Survey on Wireless Sensor Networks with security Issues

G.Suhasini,email: suhasinigadala@rediffmail.com
Department of computer science and engineering
Ganapathi college of Engineering, Warangal, INDIA

Abstract*— *security issues in wireless sensor networks plays key role. We have to provide adequate security for wireless sensor networks. This work deals with some security issues over wireless sensor networks (WSNs). A survey of recent trends in general security requirements, typical security treats, intrusion detection system, key distribution schemes and target localization is presented. In order to facilitate applications that require packet delivery from one or more senders to multiple receivers, provisioning security in group communications is pointed out as a critical and challenging goal. Presented issues are crucial for future implementation of WSN.

Keywords*—Intrusion Detection System, Key Distribution Schemes, Security, Target Localization, Wireless Sensor Networks.*

I. INTRODUCTION

Fundamental goals for Wireless Sensor Networks (WSNs) is to collect information from the physical world. Although a number of proposals have been reported concerning security in WSNs, provisioning security remains critical and challenging task. WSNs have attracted much attention due to its great potential to be used in various applications. Comparing to existing infrastructure – based networks, wireless sensor networks can virtually work in any environment, especially those where wired connections are not possible. Unlike conventional networks supporting mostly point-to-point or point-to-multipoint data forwarding, WSNs are often deployed to sense, process and disseminate information of targeted physical environments. In general, WSNs consist of battery-operated sensor devices with computing, data processing, and communicating components. The ways the sensors are deployed can either be in a controlled environment where monitoring and surveillance are critical. In the uncontrolled environments, security for sensor networks becomes extremely important. Sensors are usually deployed in large numbers of sensor date, which are often impractical to gather from the individual sensors, particularly from the energy consummation point of view. Thus data fusion (or aggregation) offers a key strategy to reduce energy consumption. Performing data fusion in WSNs can be largely attributed to two reasons. On one hand, the user may be interested only in the aggregated results on the sensor data (for example, only the average measure of the relevant parameters may be of interest).

On the other hand, data from sensors in close proximity may be highly correlated, and data fusion can effectively reduce redundancy and hence network load [1]. Data fusion operation has been incorporated into a wide range of existing WSN design [2]. Although diverse work exists on data fusion, a fundamental supporting mechanism is the data routing which dictates when and where data streams will meet and hence how fusion will be performed. Manuscript received December 10, 2008: Revised version received, 2008. Z. S. Bojkovic is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, 11 000 Belgrade, Serbia (phone: + 381 11 3091-217; e-mail: z.bojkovic@yahoo.com). B. M. Bakmaz, is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, 11 000 Belgrade, Serbia (e-mail: b.bakmaz@sf.bg.ac.yu). M. R. Bakmaz is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, 11 000 Belgrade, Serbia (e-mail: bakmaz@sf.bg.ac.yu). Apart from the wireless medium, the primary challenges for sensor networks stem from two facts. First, sensors are extremely resource constrained. Second, in many applications sensor nodes will be randomly deployed. This randomness raises the issue of dimensioning the network. Scattering too few nodes may result in lack of coverage of the sensor field and a disconnected network. On the other hand, scattering too many nodes may result in an inefficient network due to increased medium access control (MAC) collision and interference. WSNs are exploited to be deployed for a long period, and the nodes are likely to need software updates during their lifetime in order to support new requirements. In many cases the nodes will be inaccessible or too numerous to be physically accessed. This drives the need for software updates support. This paper is outlined as follows. We first introduce the general security requirements in WSNs. We summarize typical security treats and adequate defense techniques. Key distribution schemes with some technical aspects are introduced in the next session. Then, we deal with target localization problem and security in group communications over WSNs. Finally, the importance for updating software is pointed out. Proposals for future work conclude the presentation.

II. GENERAL SECURITY REQUIREMENTS IN WIRELESS SENSOR NETWORKS

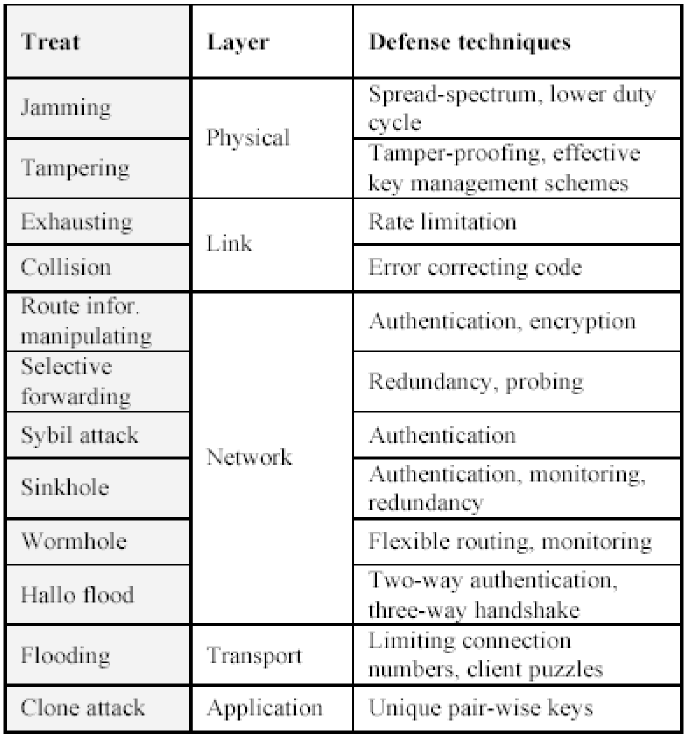
Because of the nature of wireless communications, resource limitation on sensor nodes, size and density of the networks, unknown topology prior to deployment, and high risk of physical attacks to unattended sensors, it is a challenge to provide security in WSNs. The ultimate security requirement is to provide confidentiality, integrity, authenticity, and availability of all messages in the presence of resourceful adversaries.

To provide secure communications for the WSNs, all messages have to be encrypted and authenticated. Security attacks on information flow can be widespread. Modification of information is possible because of the nature of the wireless channels and uncontrolled node environments. An opponent can use natural impairments to modify information and also render the information unavailable. Security requirements in WSNs are similar to those of wireless ad hoc networks due to their similarities [3]. WSNs have the general security requirements of availability, integrity, authentication, confidentiality and non-repudiation. These security requirements can be provided by distribution mechanism with the requirements of scalability, efficiency key connectivity and resilience. Scalability is the ability to support large sensor nodes in the networks. Key distribution mechanism must support large network, and must be flexible against substantial increase in the size of the network even after deployment. Efficiency is the consideration of storage processing and communications limitations on sensor nodes. Key connectivity is the probability that two or more sensor nodes store the same key or keying material. Enough key connectivity must be provided for a WSN to perform its intended functionality. Resilience is about the resistance against node capture. Compromise of security credentials, which are stored on a sensor node or exchanged over radio links, should not reveal information about security of any other links in a WSN. Higher resilience means lower number of compromised links.

III. TYPICAL SECURITY TREATS AND DEFENSE TECHNIQUES IN WIRELESS SENSOR NETWORKS

Communications over wireless channels are, by nature, insecure and easily susceptible to various kinds of treats. A large-scale sensor network consists of huge number of sensor nodes and may be dispersed over a wide area. Typical sensor nodes are small with limited communication and computing capabilities. These small sensor nodes are pervious to several key types of treats. For a large-scale sensor network, it is impractical to monitor and protect each individual sensor from physical or logical attack. Treats on sensor networks can be classified into attacks on physical, link (MAC), network, transportation, and application layers [4]. Treats can also be classified based on the capability of the possible attacker, such as sensor-level and laptop-level. A powerful laptop-level adversary can do much more harm to a network than a malicious sensor node, since it has much better power supply, as well as larger computation and communication capabilities than a sensor node. Treats can also be classified into outside and inside treats. An outside attacker has no access to most cryptographic materials in sensor networks, while an inside attacker may have partial key materials and the trust of other sensor nodes. Inside attacks are much harder to detect and defend against. Typical treats and adequate defense techniques in WSNs are summarized as in Table I. Table I. Typical treats in WSNs

Table I. Typical treats in WSNs



Treat	Layer	Defense techniques
Jamming	Physical	Spread-spectrum, lower duty cycle
Tampering		Tamper-proofing, effective key management schemes
Exhausting	Link	Rate limitation
Collision		Error correcting code
Route infor. manipulating	Network	Authentication, encryption
Selective forwarding		Redundancy, probing
Sybil attack		Authentication
Sinkhole		Authentication, monitoring, redundancy
Wormhole		Flexible routing, monitoring
Hallo flood		Two-way authentication, three-way handshake
Flooding	Transport	Limiting connection numbers, client puzzles
Clone attack	Application	Unique pair-wise keys

IV. KEY DISTRIBUTION SCHEMES

The three simplest keying models that are used to compare the different relationships between the WSN security and operational requirements are [5]: • network keying, • pair-wise keying, and • group keying. The network keying model has inherent advantages over the other two schemes. It is simple, easy to manage, and uses very small amount of resources. Network keying also allows easy collaboration of nodes since neighboring nodes can read and interpret each other's data, satisfying the self-organization and accessibility requirements. It is also excellent in terms of scalability and flexibility because there is only one key for the entire network, and it does not change with the addition of nodes. However, an unacceptable drawback in robustness exists. Suppose one node is compromised, and the network-wide key is exposed. With this key, an adversary can eavesdrop on all messages in the network and even inject forged messages into the network, possibly disrupting the proper operation of the network. At the other extreme, the pair-wise keying model employs N–1 keys in each node, where N is the size of the network. Although this model provides the ultimate in robustness against node capture because the compromise of one node does not compromise any other node. It fails to satisfy the scalability requirement because the storage cost grows rapidly with network size. In the case of several thousand nodes, the number of keys each node must maintain becomes unmanageable. Consider the storage of N–1 keys per node.

The total number of distinguishable keys in the network is N(N–1)/2, which grows at a rate of N2. This is not maintainable when N is a large value. Another issue with the pair-wise keying model is that it is difficult to add new nodes to the network, affecting the flexibility requirement. When a new node is added, every node must obtain a new key to communicate with it. This is a resource-intensive process that uses much more precious energy when compared with the simple preloading of a network-wide key as in the previous model. Similarly, key revocation and key refreshing suffer from the same scalability problem. Additionally, the accessibility requirement is in jeopardy as nodes cannot passively monitor event signals. Lastly, in the case of pair-wise key distribution schemes, self-organization comes into question, because they tackle the scalability problem by reducing the number of shared keys, resulting in some nodes being unable to communicate with others and compromising the self-healing and self-organizing abilities of the network. The group keying scheme combines the features of both network and pair-wise keying schemes. Within a group of nodes that form a cluster, communications are performed using a single, shared key similar to network keying.

However, communications between groups employ a different key between each pair of groups in a manner identical to the pair-wise keying scheme. Thus, for a group of nodes, the accessibility requirement is satisfied because data aggregation can occur with no additional cost while some degree of robustness is maintained. When one of the nodes is compromised, the worst-case scenario is the compromise of the entire cluster that it belongs to, which is considerably more isolated than the entire network. In terms of scalability, an acceptable trade off is possible in this scheme, because the number of keys increases with the number of groups, not with the size of the network. However, the problem with this scheme is that it is difficult to set up and also, the formation of the groups is a very application dependent process. To efficiently distribute the keys, a keying scheme would require group formation information. Security solutions depend on the use of strong and efficient key distribution mechanisms in uncontrolled environments. To implement a fundamental security service pair-wise key establishment should be used, enabling secure communications among the sensor nodes using cryptographic techniques.

Due to resource constraints on sensor nodes, it is not feasible for sensor to use traditional pair-wise key establishment techniques such as public key cryptography and key distribution center. In the case where sensor nodes should use pre-distributed keys directly, or use keying materials to dynamically generate pair-wise keys, the main challenges is to find an efficient way of distributing keys and keying materials to sensor nodes prior to deployment. Different pair-wise key distribution schemes have been developed for peer-to-peer WSNs and hierarchical WSNs [6]. In peer-to-peer WSNs, there is no fixed infrastructure, and network topology is not known prior to deployment. As for sensor nodes, they are usually randomly scattered all over the target area. Once they are deployed, each sensor node scans its radio coverage area, to figure out its neighbors. In the case of hierarchical WSNs, there exists a hierarchy among the nodes based on their capabilities: base stations and sensor nodes [7]. Comparing to the sensor nodes in terms of transmission rate, data processing capabilities, storage capacity and temper-resistance, the base stations can be much more powerful. Base stations can form the backbone of the sensor network, while sensor nodes can be deployed around single or multi-hop neighborhood of the base stations. In general, the base stations are also the key distribution centers in the sensor networks. With the advances in antenna technologies like multiple-input-multiple-output (MIMO) systems, directional antennas and cooperative communications, the heterogeneity in terms of transmission rate in wireless sensor nodes has become a reality. Such heterogeneity can improve network performance and network lifetime without significantly increasing the cost.

For hierarchy wireless sensor networks, base stations act like key distribution centers. Base stations may share a distinct pair-wise master-key with each sensor nodes within a cluster. These master-keys can then be used to establish other security keys. In hierarchical WSNs, pair-wise keys are required for the communications between a base station and sensor node, and between two sensor nodes. The requirement can be easily resolved if a base station shares a distinct pair-wise master key with each sensor node. Example 1 As an example consider now a typical heterogeneous WSN that is established to collect data in a distributed scenario. A sensor node should submit its observation to a sink node (or some nodes depending on the configuration of the network) through the network in a hop-by-hop manner as shown in Fig. 1.

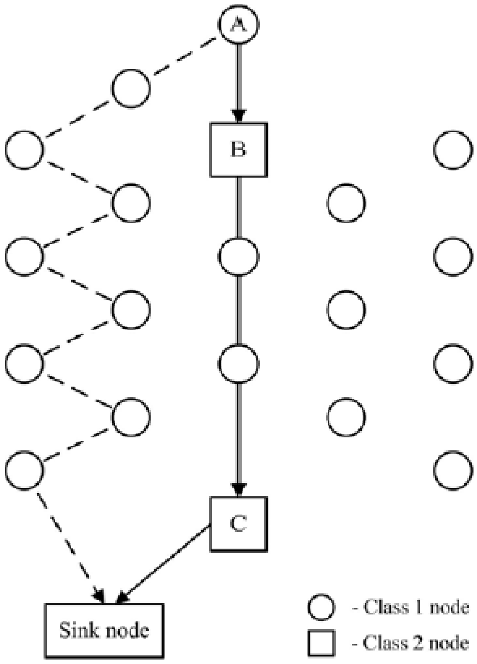


Fig. 1. An example of WSN to collect data in a distributed scenario.

As it can be seen, there exist two types of sensor nodes: higher class nodes and lower class nodes. Connectivity in this heterogeneous WSN between a low class node and a high class node will be more important than the connectivity between two low class nodes. Example 2 As a second example let us take now two key distribution schemes, where there are only two classes of the heterogeneous sensor nodes (I = 2). The first scheme is a key-pool based key distribution scheme. It is based on the random key distribution and polynomial based key pre-distribution protocol. A pool of randomly generated bipartite polynomials is used to establish pair-wise keys between sensor nodes with the consideration of I classes of heterogeneity among the wireless sensor nodes. Compared to the key-pool based scheme, the polynomial-pool based scheme may be more resilient and require less memory storage as well as communication overhead. For both schemes, we can denote C1 as the class of the less powerful sensor nodes, and denote C2 the class of the more powerful sensor nodes. A C2 node is in the neighborhood of a C1 node, if this node can directly receive a broadcast message from C2 node. It means that C1 node can receive the key (polynomial) pool information of the C2 node without the relay of other sensor nodes. Key management scheme in WSN is shown in Fig. 2. Node A is a C1 node, while nodes X, Y, Z are C2 nodes.

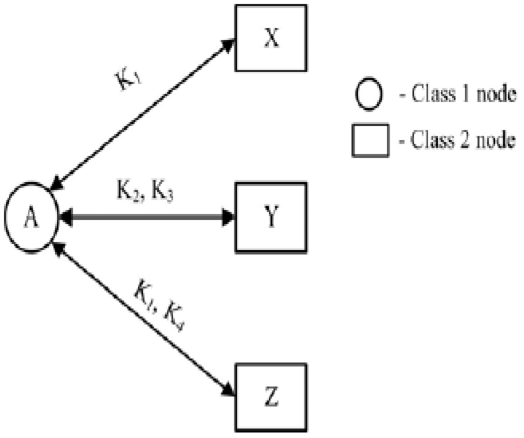


Fig. 2. An example of key management scheme in WSN.

An example of key management scheme in WSN. In this example, nodes X, Y, and Z are the only C2 neighbor nodes of node A. In addition, node A shares key K1 with node X, K2 and K3 with node Y, while K1 and K4 with node Z, respectively. Node A is connected if $q \leq 4$. In such a case, if node A wants to submit new information to the sink node, it can first randomly selected a key from K1 to K4. Then, it can randomly select a neighbor node that shares the same key with it. In this way, the communication is more resilient, while maintaining the connectivity.

V. INTRUSION DETECTION SYSTEM

There are some mechanisms that try to detect abnormal situations caused by malicious nodes, either by analyzing the behavior of the network, or by using protocol-specific technologies such as for example, automate theory. An intrusion detection system (IDS) is an interesting, underdeveloped service, useful for scenarios where there is a possibility for a node being subverted and controlled by an adversary. The major task of IDS is to monitor networks and systems to detect eventual intrusions in the network, alert users after specific intrusions have been detected and finally, if possible reconfigure the network and mark the root of the problem as malicious [8]. An IDS protects data integrity and manages system availability during an intrusion. With self-regulating protocols, it must deal with challenges related to resource-constrained fully mobile, self-configuring wireless networks with varying resources and limited bandwidth. This system should be able to detect intrusion by monitoring unusual activities in the system and comparing them to a user's profile and evolving trends.

The distributed and cooperative nature of nodes makes it possible for a malicious node to exploit the weakest node by launching an attack through it. This inherent vulnerability can disable the whole network cluster and further compromise security by impersonating, message contamination or acting as a malicious router. Various routing techniques have been researched in the area of trying to resist attacks [4]. Intrusion can be through of as a pattern of an observed sequence. Its detection is similar to an immune system that identifies and eliminates anomalies by measuring deviations from normal process using distributed identifiers over the system with an identifiable and adaptable relationship.

The objective of the modeling is to identify the intrusion while reducing the number of false positives. An instantaneous deviation from a normal profile can be constructed as an intrusion due to a monetary change in the system environment. Aside from the detection of abnormal events, there are other aspects in the development of IDS that must be solved (for example the exact location of the detection agents and their tasks). On the other hand, when considering the existence of a fully functional IDS, there is a need for filtering the information provided by the system to detect malicious nodes and distinguish between possible errors and attacks launched against the network. There are two main types of approaches to IDS [9]: • Misuse (signature-based) detection, where known security attack signatures are kept and matched against the monitored system. This type of detection can accurately detect known attacks, but it is unable to detect any new attacks that emerge in the system. • Anomaly detection, where a normal profile of the monitored data is established, and then anomalies are identified as measurements that deviate from default profiles. Because of that, anomaly detection is capable of detecting new types of security risks. A problem with this approach is the high level of false alarms. Due to, reducing level of false alarms while still being responsive to detecting security risks, is major issue for intrusion detection. Functional IDS have to fulfill multiple objectives related to accurate intrusion detection using various ingredients like: • Intrusion checkpoints represent the observable states of the IDS and analyze the sensor activity that predicts the transition form normal to intrusion state. • Creation of an activity profile that identifies abnormal activity of the observable states by measuring the sensor deviation from normal behavior. • Concept drift that measures the change in user behavior over a period of time. • Control loop which adopts the trigger based on the weighted sum of proportional, average, and derivative sensor measurements over derivative and integral time window. A hidden Markov model (HMM) correlates observations like parameters changes, fault frequency etc., to predict hidden state in the system design [10]. Observation points are optimized using an acceptable set of system-wide intrusion checkpoints, while hidden states are created using explicit knowledge of probabilistic relationships with these observations.

For modeling a large number of temporal sequences, HMM acts as an excellent alternative, as it has been widely used for pattern matching in speech recognition and image identification. We can say that HMM-based approaches correlate the system observation (usage and activity profile) and state transitions to predict the most probable so called intrusion state sequence. HMM is a stochastic model of discrete events and a variation of the Markov chain. It consists of a set of discrete states and matrix A = || aij || of state transition probabilities. The states of the HMM can only be inferred from the observed symbols. Hence, it is called hidden, generally speaking, HMM modeling schemes consist of observed states (intrusion, checkpoints), hidden (intrusion) states, and HMM (activity) profiles. HMM training using initial data and continuous re-estimating creates a profile that consists of transition probabilities and observation symbol probabilities. It is important to point out that HMM modeling involves the following steps: 1) measuring observed states, 2) estimating an instantaneous observation probability matrix, 3) estimating hidden states, and 4) estimating a hidden state transition probability matrix. Observed states are analytically or logically derived from the intrusion indicators.

These indicators are test points spread all over the system representing competing risks derived analytically or logically using intrusion check indicators. Instantaneous observation probability matrix indicates the probability of an observation, given a hidden state p(Si/Oi). Here S represents hidden states, while O performs visible states. The density function can be estimated using an explicit parametric model or implicitly from data via nonparametric methods. For example, an explicit parametric model is multivariate Gaussian, while nonparametric methods refer to multivariate kernel density emission. In estimating hidden states, we use clustering the homogeneous behavior of single or multiple components together. These states are indicative of various intrusion activities. They need to be identified by the administrator. Hidden state S = {S1, S2, ... , SN-1, SN} are the set of states that are not visible, but each randomly generates a mixture of the M observations or visible states O. The probability of the subsequent state depends only on the previous state. Estimating a hidden state transition probability matrix is carried out using prior knowledge or random data. This prior knowledge and long-term temporal characteristics are an approximate probability or state components transitioning from one intrusion state to another.

VI. TARGET LOCALIZATION PROBLEM

Sensor locations play a critical role in many sensor network applications, such as environmental monitoring and target tracking. Fundamental techniques developed for wireless sensor networks also require sensor location information, such as routing protocols that make routing decisions based on node locations. Location discovery/estimation protocols, also called localization protocols, use some special nodes, called beacon nodes which are assumed to known their own locations. These protocols work in two steps. First step: Non beacon nodes receive radio signals called reference messages from the beacon nodes. A reference message includes the location of the beacon node. Second step: The non beacon nodes make certain measurements, for example distance between the beacon and non beacon nodes.

The measurements are based on features of the reference messages like received signal strength indicator and time difference of arrival. Without protection, an attacker may easily mislead the location estimation at sensor nodes and subvert the normal operation of sensor networks. An attacker may provide incorrect location references by replaying the beacon packets intercepted in different locations. Also, an attacker may compromise a beacon mode and distribute malicious location references by lying about the location or manipulating the beacon signals. In either case, non beacon nodes will determine their locations incorrectly. From the point of view of coverage and connectivity, the dimensioning problem has been intensely studied in recent years [11]. The most commonly used problem in coverage problems is the disk model, which assumes that sensing region for a sensor is a circular region centered at it. A point is said to be covered by the sensor if it is within its sensing region. The disk model has certain limitations in describing haw well the field is covered. When several nearby sensors are monitoring an event at the same time, the estimation error can be reduced through cooperative signal processing [12]. Although each single sensor may not be able to provide precise information about the event, the information about the environment can still be reconstructed when the measurements from multiple sensors are combined. Thus, the sensing region for a cluster of sensors can be much greater then the union of their sensing disks. Second, the disk model is inadequate for certain applications.

For example, when the objective of coverage is to localize a target within a certain error margin, ensuring that the region is covered by sensing disk can not guarantee precise localization. Even with the concept of k-coverage, where every point should be covered by at least k sensors, the network still cannot provide a high localization bound. Due to the limited capabilities of sensor nodes, providing security and privacy to a sensor network is a challenging task. The primary functionality of wireless sensor networks is to sense the environment and transport the acquired information to base stations for further processing. A number of routing protocols have been proposed for sensor networks. Sensor network routing was focused on efficiency and effectiveness of data distribution. Studies and experiences have shown that considering security in the design stage is the best way to provide security for sensor network routing.

VII. SECURITY IN GROUP COMMUNICATIONS OVER WSNS

Secure group communications provide security protection over WSNs. Zhu et al., proposed a key management protocol called a localized encryption and authentication protocol (LEAP) for large-scale distributed sensor networks, where each sensor node can establish pair-wise keys with its one-hop neighbor [13]. Multi-hop pair-wise key may be require to reach clusters heads and it can be done by each node generating a secret key and finding m intermediate nodes. The protocol is designed based on two observations: different packet types exchanged among sensor nodes require different security services, and a single key-management scheme may not be suitable for various security requirements. Four types of keys for fundamental security services can be used to secure communications [14]. These four types of keys include a pair-wise key used between a sensor node and the base station, a pair-wise key used between a pair of two sensor nodes, a shared cluster key used among all sensor nodes in the same cluster, and the group key used among all sensor nodes. Security services that can mitigate several attacks can be provided. For example, authentication of one-hop broadcast communications among nodes with one-way key chains can mitigate the impersonation attack, while a time stamp is used to expire keys to prevent node capture and sybil attacks.

VIII. SOFTWARE UPDATING IN WSNS

A critical issue in the effective deployment of these networks is the ability to update software after deployment. The WSNs related software include all application specific tasks and functions of the middleware to build up and maintain the network e.g., routing, looking for nodes, discovering services, and self localization [15]. There are a number of reasons why the software may require updating in a WSN. The Software Engineering Institute (SEI) at Carnegie-Mellon University identifies four categories of software updates for defendable systems, which help to provide an insight into these reasons: maintenance releases, minor releases, major releases (technology refresh), and technology insertion.

Embedded wireless sensor systems programmed by specialists are likely to experience higher levels of maintenance than normal. Minor release will be used to improve data collection and performance. As the needs of WSNs are likely to develop dynamically over time, major releases can be expected in response. Finally, due to the active research on WSNs and related technologies and the associated development of new algorithms and protocols technology insertation will be an important driver of software updates [16]. Wireless sensor nodes are characterized by very limited resources and by large-scale deployment [17, 18]. Accessing these nodes in the field to perform software updates can be difficult to locate or inaccessible, or the scale of the deployment can preclude individual access. Remote update poses its own problems. Three key issues are: • Avoiding interference with data collection while sharing the same communication infrastructure; • Minimizing the cost of upgrades in terms of the impact on sensor network lifetime; • Avoiding the loss of part or all of a sensor network due to an upgrade fault. WSN software update model is shown in Fig. 3. The high level data – flow diagram highlights the interactions between the three key elements of software update functionality: generation, propagation and activation.

Fig. 3. Software update model for WSNs

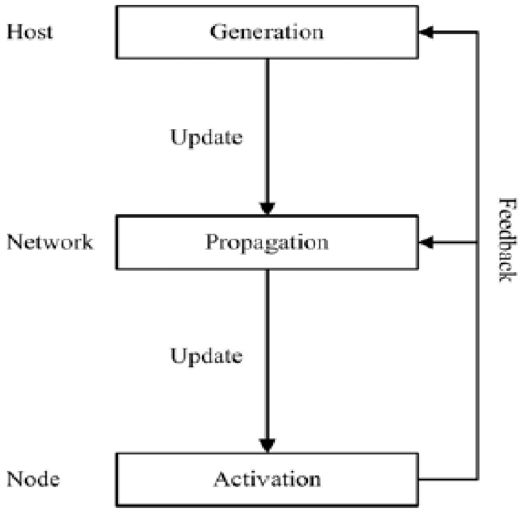


Fig. 3. Software update model for WSNs

IX. SECURE TIME SYNCHRONIZATION

Due to the collaborative nature of sensor nodes, time synchronization is important for many sensor network operations, such as coordinated sensing tasks, sensor scheduling, moving object tracking, time division multiple access (TDMA), medium access control, data aggregation, and multicast source authentication protocol. The network time protocol (NTP) is used for synchronization in the Internet [19]. A sensor network is a resource constrained distributed system and the NTP can not be directly used by sensor networks. All network time synchronization methods relay on some kind of message exchange between nodes.

No determinism in the network in the network dynamics, such as physical channel access time and operation system overhead (system calls) makes synchronization implementation challenge in sensor networks. The time synchronization schemes include reference-broadcast synchronization (RBS), timing-sync protocol for sensor networks (TPSN), etc. All of these time synchronization algorithms try to achieve either pear-wise clock synchronization or global clock synchronization. Pair-wise clock synchronization aims to obtain high precision clock synchronization between pairs of sensor neighbors, while global clock synchronization aims to provide network-wide clock synchronization in the whole network. Existing pair-wise clock synchronization protocols use receiver-to-receiver synchronization in which a reference node broadcast a reference packet to help pairs of receivers identify the clock differences,orsender-receiver synchronization, where a sender communicates with a receiver to estimate the clock difference. Most of the global clock synchronization protocols establish multi-hop paths in a sensor network.

In that way, all nodes can synchronize their clocks to a given source based on these paths and the pair-wise clock differences between adjacent nodes in these paths. Most existing time synchronization schemes are vulnerable to several attacks. Four possible attacks on sensor time synchronization are identified in [20], i.e.: • masquerade attack, • replay attack message manipulation, and • delay attack. In masquerade attack, we suppose that, for example, the node A sends out a reference beacon to its two neighbors, B and C. An attacker E, can pretend to be B and exchange wrong time information with C, distributing the time synchronization process between B and C. As for replay attack, when using the same scenario as mentioned in the first attack, the attacker E can replay B's old timing packets, misleading C to be synchronized to wrong time. In the message manipulation attack, an attacker may drop, modify or even forge the exchanged timing messages to interrupt the time synchronization process. Finally, in the case of delay attack, the attacker delays some of the time messages so as to fail the time synchronization process. This attack cannot be defined against by cryptographic techniques.

X. RECEIVER'S LOCATION PRIVACY PROTECTION WSN

Technologies promise drastic enhancement in automatic data collection capabilities through efficient deployment of small sensing devices. With the availability of cheap wireless technologies and micro sensing devices, WSNs are expected to be widely deployed in the near future. On the other hand, the open nature of wireless communications makes it easy for attackers to inject data packets in a WSN. Also, unlike other wireless networks composed of mobile nodes with human presence (PDA's, laptops), sensor networks are usually deployed in the open areas, where unattended sensor nodes lack physical protection. For the attackers, this means to encounter much fewer obstacles when attacking a sensor network [21]. As for privacy in WSNs, there exists content privacy and contextual privacy. Threats against content privacy and contextual privacy arise due to the ability of adversaries to observe and manipulate the content of packets sent over a sensor network. This type of treats is countered by encryption and authentication. Even after strong encryption and authentication are applied, wireless communication media still exposes contextual information about traffic carried in the network.

In particular, the location information about senders/receivers may be divided based on the direction of communications. In WSNs, protection of the receiver's location privacy is very important. Namely, the receiver is the most critical node of the whole network, because the duty of the receiver is to collect data from all sensors. Since all sensors send data to a single node, this creates a single point of failure in the network. In some scenarios, the receiver itself can be highly sensitive. There are several ways that an adversary can trace the location of a receiver. First, an adversary can deduce the location of the receiver by analyzing the traffic rate. This is traffic-analyzing attack. Here, the basic idea is that sensor near the receiver forward a greater volume of packets than sensor further away from the receiver.

By eavesdropping the packets transmitted as various locations in a wireless sensor network, an adversary is able to compute the traffic densities at these locations, based on which it deduces the location of the direction to the receiver. To perform the traffic-rate analysis, an adversary has to stay at each location long enough such that sufficient data can be gathered for computing the traffic rate. This process takes long time as the adversary moves from location to location. Second, an adversary can reach the receiver by following the movement of packets. In packet-tracing attack, an equipped adversary can reveal the location of the immediate transmitter of an overhead packet, and therefore he is able to perform hop-by-hop trace towards the original data source. Because the packet-tracing attack does not have to gather traffic rate information, it allows an adversary to move quickly from location to location towards the receiver. The packet tracing attack may even be able to trace a mobile receiver due to its fast response, whereas the slow response of the traffic-analysis attack makes it unsuitable for such a task. By eavesdropping the packet transmission, an adversary is able to move one hop along the shortest path towards the receiver for each packet overhead. In order to protect the receiver's location privacy, a new location-privacy routing (LPR) protocol can be used to provide path diversity. This protocol can be combined with fake packet injection to minimize the information that an adversary can deduce from the overhead packets about the direction towards the receiver [22]. As for an adversary, he can hardly distinguish between real packets and fake packets, or tell which direction is towards the receiver. Path diversity provided by LPR leads to larger routing paths, while transmitting spurious packets consumes extra energy. The stronger the protection for the receiver is required, the higher the overhead will be. If the security of the receiver is of great importance, overhead may be a price that one has to pay even in sensor networks, when there is no better alternative. Different approaches are designed to protect user's privacy in location tracking systems, which determine the positions for location-based services. Location privacy in these studies is content-oriented, where location information is collected and protected at the user's private data. A routing protocol is needed for packets to be forwarded from sources to the receiver in order to collect data from the field. Broadcast protocol can be used in which every data packet is flooded to all nodes in the network, including the receiver. Broadcast is extensively used in the route discovery phase in many routing protocols [23].

A broadcast protocol is able to achieve location privacy for the receiver because, under broadcast routing, every packet is equally forwarded to all directions and every node in the network "receives" a copy of the packet which makes it impossible for an adversary to tell which direction points to the receiver. Broadcast routing has an extremely high energy cost, which renders this approach impractical. Another security problem of broadcast routing is that it quickly exposes the locations of all sensors in the network. Some routing protocols establish a single path from each source node to the receiver. Each time the receiver moves to a new location, it broadcast a beacon packet in the network. When a node receives a beacon for the first time, it forwards the beacon to its neighbors by a local broadcast. The beacon roughly follows a shortest-path tree to all sensors, which record their parents as the next hops to the receiver.

Data packets will then follow the reverse direction of the broadcast tree towards the receiver. This procedure is similar to the interest propagation phase and the data propagation phase in the directed diffusion scheme where "gradients" from each node towards the receiver are first built before data packets can be routed. Of course, single path routing is vulnerable to the packet-tracing attack. Taking into account that single path routing is not safe for the receiver and broadcast routing is not practical, a different routing scheme is needed [24, 25]. Example 3 Let us illustrate haw an adversary traces packets in a sensor network. When a packet is transmitted as a local broadcast an adversary overhearing the transmission can only tell the location of the immediate transmitter, but not the location of the node that is receiving the packet. A behavior of the adversary is shown in Fig. 4.

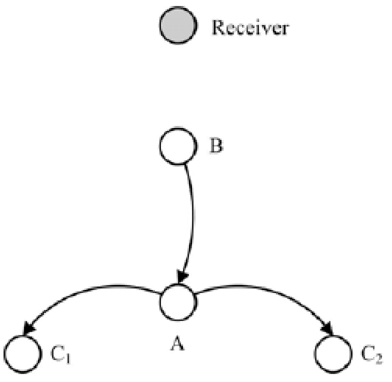


Fig. 4. Behavior of the adversary

Fig. 4. Behavior of the adversary Suppose that the adversary resides at node A. He overhears a transmission mode from node B. Shortly after, he overhears a transmission from node A. Then, he overhears a transmission from node C1. For now, we ignore the arrow from A to C2. Based on the above sequence of transmissions the adversary learns that a packet was sent from B to A and then to C1. The adversary will move to C1, hoping that he is one hop closer to the receiver. In order to camouflage the movement of the real packet, node A may send a spurious packet to C2. After overhearing two transmissions from A and then two subsequent transmissions from C1 and C2 respectively, the adversary knows that the packet from B to A has been forwarded to either C1 or C2, and he has to pick one to trace.

The adversary may guess that the packet sent to C1 is the real one and the one sent to

C2 is a fake. Namely, with respect to the forwarding line from B to A, the deviation of C2 from this line is greater than that of C1. Starting from the point that the goal of the routing protocols is to deliver a packet to its destination along a shortest path as possible, the adversary reasonable decides that C1 has a greater chance to be the real next hop to the receiver. This analysis demonstrates the ability of an adversary to infer the receiver's location through information overhead.

XI. CONCLUSION

Security in sensor networks has been an increasingly important issue for both academia and in industry individuals and groups working in this fast growing research area. In a WSN, physical security of wireless links is virtually impossible because of the broadcast nature and resource limitation on sensor nodes and uncontrolled environments where they are left unattended. Consequently, security attacks on information flow can be widespread. Key management has become a challenging issue in the design and deployment of secure WSNs.

A common assumption is most existing distributed key management schemes is that all sensor nodes have the same capability. However, connectivity and lifetime of a sensor network can be improved if some nodes are given greater power and transmission capability. Target tracking and localization are important applications in WSNs. Although the coverage problem for target detection has been intensively studied, few consider the coverage problem from the perspective of target localization. Due to their role in WSNs, localization algorithms/systems can be the target of an attack that could compromise all the functionalities of a WSN, and lead to incorrect decision making in addition to other problems that may arise. Current localization systems are vulnerable to the security attacks, while the existing techniques can be used to prevent the attacks in WSNs. Efficient software updating is in many ways one of the most challenging features to provide on a WSN. It requires reliability large amounts of data to be reliably disseminated to the nodes, sophisticated mechanisms to minimize the cost of this dissemination, and aggregated status to be returned to a host. It may also require tracking of software failures and recovering from its failures in a network-wide manner. It must provide support to handle network partitioning, node failures, software failures, data transmission failures and other intermittent and persistent faults. Presented issues are crucial for the future implementation of wireless sensor networks.

REFERENCES

[1] B. Krishnamashari, D. Estrin and S. Wicker, "Impact of Data Aggregation in Wireless Sensor Networks", Proc. 22nd International Conference Distrib. Comp. Systems, Jul. 2002, pp. 575-578.

[2] H. Luo, Y. Lin and S. K. Das, "Routing Correlated Data in Wireless Sensor Networks: A Survey", IEEE Network, vol. 21, no.6, Nov/Dec. 2007, pp. 40-47.

[3] K. Lu et al., "A Framework for a Distributed Key Management Scheme in Heterogeneous Wireless Sensor Networks", IEEE Transactions on Wireless Communications, vol. 7, no. 2, Feb. 2008, pp. 639-647.

[4] X. Du, and H-H. Chen, "Security in Wireless Sensor Networks", IEEE Wireless Communications, vol. 15, no. 4, Aug. 2008, pp.60-66.

[5] J. C. Lee, et al., "Key Management Issues in Wireless Sensor Networks: Current Proposals and Future Developments", IEEE Wireless Communications, vol. 14, no. 5, Oct. 2007, pp. 76-84.

[6] S.-P. Chan, R. Poovendran, and M.-T. Sun, "A Key Management Scheme in Distributed Sensor Networks Using Attack Probabilities", Proc. IEEE GLOBECOM, vol. 2, Nov. 2005, pp. 1007-1011.

[7] Y. W. Law et al., "A Formally Verified Decentralized Key Management for Wireless Sensor Networks", Personal Wireless Communications, ser. Lecture Note in Computer Science, Springer Berlin/Heidelberg, Sept. 2003, vol. 2775/203, pp. 27-39.

[8] R. Ramen, J. Lopez, S. Gritzalis, "Situation awareness mechanisms for wireless sensor networks ", IEEE Communication Magazine, vol. 46, no. 4, pp. 102-107, Apr. 2008.

[9] S. Rajasegarar, C. Leckie, and M. Palansiwami, "Anomaly detection in wireless sensor networks", IEEE Wireless Communications, vol. 15, no. 4, Aug. 2008, pp. 34-40.

[10] R. Khanna and H. Lin, "Control Theoretic Approach to Intrusion Detection Using Distributed Hidden Markov Model", IEEE Wireless Communications, vol. 15, no. 4, Aug. 2008, pp. 24-33.

[11] M. Cardei and J. Wu, "Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks", Computer Communications, vol. 29, no. 4, Feb. 2006, pp. 413-420.

[12] B. Wang, et al., "Information Coverage and its Applications in Sensor Networks", IEEE Communications Letters, vol. 9, no.11, Nov. 2005, pp. 967-969.

[13] S. Zhu, S. Setia and S. Jejodia, "LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks", Proc. CCS’03: 10th ACM Conference on Computer and Communications Security, New York, ACM Presss, 2003, pp. 62-72.

[14] P. Sakerindr and N. Ansari, "Security Services in Group Communications over Wireless infrastructure, Mobile Ad Hoc and Sensor Networks", IEEE Wireless Communications, vol. 14, no. 5, Oct. 2007, pp. 8-20.

[15] R. Dobrescu, et al., "Embedding Wireless Sensors in UPnP Services Networks", NAUN International Journal of Communications, vol. 1, no. 2, 2007, pp. 62-67.

[16] S. Bravn and C. J. Screenen, "A New Model for Updating Software in Wireless Sensor Networks", IEEE Network, vol. 20, no. 6, Nov/Dec. 2006, pp. 42-47.

[17] K. Xing-Hong and S. Hui-He, "Localization Assisted by the Mobile Nodes in Wireless Sensor Networks", WSEAS Transactions on Communications, vol. 6, no. 8, Aug. 2007, pp. 767-772.

[18] T-F. Shin and W-T. Chang, "Hierarchical Localization Strategy for Wireless Sensor Networks", WSEAS Transactions on Computers, vol. 7, no. 8, Aug. 2008, pp. 1260-1269.

[19] D. L. Mills, Computer Network Time Synchronization: The Network Time

Protocol, CRC Press, Taylor and Francis Group, 2006.

[20] H. Song, S. Zhu and G. Cao, "Attack Resilient Time Synchronization for Wireless Sensor Networks", Proc. 2nd IEEE Int. Conf. Mobile Ad Hoc and Sensor Systems, Washington, DC, Nov. 2005, pp. 765–772.

[21] F. Akyilidiz, et al., "Wireless Sensor Networks: A Survey", Computer Networks, vol. 38, no. 4, Mar.2002, pp. 393-422.

[22] Y. Jian, et al., "A Novel Scheme for Protecting Receiver's Location Privacy in Wireless Sensor Networks", IEEE Transactions on Wireless Communications, vol. 7, no.10, Oct. 2008, pp. 3769-3779.

[23] H. Lim and C. Kim, "Flooding in Wireless Ad Hoc Networks", Computer Communications, vol. 24, no. 3-4, Feb. 2001, pp. 353-363.

[24] H. M. Park et al., "Modified Backoff Scheme for MAC Performance Enhanced in IEEEE 802.15.4 Sensor Network", WSEAS Transactions on Communications, vol. 6, no. 3, Mar. 2007, pp. 457-463.

[25] M. S. Kakasageri, S. S. Manvi and G. D. Sorgavi, "Agent-Based Information Access in Wireless Sensor Networks", WSEAS Transactions on Communications, vol. 3, no. 7, Jul. 2006, pp. 1369-1374.