,” Proc. 2nd ACM Workshop on Wireless Security (WiSe’03), San Diego, CA, 2003.

[29] A. Perrig et al., “TESLA: Multicast Source Authentication Transform Introduction,” IETF working draft, draft-ietf-msec-26 IEEE Communications Surveys & Tutorials • 3rd Quarter 2008 tesla-intro-01.txt.

[30] H. Chan et al., “On the Distribution and Revocation of Cryptographic Keys in Sensor Networks,” IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 3, July-Sept. 2005, pp. 233–47.

[31] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” Proc. 2003 IEEE Symp. Security and Privacy, May 2003,pp.197–213.