International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.1, Issue.2, pp-520-526 ISSN: 2249-6645

Security Issues in Wireless Sensor Networks

Priyanka Shrivastava
(Department of Computer Science & Engineering, Shri Ram Institute of Technology, Jabalpur
Rajiv Gandhi Technical University, Bhopal)

ABSTRACT

Wireless Sensor Networks (WSN) is a recent advanced technology of computer networks and electronics. The WSN increasingly becoming more practicable solution to many challenging applications. The sensor networks depend upon the sensed data, which may depend upon the application. One of the major applications of the sensor networks is in military. So security is the greatest concern to deploy sensor network such hostile unattended environments, monitoring real world applications. But the limitations and inherent constraints of the sensor nodes does not support the existing traditional security mechanisms in WSN. Now the present research is mainly concentrated on providing security mechanism in sensor networks. In this context, security aspects of the sensor networks like requirements, classifications, and type of attacks etc., is analyzed in this survey paper.

1. INTRODUCTION

The sensor network is a group of self- organized, low priced sensor nodes and creates network in spontaneous manner. The WSN combines sensing, computation and communication in a single small device, called Sensor Node. The sensor node mainly contains radio, battery, microcontroller and power devices. Another term of sensor node is "mote". The sensors in a node provides the facility to get the data like pressure, temperature, light, motion, sound etc and capable of doing data processing. The main goal of the applications is achieved by the cooperation of all sensor nodes in the network. There are many sensor network applications like such environmental data collection, security monitoring, medical science, military, tracking etc. when sensor networks are randomly deployed in a hostile environment, security becomes extremely important factor. Because sensed data of sensor nodes is prone to different types of malicious attacks before reaching base station. Security mechanisms are needed in communication part of the networks to provide safe data. The security is also important concern to get full advantage of in-network data processing sensor networks. Protecting such a sensed data is complicated task. Even through wireless sensor network is an advanced technology of network, it is extremely different from traditional wireless networks. This is, due to the unique characteristics of sensor nodes in WSN. So existing security mechanisms of traditional wireless networks are not directly applied in WSN. Sensor networks are closely interacting physical environment. So sensor nodes are also deployed in all areas even physical accessible attacks and broadcasting sensed data in network. So these reasons give a scope to new security mechanism rather than applying existing traditional security mechanisms in WSN.

2. SECURITY REQUIREMENTS IN WSN

The objective of confidentiality is required in sensors environment to protect information traveling among the sensor nodes of the network or between the sensors and the base station from disclosure. Authentication in sensor networks is essential for each sensor node and base station to have the ability to verify that the data received was really sent by trusted sender or not. This authentication is needed during the clustering of sensor node in WSN. We can trust the data sent by the nodes in that group after clustering. Integrity controls must be implemented to ensure that information will not be altered in any unexpected way. Many sensor applications such as pollution and healthcare monitoring rely on the integrity of the information to function with accurate outcomes. Secure management is needed at base station, clustered nodes, and protocol layer in WSN. Because security issues like key distribution to sensor nodes in order to establish encryption and routing information need secure management.

3. ATTACKS IN WSN

The basic categories of attacks against privacy in sensor networks are eavesdropping, disruption and hijacking. The eavesdropping is used to know the output of sensor networks by listing transmitted messages of sensor nodes. There are mainly two ways to know about output data by concealing from sensor nodes or sending queries to sensor nodes or root nodes or aggregation points or attacks sensor nodes. The former approach is called passive eavesdropper and later approach is called active eavesdropper. The location of eavesdropper plays major role in getting information. This attack affects the property of confidentially, authentication in WSN. So proper encryption mechanism, message authentication code are needed before broadcasting data. The disruption mainly influences output of the network. The semantic disruption injects messages, corrupts data or changes values in order to render the aggregate data corrupted, useless and incomplete. Physical disruption renders the sensor readings by directly manipulating the environment. The hijacking approach is used to take the control over sensor node in network. The hijacking mechanism gives more power to eavesdropping and disruption by hijacking main sensor nodes. Another major attack in WSN is Denial of Service attacks. Some of the denials of service attack are at routing layer, link layer and transport layer. One of the denials of service attack is jamming networks. That is simply interfaces transmission frequency of WSN. There are mainly two types in jamming. In constant jamming, no messages are able to send or receive by a node in WSN. So this is complete jamming of network. In Intermittent jamming, the nodes are exchange messages with highly risks. Another new attack in WSN is Sybil attack. This Sybil attack is defined as a "malicious

device illegitimately taking on multiple identities". This attack is affecting redundancy mechanism, routing algorithms, resource allocation procedure and data aggregation mechanism. With little effort, an adversary may capture nodes, analyze and replicate them, and surreptitiously insert these replicas at strategic locations within the network. They may allow the adversary to corrupt network data or even disconnect significant parts of the network. This attack can change entire network goal. This attack affects Integrity, confidentiality.

4. SECURITY MECHANISMS

Now days, the researchers are attracted by security concepts of wireless sensor networks. Many researchers have proposed some security mechanisms in wireless sensor networks. In this section, we are dealing briefly on several existing security mechanisms for WSN's. These are:

4.1 "SecFleck: Public key cryptography in wireless senor networks"

Approach is used to provide the message security services as confidentiality, Integrity and Authenticity in WSN at computationally fast and lower energy utilization. To design and implementation of public key system in WSN needs new version hardware and software in mote. This approach is named as secFleck. It uses trusted module platform chip at hardware level and some software primitives. This approach uses RSA algorithm to implement asymmetric public key system. This approach has taken smaller RSA exponent (65537) and key size (2048) to provide security levels. This approach uses new operating system called Flack OS (FOS). FOS is a c-based cooperative multi-threaded operating system with public key cryptography primitives like encryption, decryption, singing, signature verification etc. Even this approach works fine for message security level, the learning new OS functions is length and complicated process. It also needs new hardware to provide message security level.

4.2 "LiSP: A Lightweight security protocol for wireless sensor networks"

LISP aims to provide authentication without retransmission of keys and also provides scalability in computing. It uses symmetric key system approach. It uses temporary keys and master keys. Temporary keys (TK) are used to encrypt and decrypt data packets. The master key (MK) is used to send temporary keys to single node. After network had been deployed, this protocol automatically selects one group of cluster heads as key server. The key server is used to distribute the temporal key, authenticate new nodes and detect nodes that have been compromised. When a key server transmits a packet for the first time it contains the length of the TK buffer, the key refresh rate, and the initial TK. The need for a Message Authentication Code is eliminated because the nodes are able to implicitly authenticate the TK by checking to see if the new TK matches the sequence of the other TK's in the TK buffer. LiSP provides a great deal of protection from compromised nodes and key servers. The keying system with implicit authentication allows the sensor to quickly detect whether or not the key that was sent from the key server is authentic or not. As long as the refresh rate is not very fast the sensors will not run out of battery power at a fast rate. LiSP is very scalable because the key server does most of the calculations and the key server can change depending on whether the key server has been compromised or not. This protocol is used to reduce the retransmission of keys and provides implicit message authentication scheme to reduce the overhead. The keying mechanism depends upon application of wireless sensor networks.

4.3 TinySec: A link layer security architecture for wireless sensor networks"

TinySec is a light weight and link layer security protocol. It provides security services as message Integrity, message authentication and access control at routing level and Reply protection in Adversary. It supports two different security options. They are Authenticated Encryption and Authentication only. In the Authenticated Encryption, the payload is encrypted first and then packet is encrypted using MAC. In Authentication only, the packet is directly encrypted with MAC without encrypting payload. This approach is used Cipher Blocked Chaining to encryption. TinySec is independent of cipher, key scheme, and application. The TinySec packets are more in size then WSN packets, due to this; it needs more computing and processing power.

4.4 "SPINS: Security Protocol for Wireless Sensor Networks"

This protocol is used to provide security services as freshness, Authentication, Confidentiality and Integrity. The two-way authentication, data confidentiality, freshness and integrity are provided with the help of Secure Network Encryption Protocol (SNEP) scheme and Authentication for Broadcast messages is provided with the help of µTELSA (the "micro" version of the Timed, Efficient, Streaming, and Loss-tolerant Authentication Protocol) scheme. A block cipher RC5 algorithm was used by SNEP But it gives chances to eavesdropping to get plain and cipher text in way. Due to semantic security is low in SNEP implementation. The Localized Encryption and Authentication Protocol security mechanism provides confidentiality and authentication mechanisms in sensor networks. This mechanism uses four different keys for each sensor node and controller to maintain master keys. They are individual key, pair-wise key, cluster key and group key. The individual key is unique for each node and used to provide secure communication between node and base station. This key is pre-loaded into each sensor node before deployment. A cluster key is a shared key and is shared by all neighbor nodes in the cluster. It is mainly used for securing broadcast messages in cluster groups because in-network computation is done at the cluster heads in WSN.

The pair-wise shared key used to provide secure communication and authentication between immediate nodes or one hop nodes in WSN. This key is used before transmitting cluster key in cluster group. It is generated when the same key nodes are deployed in a single hop distance. The group is also a shared key. This key is shared by base station and set of nodes for broadcasting encrypted messages. This key used for hop-by-hop translation messages. The nodes are stationary in this approach. This approach needs more resource in-terms of computation power, memory to store keys and processing resources. But

according to sensor network characteristics, this approach is inefficient and power consumable. This approach does not give good results on security damaged sensor applications. This approach should be applied prior to deployment of sensor network application.

In Random key pre-distribution schemes, a centralized key server generates a large key pool at offline. This generation of keys is done in key distribution phase. In key discovery phase, each sensor broadcasts their key identifiers or private shared keys. Then sensor nodes get the information about neighbor and network information after processing shared keys. The communication of data has to be done by shared key authentication. Too many sensor nodes are usually deployed for any sensor applications. Assigning unique keys to sensor node is a cumbersome problem. Even thorough, this mechanism used modified schemes like Purely Random Key Pre-distribution and Structured Key Pool Random Key Pre-distribution are inefficient to assigning keys to nodes in WSN. The attackers make use of advantage of decentralized pool key generation. Public cryptography such as such as Diffie-Hellman key establishment at booting stage in base station, gives single point of failure of sensor network. So to provide efficient security mechanism, decryption should be done at cluster nodes and communicates the nodes or distributes messages in hierarchical manner. This scheme reduces number of keys in network, resource utilization and make utmost impossible to attacker to hijack.

4.5 "Fast Authenticated Key Establishment Protocols for Self-Organizing wireless Sensor Networks"

This protocol has a goal to provide efficient authenticated key transferring mechanism. It uses elliptic Curve Cryptography (ECC) to provide encryption for sensor nodes. Cracking the private key is very difficult even the size of ECC keys length is less. Public keys are used to authenticate keys certificates. So during the process of authenticate keys certificates, this approach is usually finds public keys. These certificates are generated by sensor node and security manager. This work is accomplished by computation server if needed. The main drawback of using this key establishment protocol is that sometimes a computation server may be needed for some of the computations. The amount of packets that are exchanged to authenticate a key seems like lengthy process to authenticate a key. It is difficult to figure out the strength of this protocol. Because this depends upon the keys and they contains random values.

The adversary attack leads to node replication attack with little effort. One approach to detect the replication node in wireless sensor networks is centralized scheme. In the Centralized scheme, all nodes in the network transfers a list of their neighbor's claimed locations to a central base station. Then base station can examine the lists for conflicting location claims. Even though this approach is efficient, the nodes closest to the base station will receive the brunt of the routing load and will become attractive targets for the adversary. This protocol is also delays revocation, since the base station must wait for all of the reports to come in, analyze them for conflicts and then flood revocations throughout the network. Suppose adversary attacks at base station then centralized approach is inefficient and does not do well. At this case, this protocol gives single point of failure. The network lifetime is also decreases due to high traffic at base station surroundings. Even though this approach detects all replicated node in easy way, it requires more storage area in each node and also requires communication messages. Another scheme to overcome the difficulties in centralized scheme is Location Detection scheme. In this scheme, instead of implementing node replication detection scheme at base station, it process at node's neighbor. It uses a voting mechanism; it collects neighbor's opinions on the legitimacy of the node. This approach is unable to detect the clones (i.e. nodes giving support to adversary) in disjoint neighborhood in network. It fails to detect subvert and clone if they are more than two hops away. Due to these drawbacks, this protocol became inefficient to find replication nodes in WSN. One simple approach to detect the distributed replication nodes is Simple Broadcast protocol. In this approach, each node broadcast authenticated messages about their location and also stores the information about neighbor nodes. Even though this approach gives 100% results, it may not works if adversary attacks at key areas or communication paths. This approach costs more in form of communication for large networks. One of the improvements of Simple Broadcast Scheme is Deterministic Multicast Protocol. The main of this approach is to reduce the communication of simple broadcast scheme by sharing the node's location to a subset of deterministically chosen node, called witness node. This subset may be fixed for a particular node. The witness nodes are selected based on function of node ID's and probability. So it uses multicast approach to give judgment over nodes location claim. Due to this, the number of message transfers in the network is decreased. This is also fails if adversary attacks or jams the messages in the network. Because it shares the node's location to a limited subset of deterministically chosen nodes only. This approach is not doing well, if any one of the witness node is caught by adversary.

4.6 "Distributed Detection of node replication attacks in wireless sensor networks"

This protocol deals with detection of node replication attacks due to adversary at protocol level (routing layer). It uses two routing algorithms Randomized Multicast and Line selected Multicast. The adversaries have to be detected as soon as it occurs otherwise replicated nodes are increases in next data gathering cycle. Assume that the adversary cannot readily create new IDs for nodes. In the cloned formation, this assumed to be at least one node as legitimate neighbor to clone. It also assumes the adversary in stealthy manner. Due to this, the detection of adversary is complex. So it uses one protocol that sweeps the network, using SWATT technique to remove compromised node and human interactions. Here it assumes that the adversary can read and write the messages using only nodes under adversary control. [i.e. read and writing messages should do in adversary control parts by adversary.]

This also works in a situation that, the adversary can change the topology of the network by adding replicas.

4.7 Commutative Cipher based En-route Filtering (CCEF)

CCEF exploits a bootstrapping phase to establish trust between individual sensor nodes and the base station. In the operational phase, the base station can initiate a query-response session and install per-session security states in the sensor nodes at any time. The tasked sensor nodes response by generating and endorsing data reports on their sensing results. When the reports are forwarded to the base station, each intermediate node verifies the authenticity of the reports, and filters the fabricated ones. The base station further verifies the reports that it receives, and reacts to the compromised nodes by refreshing the session state.

Commutative Cipher based En-route Filtering scheme (CCEF) defends against event fabrication attacks without symmetric key sharing among sensor nodes. CCEF exploits the typical operational mode of query-response in sensor networks, and installs security states in the nodes in an on-demand manner. Specifically, in CCEF, each node has a unique ID and is preloaded with a unique node key. The base station initiates a query-response session by sending out a query to task specific sensor nodes to report their sensing results. The base station prepares two keys for each session: one session key and one witness key. The session key is securely sent to source node, i.e., the node tasked to generate reports, while the witness key is in plaintext and recorded by all intermediate nodes. A legitimate report is endorsed by a node MAC jointly generated by the detecting nodes using their node keys, and a session MAC generated by the source node using the session key. Through the usage of a commutative cipher, a forwarding node can use the witness key to verify the session MAC, without knowing the session key, and drop the fabricated reports. The base station further verifies the node MAC in the report that it receives, and refreshes the session key upon detection of compromised nodes.

4.8 Interleaved hop-by-hop authentication (IHA)

This technique deals with false data injection attack by enabling the base station to verify the authenticity of a report that it has received as long as the number of compromised sensor nodes does not exceed a certain threshold. Further, it attempts to filter out false data packets injected into the network by compromised nodes before they reach the base station, thus saving the energy for relaying them.

This scheme is particularly useful for large-scale sensor networks where a sensor report needs to be relayed over several hops before it reaches the base station and for applications where the information contained in the sensor reports is not amendable to the statistical techniques used by SIA (e.g., non-numeric data). In this scheme at least t + 1 sensor nodes have to agree upon a report before it is sent to the base station. Further, all the nodes that are involved in relaying the report to the base station authenticate the report in an interleaved, hop-by-hop fashion. Here t is a security threshold based on the security requirements of the application under consideration and the network node density. This scheme guarantees that if no more than t nodes are compromised, the base station will detect any false data packets injected by the compromised sensors. In addition, for a given t, this scheme provides an upper bound B for the number of hops that a false data packet can be forwarded before it is detected and dropped. If every non compromised node on the path between a cluster head and the base station knows the ids of the nodes that are t +1 hops away from it on the path, then B = t; otherwise, without this knowledge, B = (t - 1)(t - 2).

4.9 Localized Encryption and Authentication Protocol (LEAP)

Localized Encryption and Authentication Protocol, a key management protocol for sensor networks is designed to support in-network processing, while at the same time providing security properties similar to those provided by pair wise key sharing schemes. In other words, the keying mechanisms provided by LEAP enable in-network processing, while restricting the security impact of a node compromise to the immediate network neighborhood of the compromised node. LEAP includes support for multiple keying mechanisms. The design of these mechanisms is motivated by the observation that different types of messages exchanged between sensor nodes have different security requirements, and that a single keying mechanism is not suitable for meeting these different security requirements. Specifically, this protocol supports the establishment of four types of keys for each sensor node– an individual key shared with the base station, a pair wise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key shared by all the nodes in the network. Moreover, the protocol used for establishing these keys for each node is communication and energy-efficient, and minimizes the involvement of the base station.

LEAP also includes an efficient protocol for inter-node traffic authentication based on the use of one-way key chains. A salient feature of the authentication protocol is that it supports source authentication (unlike a protocol where a globally shared key is used for authentication) without preventing passive participation (unlike a protocol where a pair wise shared key is used for authentication).

The packets exchanged by nodes in a sensor network can be classified into several categories based on different criteria, e.g. control packets vs data packets, broadcast packets vs unicast packets, queries or commands vs sensor readings, etc. The security requirements for a packet will typically depend on the category it falls in. Authentication is required for all type of packets, whereas confidentiality may only be required for some types of packets. For example, routing control information usually does not require confidentiality, whereas (aggregated) readings transmitted by a sensor node and the queries sent by the base station may need confidentiality. No single keying mechanism is appropriate for all the secure communication that is needed in sensor networks. As such, LEAP supports the establishment of four types of keys for each sensor node –an individual key shared with the base station, a pair wise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network.

Individual Key Every node has a unique key that it shares pair wise with the base station. This key is used for secure communication between a node and the base station. For example, a node can use this key to compute message authentication codes (MACs) for its sensed readings if the readings are to be verified by the base station. A node may also send an alert to the base station if it observes any abnormal or unexpected behavior of a neighboring node. Similarly, the base station can use this key to encrypt any sensitive information, e.g. keying material or special instruction that it sends to an individual node. **Group Key** is a globally shared key that is used by the base station for encrypting messages that are broadcast to the whole group. For example, the base station issues missions, sends queries and interests. Note that from the confidentiality point of view there is no advantage to separately encrypting a broadcast message using the individual key of each node. However, since the group key is shared among all the nodes in the network, an efficient rekeying mechanism is necessary for updating this key after a compromised node is revoked.

Cluster Key is a key shared by a node and all its neighbors, and it is mainly used for securing locally broadcast messages, e.g., routing control information, or securing sensor messages which can benefit from passive participation. Researchers have shown that in-network processing techniques, including data aggregation and passive participation are very important for saving energy consumption in sensor networks. For example, a node that overhears a neighboring sensor node transmitting the same reading as its own current reading can elect to not transmit the same. In responding to aggregation operations such as MAX, a node can also suppress its own reading if its reading is not larger than an overheard one. For passive participation to be feasible, neighboring nodes should be able to decrypt and authenticate some classes of messages, e.g., sensor readings, transmitted by their neighbors. This means that such messages should be encrypted or authenticated by a locally shared key. Therefore, in LEAP each node possesses a unique cluster key that it uses for securing its messages, while its immediate neighbors use the same key for decryption or authentication of its messages.

Pair wise Shared Key Every node shares a pair wise key with each of its immediate neighbors. In LEAP, pair wise keys are used for securing communications that require privacy or source authentication. For example, a node can use its pair wise keys to secure the distribution of its cluster key to its neighbors, or to secure the transmissions of its sensor readings to an aggregation node. Note that the use of pair wise keys precludes passive participation.

4.10 Location-aware end-to-end data security (LED)

LED is an integrated security design providing comprehensive protection over data confidentiality, authenticity, and availability. It overcomes the limitations of the existing hop- by-hop security paradigm and achieves an efficient and effective end-to-end security paradigm in WSNs. It exploits the static and location-aware nature of WSNs, and proposes a novel location-aware security approach through two seamlessly integrated building blocks: a location-aware key management framework and an end-to-end data security mechanism. In this approach, each sensor node is equipped with several types of symmetric secret keys, some of which aim to provide end-to-end data confidentiality, and others aim to provide both end-to-end data authenticity and hop-by-hop authentication. All the keys are computed at each sensor node independently from keying materials preloaded before network deployment and the location information obtained after network deployment, without inducing extra communication overhead for shared key establishment. Location-aware end-to-end data security design (LEDS) then provides a secure and reliable data delivery mechanism, which is highly resilient to even a large number of compromised nodes.

The features of LEDS and the contributions are outlined as follows:

In LEDS, the targeted terrain is virtually divided into multiple cells using a concept called virtual geographic grid. LEDS then efficiently binds the location (cell) information of each sensor into all types of symmetric secret keys owned by that node. By this means, the impact of compromised nodes can be effectively confined to their vicinity, which is a nice property absent in most existing security designs. What the attacker can do is to misbehave only at the locations of compromised nodes, by which they will run a high risk of being detected by legitimate nodes if effective misbehavior detection mechanisms are implemented. Second, LEDS provides end-to-end security guarantee. Every legitimate event report in LEDS is endorsed by multiple sensing nodes and is encrypted with a unique secret key shared between the event sensing nodes and the sink. Furthermore, the authenticity of the corresponding event sensing nodes can be individually verified by the sink. This novel setting successfully eliminates the possibility that the compromise of nodes other than the sensing nodes of an event report may result in security compromise of that event report, which is usually the case in existing security designs.

Third, LEDS possesses efficient en-route false data filtering capability to deal with the infamous bogus data injection attack. As long as there are no more than t compromised nodes in each single area of interest, LEDS guarantees that a bogus data report from that cell can be filtered by legitimate intermediate nodes or the sink deterministically. Last, LEDS provides high level assurance on data availability by counteracting both report disruption and selective forwarding attacks, simultaneously. By taking advantage of the broadcast nature of wireless links, LEDS adopts a one-to-many data forwarding approach, which is fully compatible with the proposed security framework. That is, all reports in LEDS can be authenticated by multiple next-hop nodes independently so that no reports could be dropped by a single node(s). Thus, LEDS is highly robust against selective forwarding attacks as compared to the traditional one-to-one forwarding approach used by existing security designs. In addition, LEDS adopts a (t; T) threshold linear secret sharing scheme (LSSS) so that the sink can recover the original report from any t out of T

legitimate report shares. Not only this approach enhances the event report authenticity by requiring T sensing nodes to collaboratively endorsement the report, but also makes LEDS resilient to the interference from up to T;t compromised nodes in the event area. LEDS is highly resilient to both types of attacks.

4.11 Location-based resilient security (LBRS)

This technique overcome the threshold limitation and achieves graceful performance degradation to an increasing number of compromised nodes. Location-based security approach based on two techniques: location-binding keys and location-based key assignment. In this approach, symmetric secret keys bind to geographic locations, as opposed to sensor nodes, and assign such location-binding keys to sensor nodes based on their deployed locations. a Location-Based Resilient Security (LBRS) solution, demonstrates that such a location-based approach can effectively limit the damage caused by even a large collection of compromised nodes. In LBRS, the terrain is divided into a regular geographic grid, and each cell on the grid is associated with multiple keys. Based on its location, a node stores one key for each of its local neighboring cells and a few randomly chosen remote cells. To detect fabricated reports, it is required that a real event be endorsed through multiple keys bound to the specific location of the event. An attacker that has compromised multiple nodes may obtain keys bound to different cells, but he cannot combine such keys to fabricate any event without being detected. To limit the damage of network re source waste, each node uses its keys of remote cells to verify and drop forged reports passing through it.

Location-based security design is highly resilient to compromised nodes for three reasons. First, it prevents the attacker from arbitrarily abusing a compromised key, be cause a key bound to a geographic location can only be used for purposes related to that particular location (e.g., to endorse events detected there). Second, it constrains the damage when the attacker compromises multiple nodes and accumulates their keys, because a collection of keys bound to different locations cannot be used together for any meaningful purpose. Finally, it limits the keys stored by individual nodes, because each node is assigned only a few keys based on its location. As a result, the security protection offered by our design degrades gracefully, without any threshold break-down, when more and more nodes are compromised.

4.12 Statistical En-route Filtering (SEF)

SEF exploits the sheer scale and dense deployment of large sensor networks. To prevent any single compromised node from breaking down the entire system, SEF carefully limits the amount of security information assigned to any single node, and relies on the collective decisions of multiple sensors for false report detection. When a sensing target (henceforth called “stimulus” or “event”) occurs in the field, multiple surrounding sensors collectively generate a legitimate report that carries multiple message authentication codes (MACs). A report with an inadequate number of MACs will not be delivered. As a sensing report is forwarded towards the sink over multiple hops, each forwarding node verifies the correctness of the MACs carried in the report with certain probability. Once an incorrect MAC is detected, the report is dropped. The probability of detecting incorrect MACs increases with the number of hops the report travels. Depending on the path length, there is a non-zero probability that some reports with incorrect MACs may escape enroute filtering and be delivered to the sink. In any case the sink will further verify the correctness of each MAC carried in each report and reject false ones. This is the first effort that addresses false sensing report detection problems in the presence of compromised sensors. SEF is able to detect and drop 80 to 90% injected reports by a compromised node within 10 forwarding hops, thus reducing energy consumption by 50% or more in many cases. The SEF design seeks to achieve the following goals:

4.12.1 Early detecting and dropping of false data reports

Identifying false reports allows the user to avoid taking responses to fabricated events. Although this can be done either during the data delivery process or at the sink after the data is delivered, early en-route detection of such reports can prevent them from reaching the sink, thus saving energy and bandwidth resources of nodes on data forwarding paths.

4.12.2 Low computation and communication overhead

Given the resource constraints of low-end sensor nodes, SEF strives to scale to large sensor networks and be resilient against node failures. We will show that by using only hash computations which are efficient even on low-end sensor hardware, SEF can detect and en-route drop false reports injected by an attacker who captures up to a threshold number of nodes. SEF consists of three components which work in concert to detect and filter out forged messages: (1) each legitimate report carries multiple MACs (in the form of a Bloom filter) generated by different nodes that detect the same stimulus, (2) intermediate forwarding nodes detect incorrect MACs and filter out false reports en-route, and (3) the sink verifies the correctness of each MAC and eliminates remaining false reports that elude en-route filtering.

In SEF there is a global key pool. However only the sink has the knowledge of the entire pool. Each sensor stores a small number of keys that are drawn in a randomized fashion from the global key pool before deployment. Once a stimulus appears in the field, multiple detecting nodes elect a Center-of-Stimulus (CoS) node that generates the report. Each detecting node produces a keyed MAC for the report using one of its stored keys. The CoS node collects the MACs and attaches them to the report in the form of a Bloom filter. These multiple MACs collectively act as the proof that a report is legitimate. A report with an insufficient number of MACs will not be forwarded. The key assignment procedure should ensure that each node can only generate part of the proof for a legitimate report. Only by the joint efforts of multiple detecting nodes can the complete proof be produced. Therefore to get a forged data report forwarded a compromised node has to forge MACs to assemble a seemingly complete proof. At the same time, the key assignment procedure should also ensure that any two nodes share common keys with a certain probability. When the report with forged MACs is forwarded by

intermediate nodes, probabilistic key sharing allows them to examine the correctness of the MACs probabilistically, thus detecting and dropping false reports en-route. The sink serves as the final goal-keeper for the system. When it receives reports about an event, the sink verifies every MAC carried in the report because it has complete knowledge of the global key pool. False reports with incorrect MACs that sneak through en-route filtering will then be detected.

Currently SEF also does not address the issues of how to identify compromised nodes or revoke compromised keys. For identification, neighbor nodes may overhear the channel to detect unusual activities of compromised nodes such as high traffic volume and notify the sink. After the nodes are identified, the user may deploy new nodes and the sink could flood instructions to revoke compromised keys and propagate new ones.

In summary, SEF is not designed to address all the attacks that a compromised node may launch, such as dropping legitimate reports passing through it, recording and replaying legitimate reports, or injecting false control packets to disrupt other protocols. Existing techniques can be used to address some of these issues points out that one can use multipath forwarding to effectively alleviate dropping of legitimate reports demonstrate that sensors can use a cache to store the signatures of recently forwarded reports, thus preventing identical packets from being forwarded again.

5. Conclusion

Sensor networks serving mission-critical applications are potential targets for malicious attacks. Although a number of recent research efforts have addressed security issues such as node authentication, data secrecy and integrity, they provide no protection against injected false sensing reports once any single node is compromised. These techniques aim at detecting and dropping such false reports injected by compromised nodes. Takes advantage of the large scale and dense deployment of sensor networks. Collaborative filtering of false reports requires that nodes share certain amount of security information. The more security information each forwarding node possesses, the more effective the en-route filtering can be, but also the more secret the attacker can obtain from a compromised node. Further step includes evaluation of the tradeoffs between these two conflict goals, and gaining further insight on how to build a sensor network that can be at once resilient against many compromised nodes as well as effective in detecting false data reports through collaborative filtering.

REFERENCES

[1] M. Anand, Z. Ives, and I. Lee. “Quantifying Eavesdropping Vulnerability in Sensor Networks”, In Proc. of the 2nd International VLDB Workshop on Data Mgnt. for Sensor Networks (DMSN), 2005.

[2] G. Gaubatz, J.-P. Kaps, and B. Sunar. “Public key cryptography in sensor networks”, –Revisited. In Proc. of the 1st ESAS, 2004.

[3] A. Perrig, J. Stankovic, and D. Wagner. “Security in Wireless Sensor Networks”, Communications, ACM, 47(6):53–57, 2004.

[4] David W. Carman, Peter S. Kruus, and Brian J. Matt. Constraints and approaches for distributed sensor network security. NAI Labs Technical Report #00-010, September 2000.

[5] L. Zhou and Z.J. Hass. Securing ad hoc networks. 13(6), November/December 1999.

[6] A. Wood and J. Stankovic, .Denial of Service in Sensor Networks, IEEE Computer, Oct. 2002.

[7] Elaine Shi and A. perrig, .Designing Secure Sensor Networks,. Wireless Communication Magazine, 11(6), December 2004.

[8] J. Jung, T. Park and C. Kim, .A Forwarding Scheme for Reliable and Energy-efficient Data Delivery in Cluster-based Sensor Networks,. IEEE Communication Letters, Vol.9, No.2: 112-114, Feb. 2005.

[9] W. Du, J. Deng, Y. Han, and P. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In Proc. of 10th ACM Conference on Computer and Communications Security (CCS), Washington DC, October 27-31, 2003.

[10] H. Vogt. Integrity Preservation for Communication in Sensor Networks. Technical Report No. 434, ETH Zrich, Institute for Pervasive Computing, February 2004.

[11] D. Malan. Crypto for tiny objects. Technical Report TR-04-04, Harvard University, 2004.

[12] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route filtering of injected false data in sensor networks. In INFOCOM, 2004.

[13] F. Ye, G. Zhong, S. Lu, and L. Zhang, “GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks,” ACM Wireless Networks (WINET), vol. 11, no. 2, March 2005.

[14] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In Proc. RANDOM 2002, pages 1–10, 2002.

[15] C. Karlof and D. Wagner. Secure Routing in Sensor Networks: Attacks and Countermeasures. In Proc. of First IEEE Workshop on Sensor Network Protocols and Applications, May 2003.