# Security and Privacy in Wireless Sensor Networks

Arijit Ukil
*Innovation Labs, Tata Consultancy Services*
*Kolkata, India*

## 1. Introduction

Wireless Sensor Network (WSN) consists of mostly tiny, resource-constraint, simple sensor nodes, which communicate wirelessly and form ad hoc networks in order to perform some specific operation. Due to distributed nature of these networks and their deployment in remote areas, these networks are vulnerable to numerous security threats that can adversely affect their proper functioning. Simplicity in WSN with resource constrained nodes makes them very much vulnerable to variety of attacks. The attackers can eavesdrop on its communication channel, inject bits in the channel, replay previously stored packets and much more. An adversary can easily retrieve valuable data from the transmitted packets that are sent (Eavesdropping). That adversary can also simply intercept and modify the packets' content meant for the base station or intermediate nodes (Message Modification), or retransmit the contents of those packets at a later time (Message Replay). Finally, the attacker can send out false data into the network, maybe masquerading as one of the sensors, with the objectives of corrupting the collected sensors' reading or disrupting the internal control data (Message Injection). Securing the WSN needs to make the network support all security properties: confidentiality, integrity, authenticity and availability. Attackers may deploy a few malicious nodes with similar or more hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack. The sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node. As a result, WSN has to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services. These threats can be categorized as follows:
- Common attacks
- Denial of service attack
- Node compromise
- Impersonation attack
- Protocol-specific attacks

The ad hoc or infrastructure less feature brings a great challenge to WSN security as well. For example, the dynamics of the whole network inhibits the idea of pre-distribution of a shared key between the base station and all sensors. Several random key pre-distribution schemes have been proposed in the context of symmetric encryption techniques (Chan, et al. (2003), Liu, et al. (2005)). In the context of applying public-key cryptography techniques in sensor networks, an efficient mechanism for public-key distribution is necessary as well. In the same way that distributed sensor networks must self-organize to support multi-hop routing, they must also self-organize to conduct key management and building trust relation among sensors. If self-organization is lacking in a sensor network, the damage resulting from an attack or even the hazardous environment may be devastating. Since WSN is a wireless service-oriented infrastructure, one of the most problematic attacks that it may face is the Denial of Service (DoS) attack. A DoS attack on WSN may take several forms: node collaboration, in which a set of nodes act maliciously and prevent broadcast messages from reaching certain section(s) of the sensor network; jamming attack, in which an attacker jams the communication channel and avoids any member of the network in the affected area to send or receive any packet; and exhaustion of power, in which an attacker repeatedly requests packets from sensors to deplete their battery life. Newsome et al. describe the Sybil attack as it relates to wireless sensor networks (Newson, et al. 2004). Simply put, the Sybil attack is defined as a "malicious device illegitimately taking on multiple identities" (Newson, et al. (2004)). It was originally described as an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks. In a nutshell, the security vulnerability of a WSN can be listed as:

- Denial of Service (DoS) attacks
- Link layer attacks
- Network layer attacks
- Transport layer attacks
- Link and physical layer attacks

Apart from security concern, privacy preservation in WSN is a big challenge. The explosive growth and advancement of the information age, data collection and data analysis have exploded both in size and complexity. This in turn has impacted on the privacy preservation of the data of individual users or the network itself. Privacy in our context can be defined as the control over access to information about oneself. Privacy is also the limited access to a person or a process and to all the features related to the person. Privacy preservation is important from both individual as well as organizational perspectives. There are three types of privacy threats. If an adversary can determine the meaning of a communication exchange because of the existence of a message and the context of the situation, there is a content privacy threat. If an adversary is able to deduce the identities of the nodes involved in a communication, there is an identity privacy threat. And if the adversary is able to infer the physical location of a communication entity or to approximate the relative distance to that entity, there is a location privacy threat.

In this book chapter, more emphasis will be given to privacy issues. It is understood that good amount of research works are directed (Karlof, et al. (2003), Law, et al. (2006), Gaubatz, et al. (2005) towards solving the problems of WSN security, whereas lesser effort have been put towards mitigating the problems related to WSN privacy. In fact, with the advent of the concept ubiquitous computing (Weiser, et al. (1991)), privacy becomes as important as

security. So, we mainly focus on WSN privacy issues and highlight the WSN security in brief considering the large volume of work has been already done.

## 2. WSN Security

WSNs provide unique opportunities of interaction between computing devices and their environment. The adhoc nature and wireless vulnerability make WSN a soft target for security attacks. In order to understand the security aspects of WSN, we provide a brief description of the different attacks and then present the possible solutions. First, we find out the requirements of WSN security. Then we present some of the typical attacks on WSN security and lastly we describe some well-known mechanisms for preventing some the attacks.

### 2.1 WSN requirements

WSN can be considered as a highly distributed database with wireless links. Security goals for distributed databases are very well studied. The data should be accessible only to authorized users (confidentiality), the data should be genuine (integrity), and the data should be always available on the request of an authorized user (availability). All these requirements also apply to WSNs and their users. Data confidentiality is the most important issue in network security. The objective of confidentiality is required in sensors environment to protect information travelling among the sensor nodes of the network or between the sensors and the base station from disclosure. With the implementation of confidentiality, an adversary may be unable to steal information. However, this doesn't mean the data is safe. The adversary can change the data, so as to send the sensor network into disarray. For example, a malicious node may add some fragments or manipulate the data within a packet. This new packet can then be sent to the original receiver. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit. 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 a 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. Even if confidentiality and data integrity are assured, we also need to ensure the freshness of each message. Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. This requirement is especially important when there are shared-key strategies employed in the design. Typically shared keys need to be changed over time. Another important issue is the availability factor of the nodes or the transmission media. The network should remain operational all the time. It must have some redundancy to counter link failures and have the capability to survive against different attacks.

It also needs to be understood that these requirements are to be satisfied under some kinds of limitations. Among them, limitation of device resources (limited energy, memory and computation power), unreliable communication (packet drop, latency, transmission conflicts) and unattended operation (no centralized control) need to be taken care of.

## 2.2 WSN attacks

WSNs are vulnerable to various types of attacks. These attacks can be broadly categorized as passive and active. Passive attacks do not disrupt the operation of the network. In this case the attacker snoops the data exchanged inside the network without modifying it. Detection of passive attacks is very difficult since the operation does not get affected. Where as in active attacks, data is altered and thus disturbing the normal network activities. In this chapter, we mostly focus on active attacks. It can be noted that attacks on WSNs are not limited to simply denial of service attacks, but rather encompass a variety of techniques including node takeovers, attacks on the routing protocols, and attacks on a node's physical security. We present the typical attacks from the perspective of protocol layers from where they are initiated.

### 2.2.1 Physical layer attack

Physical layer is responsible transmission of raw data bits. This is mostly involved in modulation, coding, signal detection and data encryption. Broadly two types of attacks are possible. Jamming attack is responsible for disturbing and disrupting the transmission between sender and receiver (Shi, et al. (2004)). In device tempering attack, the sensor device is physically tempered by the attacker to extract or alter the cryptographic keys and other important information (Wang, et al. (2005), Wang, et al. (2004)).

### 2.2.2 Link layer attack

In link layer, artificial collision creation, resource exhaustion, unfair and unbalanced resource allocation kind of attacks take place (Akyildiz, et al. (2002). In fact, unfairness is a kind of weak DoS attack (Wood (2002). In this scenario, the attacker attempts to degrade the time-critical applications of other nodes by disrupting their frame transmission. Another link-layer threat to WSNs is the denial-of-sleep attack. This attach prevents the node from going into sleep mode (Raymond (2006)).

### 2.2.3 Network layer attack

Network layer of WSN is vulnerable to various attacks. In wormhole attack, the attacker receives packets at one location in the network and tunnels them to another location inside the network, where the packet is resent into the network (Hu, et al. (2003)). The tunnel between the colluding attacker nodes is referred as wormhole. A particularly harmful attack against sensor networks is known as the Sybil attack, where a node illegitimately claims multiple identities. Newsome et al. describe the Sybil attack as it relates to WSNs. Sybil attack is defined as a "malicious device illegitimately taking on multiple identities" (Douceur, et al. (2002)). It was originally described as an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks. Another well-known attack which produces great amount of harm is traffic-analysis attack.

For example, a rate monitoring attack simply makes use of the idea that nodes closest to the base station tend to forward more packets than those farther away from the base station. An attacker need only monitor which nodes are sending packets and follow those nodes that are sending the most packets. In a time correlation attack, an adversary simply generates events and monitors to whom a node sends its packets (Deng, et al. (2004)). Attacks where adversaries have full control of a number of authenticated devices and behave arbitrarily to disrupt the network are referred to as Byzantine attacks. The goal of a Byzantine node is to disrupt the communication of other nodes in the network, without regard to its own resource consumption (Awerbuch, et al. (2004)). So, it is very hard to detect. In fact, a basic Byzantine attack is a black hole attack where the adversary stops forwarding data packets, but still participates in the routing protocol correctly. Routing attack is launched at disrupting the data transmission of the network. In routing attacks, routing table overflow, routing table poisoning, packet replication, rushing attacks (Hu, et al. (2003)) are reported. The most general attacks to WSN routing are spoofing, replaying, or altering routing-control information. In these attacks the adversary injects bogus routing information into the network. This leads to routing inconsistencies, and, as a consequence increases end-to-end delays and packet loss in the network. Fortunately, these types of attacks can be effectively prevented using link-layer authentication and anti-replay techniques. In a sinkhole attack, an attacker makes a compromised node look more attractive to its neighbors by forging the routing information.

### 2.2.4 Transport layer attack

At the Transport Layer attacks target the protocols that provide transfer of data between end systems. When explicit connections between identifiable nodes are used, either end of the connection maintains some form of connection control block. An attacker can issue a large number of connection setup requests that result in the exhaustion of memory at the end nodes. This is called a TCP SYN flood attack. Flooding and de-synchronization attacks are specific to transport layer. Flooding can be as simple as sending many connection requests to a susceptible node. In this case, resources must be allocated to handle the connection request. Eventually a node's resources will be exhausted, thus rendering the node useless. Another vulnerability is by session hijacking attack, where the adversary takes control over a session between two nodes. The adversary node masquerades as one of the end nodes of the session and hijacks the session. Another kind of Transport Layer attack is the desynchronization attack. This attack targets the transport protocols that rely on sequence numbers. An attacker issues forged packets with wrong sequence numbers and, as a result, causes retransmissions, which waste both energy and bandwidth.

### 2.2.5 Multilayer attack

Multilayer attacks are those that could occur in any layer of the network protocol stack. In a denial-of-service (DoS) attack, an attacker attempts to prevent legitimate users from accessing information or services (Wood, et al. (2002)). DoS attacks are commonly launched from one or more points on the Internet that are external to the victim's own system or network. In many cases, the launch point consists of one or more systems that have been subverted by an intruder via a security-related compromise rather than from the intruder's own system or systems. DoS attacks on the Internet may be launched by botnets and carried

out by compromised machines running zombie processes in the background unbeknownst to the owner of the machine, thus the risk for physical identification and apprehension of the attacker is reduced.

## 2.3 WSN security mechanisms

In this section, we briefly describe the different important security mechanisms to prevent some of the above mentioned attacks. Good amount of research efforts are engaged in finding solutions to nullify the adversary's intention. WSN security mechanisms mainly consist of robust cryptographic techniques, efficient key management, certification and other advanced methods. It is indispensable to provide basic security primitives to the sensor nodes in order to give a minimal protection to the information flow and a foundation to create secure protocols. Those security primitives are Symmetric Key Cryptography (SKC), hash primitives, and Public Key Cryptography (PKC). Since sensor nodes are highly constrained in terms of resources, implementing the security primitives in an efficient way (using less energy, computational time and memory space) without sacrificing the strength of their security properties is one of the major challenges in this area, a challenge that most of the state-of-the-art have managed to achieve. SKC primitives use the same secret key for both encryption and decryption. Instances of these primitives are able to provide confidentiality to a certain information flow, given that the origin and the destination of the data share the same secret key. They can also provide integrity and authentication if a certain mode of operation is used. These algorithms are usually not very complex, and they can be implemented easily in resource-constrained devices. Symmetric cryptography is therefore the typical choice for applications that cannot afford the computational complexity of asymmetric cryptography. Symmetric schemes utilize a single shared key known only between the two communicating hosts. This shared key is used for both encrypting and decrypting data. The traditional example of symmetric cryptography is DES (Data Encryption Standard). The use of DES, however, is quite limited due to the fact that it can be broken relatively easily. In light of the shortcomings of DES, other symmetric cryptography systems have been proposed including 3DES (Triple DES), RC5, AES, and so on (Schneier, (1996)). It can be noted that PKC is better solution where key management is an issue. In the case, where the sensor nodes can manage some amount of computational resources to perform PKC, it is always advisable to apply PKC. SKC suffers from key management problem. PKC, also known as asymmetric cryptography, is a form of cryptography that uses two keys: a key called private key, which has to be kept private, and another key named public key, which is publicly known. Any operation done with the private key can only be reversed with the public key, and vice versa. This nice property makes all PKC-based algorithms useful for authentication purposes. Still, the computational cost of calculating their underlying operations had hindered its application in highly-constrained devices, such as sensor nodes. One of the most promising PKC primitives in the field of WSN security is Elliptic Curve Cryptography (ECC), due to the small size of the keys, the memory and energy savings, and the simplicity of its underlying operation, the scalar point multiplication (Kobiltz. (1987), Liu, et al. (2005)). In order to securely distribute the cryptographic keys among the sensor nodes, efficient key management scheme needs to be deployed. Broadly, WSN key management has two categories: deterministic and probabilistic. In functional terms, three keying models are used tto cater for WSN Security and operational requirements: Network Keying, Pairwise Keying, and Group Keying.

Network keying has the advantage of being simple, flexible, and scalable. It allows data aggregation and fusion and it is able to self-organize (a key requirement in WSN). But it lacks robustness. Pairwise keying provides authentication for each node and it is by far the most robust in nature, which in turn makes it non-scalable, non-flexible and unable to self-organize. Group keying on the other hand is more robust than network keying. It allows group collaboration and multi-cast. It is able to self-organize with in cluster, but cluster formation information is application dependent. It also lacks efficient storage for group keying in IEEE 802.15.4. One of the promising WSN key distribution mechanisms is due to Eschenauer and Gligor (Eschenauer, L. & Gligor, V.D, 2002)). This protocol is simple, elegant and provides effective tradeoff between robustness and scalability. In this scheme a large pool of keys are generated (eg: 10,000 keys). Randomly take 'K' keys out of the pool to establish a key ring (K << N). Path key discovery is made When two nodes communicate they search for a common key within the key ring by broadcasting their identities (ID's) of the keys they have. Let M be the number of distinct cryptographic keys that can be stored on a client node. At the pre-deployment phase, a random pool of keys K out of the total possible key space is chosen. For each node, M keys are randomly selected from the key pool K and stored into the node's memory. This set of M keys is called the node's key ring. The number of keys in the key pool, |K|, is chosen such that two random subsets of size M in K shares at least one key with some probability p. After the client nodes are deployed, a key-setup phase is performed. The nodes first perform key-discovery to find out with which of their neighbors they should share a key. This key discovery is securely performed by Merkle puzzle policy (Merkle. (1978)), where each client node issues M client puzzles (one for each of the M keys) to each neighboring node. Any node that responds with the correct answer to the client puzzle is thus identified as a trusted client, who knows the associated key. Client nodes which discover that they contain a shared key in their key rings then verify that their neighbor actually holds the key through a challenge-response protocol. The shared key then becomes the key for that link. After key-setup is complete, a connected graph of secure links is formed.

One needs to find the right parameters such that the graph generated during the key-setup phase is connected. Consider a random graph G (n, $p_c$) a graph of n clients for which the probability that a link exists between any two nodes is $p_c$. Erdos and Renyi showed that for monotone properties of a graph G (n, $p_c$) there exists a value of $p_c$ over which the property exhibits a "phase transition", i.e., it abruptly transitions from "likely false" to "likely true". So, it is possible to calculate some expected degree d for the vertices in the graph such that the graph is connected with some high probability c. Eschenauer and Gligor calculated the necessary expected node degree d in terms of the size of the network n as:

$$d = \left(\frac{n-1}{n}\right)(\ln(n) - \ln(-\ln(c)))$$

From the formula, d (degree of the client node) = O(log n). It can be observed that the key distribution we presented is a generalized one and it can be deployed in multi-hop network. The scheme is scalable and it requires less than N-1 keys to be stored. But it lacks authentication process and does not clearly define any process for revoking or refreshing keys. The dynamic handshaking process prevents any form of data aggregation (eg: one

event detected by two neighboring nodes will result in two separate signals.). it provides no support for collaborative operations and no node is guaranteed to have common key with all of its neighbors, there is a chance that some nodes are unreachable. It also fails to satisfy security requirement authentication and operational requirement accessibility. LEAP is another important key management scheme which needs mentioning. LEAP (Zhu, et al. (2003)) uses four types of keys: Individual, group, cluster and pairwise shared keys. The authentication mechanism known as μ-TESLA is used for the broadcast authentication of the sink node, which ensures that the packets sent with the group are from the sink node only. It also employs one-way hash-key mechanism for source packet authentication. LEAP uses a pre-distribution key to help establish the four types of keys. The individual key is first established using a function of a seed and the ID of the node. Then nodes broadcast their IDs. The receiving node uses a function, seeded with an initial key, to calculate the shared key between it and all of its neighbors. Thirdly, the cluster key is distributed by the cluster head using pairwise communication secured with the pairwise shared key. Lastly for distributing the network-wide group key, the sink node broadcasts it in a multihop cluster-by-cluster manner starting with the closest cluster.  It has μ-TESLA and one-way key chain authentication as well as key revocation and key refreshing. The scheme is scalable and able to perform cluster communications. But it works on the assumption that the sink node is never compromised.

Another threat needs to be considred is physical tempering. It can be noted the sensor nodes are embdedded platform, so we have to provide platform security, which is temper proof. Recently, good amount of development has taken place in embedded platform security. Among the commercial relaeses, Trusted Platform Module by Atmel and Trustzone by ARM are worth mentioning. Trusted platform module (TPM) is to provide the minimal hardware needs to build a trusted platform in software. While usually implemented as a secure coprocessor, the functionality of a TPM is limited enough to allow for a relatively cheap implementation – at the price that the TPM itself does not solve any security problem, but rather offers a foundation to build upon. Thus, such a module can be added to an existing architecture rather cheaply, providing the lowest layer for larger security architecture. The main driver behind this approach is the Trusted Computing Group (TCG), a large consortium of the main players in the IT industry, and the successor to the Trusted Computing Platform Alliance (TCPA). TrustZone consists of a hardware-enforced security environment providing code isolation, together with secure software that provides both the fundamental security services and interfaces to other elements in the trusted chain, including smartcards, operating systems and general applications. TrustZone separates two parallel execution worlds: the non-secure 'normal' execution environment, and a trusted, certifiable secure world. TrustZone offers a number of key technical and commercial benefits to developers and end-users. TrustZone software components are a result of a successful collaboration with software security experts, Trusted Logic, and provide a secure execution environment and basic security services such as cryptography, safe storage and

integrity checking to help ensure device and platform security. By enabling security at the device level, TrustZone provides a platform for addressing security issues at the application and user levels. Below (fig. 1 & 2) we show the hardware and software architecture of ARM trustzone for reader's better understanding of a secure computing environment.
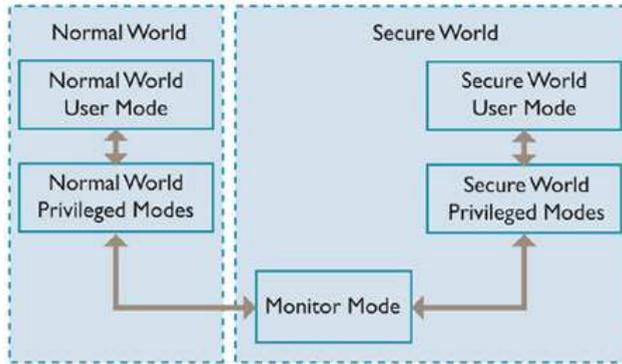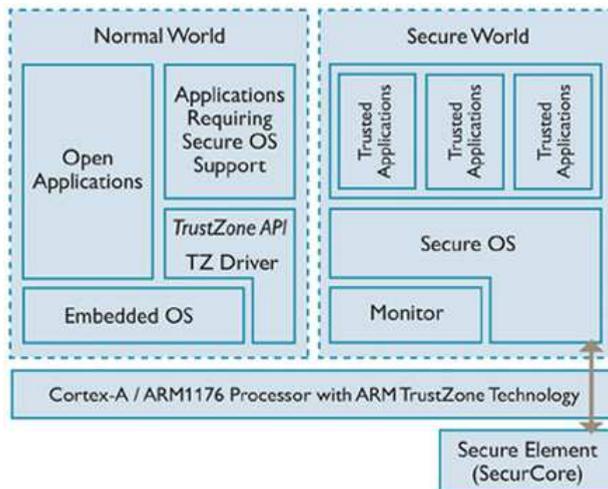


Fig. 1. Trustzone hardware architecture



Fig. 2. Trustzone software architecture

## 2.4 WSN trust and reputation management

Another important aspect of WSN security is trust and reputation management. Secure trust management policy has the responsibility that network activity can continue as securely as possible without affecting the benign entities. It has the additional duty of isolating malicious agents and also to warn benign entities. Good amount of research effort has been made to find practical and reliable trust management models (Josang, et al. (2007), Xiong, et al. (2004)). In fact, trust management which is introduced in (Blaze, et al. (1996)) defined it as "a unified approach to specifying and interpreting security policies, credentials, and

relationships which allow direct authorization of security-critical actions". In (Grandison, et al. (2002)), trust management is defined in a broader sense as: "Trust management is the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships". Traditionally trust management is studied under decentralized control environment (Li., et al. (2003). The authors described different aspects of the trust management problem. They have formulated security policies and security credentials, determined whether particular sets of credentials satisfy the relevant policies, and how deferring trust to third parties could provide better stability of the networks. Rahman and Hailes (Rahman, et al. (1997)) presented a distributed recommendation-based trust model, where conditional transitivity of trust concept is proposed. They have quantified trust as a multi-value concept.

Apart from research community, business houses and commercial organizations use and practice trust management modeling very frequently. Ebay uses reputation based trust management. It has the simple trust rating system for its users. For each successful transaction, sellers and buyers are invited to rate each other on the scale of 1. +1 is positive, 0 for neutral, -1 for negative response. Last six months ratings are taking in account by eBay to calculate a reputation of a user.

There are mainly two approaches for developing trust management system: policy based and reputation based. Policy based mechanisms employ different policy and engines for specifying and reasoning on rules for trust establishment (Stab, et al. (2004)). These mechanisms mostly rely on access control. Trust management based on distribution of certificates is presented in (Davis. (2004)) where trust is re-established by carrying out weighted analysis of the accusations received from different entities. On the other hand, reputation-based approaches have been proposed for managing trust in public key certificates, in P2P systems, mobile ad-hoc networks and in the Semantic Web. Reputation-based trust is used in distributed systems where a system only has a limited view of the information in the whole networks. It can be observed that reputation based trust management system is dynamic in nature (Duma, et al. al. (2005)) and new trust relationship is established frequently based on the malicious activities in the network. The main issues characterizing the reputation based trust management systems are the trust metric generation and the management of reputation data. In (Boukerch, et al. (2007)), agent-based trust and reputation management scheme (ATRM) for WSNs is presented. From this background we develop our reputation based trust modeling. In this model the nodes with collaboration from others form an honest opinion about each other. This model has two layers. In first layer trust model is formed against the selfish behavior of a node. This means that nodes with selfish behavior pattern will be identified, punished and if required isolated from performing any operations. The other layer is the trust modeling against malicious nodes, which falsely accuse other nodes as untrustworthy in order to disrupt the normal network activity.

In order to illustrate this, we refer to fig. 3. In this architecture, there are N number of sensor nodes and they communicate wirelessly. The sensor nodes through multi-hop routing send the sensed data to other nodes in another network or to internet through a cluster head or gateway. In order to properly maintain the self-configuring nature of the network, the nodes need to collaborate. Every node when needs to communicate to the gateway has to route the data in multi-hop. For this, it needs to take help of its neighborhood nodes. Let us consider

the case depicted in fig. 3. Node A needs to send a data to the gateway. Its neighbor consists of the nodes B, C and F. The shortest path for A to reach the gateway is through C and then C-D. But it may turn out that the shortest path is not the trusted path. Node a sends the data to C, but C maliciously drop it or send it to node I, which is another malicious node. So, for A to effectively send the data to gateway it has to first find the trustworthiness of the neighborhood nodes.  If A finds B is a trusted node, it sends the data to B for forwarding ignoring C. If A discovers F is more trusted than B, A sends the data to F.  The objective is to send the data through the most trusted node even that does not guarantee in shortest path, but this ensures reliability. We can observe that in mission critical or defense application data security and reliable transmission is often much more required than mere energy efficiency. In this case, node A needs to find out the trustworthiness of its neighborhood to update its data. Neighborhood of node A consists of node B, node C and node F. We define few terms as below:

$$T_{A \rightarrow B/C/F} = Trust\ value\ of\ A\ by\ B/C/F$$
$$R_{C \rightarrow B/F} = Reputation\ value\ of\ C\ by\ B/F$$
$$R_{B \rightarrow C/F} = Reputation\ value\ of\ B\ by\ \ C/F$$
$$R_{F \rightarrow B/C} = Reputation\ value\ of\ F\ by\ B/C$$
$$A_{A \rightarrow C} = Age\ of\ reputation\ value\ of\ A\ at\ C$$
$$A_{A \rightarrow B} = Age\ of\ reputation\ value\ of\ A\ at\ C$$
$$A_{A \rightarrow F} = Age\ of\ reputation\ value\ of\ A\ at\ F$$
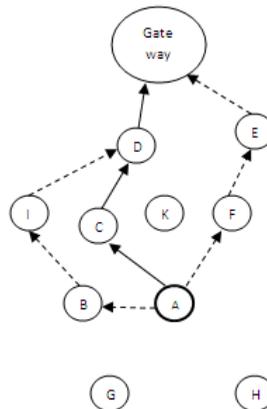


Fig. 3. Trusted node identification in WSN

In the network, individual nodes broadcast the computed reputation value of its entire neighborhood. When a particular node receives such a notification, it stores the values related to its neighbor nodes only and ignores the values of other nodes. For node A, it only accepts the reputation values of node B, C and F, i.e node A considers the reputation values $R_{C,G,H,I,F,K,F,D,E \rightarrow B}$ , $R_{B,G,H,I,F,K,F,D,E \rightarrow C}$ . and  $R_{C,G,H,I,B,K,F,D,E \rightarrow F}$  for nodes B,C and F.

Accordingly, node A finds the reputation values of other nodes C and F. It can be noted that this reputation value cannot be taken as the sole source of trustworthiness of a node. There

are other factors like the age of the reported reputation value and the previous trust value of those nodes which are to be considered to compute the overall trust factor of the node. Taking this into account, reputation value of B by C is:

$$RN_{B \to C} = R_{B \to C} \times T_{B \to C} \times A_{A \to C}$$

where, $RN_{B \to C} = New/ adjusted\ reputation\ value\ of\ B\ by\ C$

After computing the reputation value of node B, node A computes the trust value of node b as:

$$T_{B \to A} = \frac{\sum_{n \in C,F,G,H,K,D,E,I} RN_{B \to n}}{\sum_{n \in C,F,G,H,K,D,E,I}(T_{B \to n} \times A_{B \to n})}$$

Same way node A computes the trust value of node C and F (its neighborhood nodes). It should be remembered that even if node A does not require sending data, it is always required to compute the trust values of its neighborhood. Otherwise the computed trust value does not reflect the trust history of a node, which may lead to wrong judgment. Based on the latest computed trust values of its neighborhood, node A decides to send the packet through one of its neighbor nodes.

$$\text{Find Max } (T_{n \to A}), \text{where n} \in B, C, F$$

This is to find out the most trustworthy neighbor

Let,

$$T_A = \text{Max } (T_{n \to A}), \text{where n} \in B, C, F$$

Where, $T_A$ is the most trustworthy node

$$\text{Find Min } (S_{n \to A}), \text{where n} \in B, C, F$$

Where , $S_{n \to A}$ is the distance between node A to other neighborhood nodes. This is to find out the shortest possible path.

Let,

$$S_A = \text{Min } (S_{n \to A}), \text{where n} \in B, C, F$$

Where, $S_A$ is the most tshortest path node

Based on the trust values and shortest path parameters available to node A, it decides on the route to send data as per the rule below:

1.  *If, $T_A = S_A$, select that node to send data from A*
    2.   *Else if, $T_{A-1} = S_{A-1}$, select that node*
    *where $S_{A-1}$ is the node with next shortest path.*

    *where $T_{A-1}$ is the node with next best trustwhiness.*

    3.   *Else select $T_A$, irrespective of $S_A$*

Now if we consider the generalized case of N number of neighbors for node A, the selection procedure continues upto N/2, i.e. trust value from 1 upto N/2th will be compared with

that of the node with shortest path. Whichever is found the earliest, is selected, else the most trusted node is selected. In other words,

$$If, \ T_A = S_A, \ select \ that \ node \ to \ send \ data \ from \ A$$
$$Elseif, \ A = A\text{-}1, \ upto \ A = A\text{-}N/2\text{-}1$$
$$Else, \ select \ T_A$$

The above stated algorithm enforces reliability of data transfer by selecting the trusted node, even if it is required to send the data through not the shortest path. This algorithm enhances reliability to a larger extent with some extra communication cost by sending data through a non-shortest route. This is very much required for reliable transmission and to adapt to noncooperation in a collaborative computing environment. Our algorithm finds an optimized path between reliability and efficiency. Though at the end, reliability is given preference (when no matching of trusted node and shortest path is found) over efficiency.

## 3. WSN Privacy

Privacy preservation is an important issue in today's context of extreme penetration of Internet and mobile technologies. It is more important in the case of WSNs where collected data often requires in-network processing and collaborative computing. Researches in this area are mostly concentrated in applying data mining techniques to preserve the privacy content of the data. These techniques are mostly computationally expensive and not suitable for resource limited WSN nodes.

With ubiquitous connectivity, people are increasingly using electronic technologies in business-to-consumer and business-to-business settings. This in effect helps a third party to acquire the confidential and private information from various avenues. Depending upon the nature of the information, users may not be willing to divulge the individual values of records. This has lead to concerns that the private data may be misused for a variety of purposes. Privacy can be defined as the limited access to a person or a process and to all the features related to the person or the process. Privacy preservation is important from both individual as well as organizational perspectives. For example, customers might send to a remote database queries that contain private information. Two competing commercial organizations might jointly invest in a project that must satisfy both organizations' private and valuable constraints, and so on. In order to alleviate these concerns, a number of techniques have recently been proposed to perform the data mining tasks in a privacy-preserving way, which is called Privacy Preserving Data Mining (PPDM). The research of PPDM is aimed at bridging the gap between collaborative data mining and data privacy. Privacy-preserving data mining finds numerous applications in surveillance, in-network processing, which are naturally supposed to be "privacy-violating" applications. The key is to design methods (Sweeney, (2005)), which are effective without compromising on security. In the literature, number of techniques has been illustrated to effectively preserve the privacy of the source data. One of most popular method is randomization. The randomization method is a technique in which noise is added to the data to be privacy-protected. This is done to mask the attribute values of records (Agrawal, et al. (2000). The noise added to the data is sufficiently large so that individual values cannot be recovered.

Therefore, techniques are designed to derive aggregated distributions from the perturbed data values. Subsequently, data mining techniques can be developed in order to work with these aggregate distributions. The randomization method has been traditionally used in the context of distorting data by probability distribution for methods such as surveys. There are two major classes of privacy preservation schemes are applied. One is based on data perturbation techniques, where certain distribution is added to the private data. Given the distribution of the random perturbation, the aggregated result is recovered. In another technique, randomized data is used to data to mask the private values. However, data perturbation techniques have the drawback that they do not yield accurate aggregation results. It is noted by Kargupta et al. (Kargupta, et al. (2005)) that random matrices have predictable structures in the spectral domain. This predictability develops a random matrix-based spectral-filtering technique which retrieves original data from the dataset distorted by adding random values. There are two types data perturbation. In additive perturbation, randomized noise is added to the data values. The overall data distributions can be recovered from the randomized values. Another is multiplicative perturbation, where the random projection or random rotation techniques are used in order to perturb the values. In tune of their argument, we can apply the second technique of masking the private data by some random numbers to form additive perturbation.

Our one of the objectives of privacy preserved secured data aggregation falls under the broad concept of Secure Multiparty Computation (SMC) (Goldreich. (2002)). SMC and privacy preservation are closely related, particularly when some processing or computation is required on the data records. Historically, the SMC problem was introduced by Yao (Yao, et al. (2008)), where a solution to the so-called Yao's Millionaire problem was proposed. In general SMC problem deals with computing any (probabilistic) function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output. Consider a system model (fig. 4). There are N numbers of source nodes. Each source i owns a value $x_i$ which it is not willing to share with other parties. Suppose that the sum is in the range [0, M]. Our objective is to find out the sum X privately without revealing the private data $x_i$, i=1,2, … , N to each other as well as to the server.

$$X = \sum_{i=1}^{N} x_i$$

The process is initiated by the server. The server randomly chooses one of the source nodes and signals it to initiate the process. The source node first chosen by the server is denoted by $c_1$. This node possesses its private data $x_1$ and it generates one random number $r_1$ between the range [0, M], which is denoted as $r_1$. It then computes $R_1$.

$$R_1 = (r_1 + x_1)modP$$

where P is an arbitrarily large number

After computing $R_1$, the source node $c_1$ performs neighborhood discovery to find out the other source nodes it is connected to. This information $c_1$ passes to the server. Server keeps the knowledge of the nodes already participated. If the source nodes connected to $c_1$ is not already participated, the server randomly chooses one of those non-participated source nodes and sends that message to $c_1$. Let this next source node be $c_2$. Now, accordingly $c_1$ passes $R_1$ to $c_2$.

The source node $c_2$ computes $R_2$.

$$R_2 = (R_1 + x_2) mod P$$

The source node follows the same procedure as c1 and sends $R_2$ to $c_3$. This way $c_N$ is reached, which computes $R_N$.

$$R_N = (R_{N-1} + x_N) mod P$$

The server, when it finds out that all the nodes are participated, it asks the last node to send $R_N$ to it. Server now directs the first source node $c_1$ to compute the summation as:

$$X = (R_N - r_1) mod P$$

The source node after computing the summation sends that value to the server. The server may process it or sends that value for further processing.

Ukil and Sen (Ukil & Sen, (2009)) considers a scenario where data aggregation needs to be done in privacy-preserved way for distributed computing platform. There are number of data sources which collect or produce data. The data collected or produced by the sources is private and the owner or the source does not like to reveal the content of the data. But the collected data from the source is to be aggregated by an aggregator, which may be a third party or part of the network, where the data sources belong. The data sources do not trust the aggregator. So the data needs to be secure and privacy protected. The computation for the aggregation is based on the concept of SMC. SMC allows parties with similar background to compute results upon their private data, minimizing the threat of disclosure. Consider a set of parties who neither trust each other, nor the channels by which they communicate. Still, the parties wish to correctly compute some common function of their local inputs, while keeping their local data as private as possible. Generally, this problem can be seen as a computation of a function f (x1, x2, ..., xn) on private inputs x1, x2, ...,xn in a distributed network with n participants where each participant i knows only its input xi and no more information except output f (x1, x2, ..., xn) is revealed to any participant in the computation. In this case the function is SUM.In this scheme, the property of modular arithmetic to recover the aggregated value is considered and data privacy is preserved through randomization process. The security part is handled by random key pre-distribution method which is modified version of (Eschenauer, L. & Gligor, V.D, 2002). The scheme is simple in nature with low computational complexity, which makes it suitable for practical implementation particularly in the case where the source nodes do not have much computational capabilities.
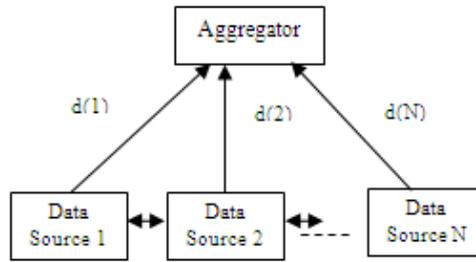
Fig. 4. SMC scheme illustration

The aggregation methods of privacy-preservation are dealt well in (Conti, et al. (2009)). In (He, et al. (2007)), He et.al. propose schemes to achieve data aggregation while preserving privacy. The scheme they proposed, CPDA (Cluster-based Private Data Aggregation) performs privacy-preserving data aggregation in low communication overhead with high computational overhead. This privacy-preservation data aggregation policy is based on the additive property of the polynomial. The objective of this algorithm is that the server or the aggregator can not make out the individual content of the data sent be the sink node. In the system model described, the friend pairs' data are aggregated together. After receiving the aggregated data of all the friend pair the server sends that to the base station. It is shown in the Fig. 5. In order to illustrtae this, we assume server/aggregator as node 'A' and two sink nodes of the friend pair is 'S1' and 'S2'.This algorithm consists of two parts:

1. Value distortion: Let the data values in the sink node S1 and S2 be x and y and z be the dummy variable at the aggregator node 'A'. In the first step, the server/aggregator sends three seeds a,b and c to the friend pairs. Based on that A computes

$$\alpha_{S1}^A = z + R_1^A b + R_2^A b^2$$
$$\alpha_{S2}^A = z + R_1^A c + R_2^A c^2$$
$$\alpha_A^A = z + R_1^A a + R_2^A a^2$$

where $R_1^A$ and $R_2^B$ are two random numbers generated by A.
Similarly, S1 computes

$$\alpha_{S1}^{S1} = x + R_1^{S1} b + R_2^{S1} b^2$$
$$\alpha_A^{S1} = x + R_1^{S1} a + R_2^{S1} a^2$$
$$\alpha_{S2}^{S1} = x + R_1^{S1} c + R_2^{S1} c^2$$

Similarly S2 computes

$$\alpha_A^{S2} = y + R_1^{S2} a + R_2^{S2} a^2$$
$$\alpha_{S1}^{S2} = y + R_1^{S2} b + R_2^{S2} b^2$$
$$\alpha_{S2}^{S2} = y + R_1^{S2} c + R_2^{S2} c^2$$

where $R_1^{S1}$ and $R_2^{S1}$ are two random numbers generated by sink node S1, $R_1^{S2}$ and $R_2^{S2}$ are other two random numbers generated by sink node S2. After that, the calculated, $\alpha_{S1}^A$ and $\alpha_{S2}^A$ are sent to sink node S1 and sink node S2 by A, securely as described earlier. Similarly,

$\alpha_A^{S1}$ and $\alpha_{S2}^{S1}$ are sent to sink node S2 and A by sink node S1 and $\alpha_A^{S2}$ and $\alpha_A^{S2}$ and $\alpha_{S1}^{S2}$ are sent to A and sink node S1 by sink node S2.

2. Value aggregation: After the private data values (x and y) are distorted, all the nodes aggregates the values available to them and generates aggregated result. Sink node calculates $\Psi_{S1}$ , sink node S2 calculates $\Psi_{S2}$ and A calculates $\Psi_A$.

$$\Psi_A = \alpha_A^A + \alpha_A^{S1} + \alpha_A^{S2} = (x + y + z) + R_1 a + R_2 a^2$$
$$\Psi_{S1} = \alpha_{S1}^A + \alpha_{S1}^{S1} + \alpha_{S1}^{S2} = (x + y + z) + R_1 b + R_2 b^2$$
$$\Psi_{S2} = \alpha_{S2}^A + \alpha_{S2}^{S1} + \alpha_{S2}^{S2} = (x + y + z) + R_1 c + R_2 c^2$$

where, $R_1 = R_1^A + R_1^{S1} + R_1^{S2}$ and $R_2 = R_2^A + R_2^{S1} + R_2^{S2}$. These aggregated results from sink node S1 and sink node S2 are securely sent to the aggregator A. Now, the aggregator has the simple task to solve the above equation for (x+y+z) with the knowledge of the values of a,b,c and $\Psi_A$ , $\Psi_{S1}$ and $\Psi_{S2}$ . After solving for D = x+y+z, node A internally knows its own data z, so it can find out the result (x+y).
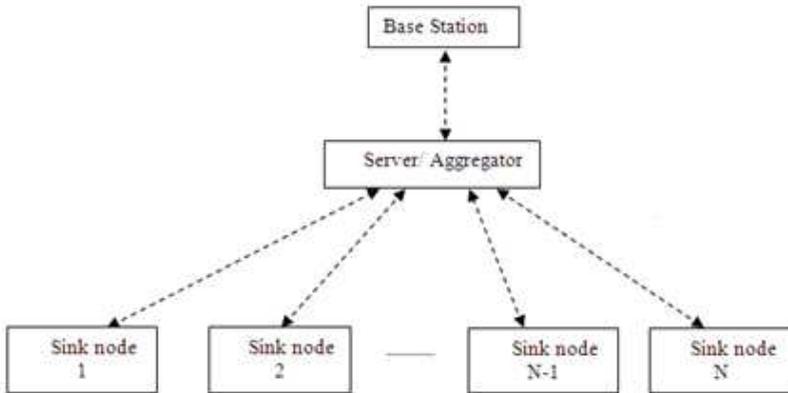


Fig. 5. CPDA scheme illustration

The privacy-preserving data aggregation scheme by Conti et al. (Conti et al. (2009)) first establishes twin keys for different pairs of sensor nodes in a network. Twin key establishment is an anonymous process that prevents each node in a pair from deriving the identity of the other node with which it is sharing a twin key. Then, for each aggregation phase, it uses an anonymous liveness announcement protocol to declare the liveness of each twin key. In the end, during the aggregation phase, each node encrypts its own value by adding shadow values computed from the lively twin keys it holds. In this way, the contribution of the shadow values for each twin key will cancel out each other and the correct aggregated result is finally obtained. Data Aggregation Different Privacy-levels Protection (DADPP) (Yao, et al. (2008))) offers different levels of data aggregation privacy based on different node numbers for pre-treating the data. This protocol is inspired by the work of Shao et al. in terms of different levels of privacy as well as the CPDA in terms of the privacy achieving method (Shao et al. (2007)). In DADPP, a hierarchical wireless sensor network is first constructed in such that sensor nodes form several clusters each of which

has a fixed cluster head below the energy efficient Base sation. According to the desired privacy level, all nodes within the same cluster are partitioned into multiple groups belonging to the same privacy level. Data are pretreated only in the same group and privacy levels are defined by the size of groups. The lowest privacy level consists of partitioned groups that have at least 3-sensor-nodes. The upper privacy level corresponds to portioned groups with 4-sensor-nodes. By analogy, if all sensor nodes of a cluster belong to a single group, they consider this case as the highest privacy level. The data aggregation process is similar to that of the CPDA. First, original data are pretreated in each group. Secondly, the cluster head aggregates all pretreated data. Finally, data are aggregated on the plane of the cluster head up to the BS. The hierarchical wireless sensor network is illustrated in Figure 6. Although DADPP reduces traffic by partitioning a cluster with n sensor nodes into multiple in-networks with pretreatment of groups according to the desired privacy-levels, it suffers from the inherent high communication and computation overheads. Furthermore, these overheads increase with increasing privacy level.
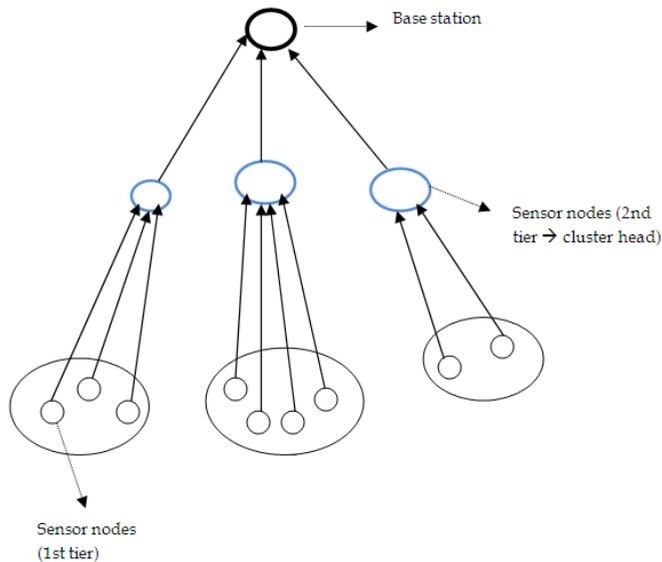


Fig. 6. Hierarchical WSN

Zhang et al. (Zhang, et al. (2008)) proposed the Perturbed Histogram-based Aggregation (PHA) to preserve privacy for queries targeted at special sensor data or sensor data distribution. The perturbation technique is applied to hide the actual individual readings and the actual aggregate results sent by sensor nodes. For this, every sensor node is preloaded with a unique secret number which is known exclusively by the sink and the node itself. Sensor nodes and the sink form a tree. The basic idea of PHA is to generalize the values of data transmitted in a WSN, such that although individual data content cannot be decrypted, the aggregator can still obtain an accurate estimate of the histogram of data distribution and thereby approximate the aggregates. In particular, before transmission, each sensor node first uses an integer range to replace the raw data. Next, with a certain

granularity, the aggregator plots the histogram for data collected and then estimates aggregates such as MIN, MAX, Median and Histogram. Although the PHA supports many data aggregation functions, it has the following disadvantages. First, the final aggregated result is an approximation value of the sensor data rather than the real data. Secondly, the PHA requires a large size payload (message/data) because all sensor data need to be replaced by an integer range. Moreover, the bandwidth consumption of this protocol increases as the number of ranges increases. Finally, storing interval ranges to replace the original data consumes a significant amount of memory. To address Privacy-preserving Integrity-assured data Aggregation (PIA) for WSNs, recently, Taban et al. proposed four distinct symmetric-key solutions (Taban et al. (2009)). In their single aggregator model, an aggregator node is used as an intermediary between the user (i.e., a third party) and the sensor nodes that aggregates the sensor data and forwards the query response to the user. The problem is that the user wants to verify the integrity of the received aggregate value whereas the network owner does not want the user to access the original data. Privacy Homomorphism (PH) has a special feature that allows arithmetic operations to be performed on cipher-text without decryption. This technique is fast and resource-efficient for privacy-preserving data aggregation, but it has a limitation that it performs only addition and multiplication operations. Before sensor data are sent to the aggregators, they are encrypted by using the respective keys of sensor nodes and they are added or multiplied without decryption. Concealed Data Aggregation (CDA) (Ferrer. (2002)) is a type of PH scheme, which conceals the process of data aggregation in WSN by using Domingo-Ferrer's (DF) approach ( Deng, et al. (2006)). In this protocol, each sensor node splits its data into d parts (d ≥ 2), encrypts them by using a public key and transmits them to the aggregator node. The aggregator node operates on the encrypted data, computes an aggregated value from the data without decryption and sends it to the sink.

Context-oriented privacy protection focuses on protecting contextual information, such as the location (Xi. Et al. (2006)) and timing (Kamat, et al. (2007)) information of traffic transmitted in a WSN. Location privacy concerns may arise for such special sensor nodes as the data source (Mehta, et al. (2007)) and the base station (Jian, et al. (2007). Timing privacy, on the other hand, concerns the time when sensitive data is created at data source, collected by a sensor node and transmitted to the base station. This type of privacy is also of primary importance, especially in the mobile target tracking application of WSNs, because an adversary with knowledge of such timing information may be able to pinpoint the nature and location of the tracked target without learning the data being transmitted in the WSN. Furthermore, the adversary may be able to predict the moving path of the mobile target in the future, violating the privacy of the target. Similar to data-oriented privacy, context-oriented privacy may also be threatened by both external and internal adversaries. Nonetheless, existing research has mostly focused on defending against external adversaries, because such adversaries may be able to compromise context privacy easily by monitoring wireless communication. Within the category of external adversaries, one can further classify adversaries into two categories, local attackers and global attackers; based on the strength of attacks an adversary is capable of launching. Local attackers can only monitor a local area within the coverage area of a WSN, and therefore have to analyze traffic hop-by-hop to compromise traffic context information. On the other hand, a global attacker has the capability (e.g., a high-gain antenna) of monitoring the global traffic in a WSN. One

can see that a global attacker is much stronger than a local one. To further protect the location of the data source, fake data packets can be introduced to perturb the traffic patterns observed by the adversary. In particular, a simple scheme called Short-lived Fake Source Routing was proposed in (Kamat, et al. (2005)) for each sensor to send out a fake packet with a pre-determined probability. Upon receiving a fake packet, a sensor node just discards it. Although this approach perturbs the local traffic pattern observed by an adversary, it also has limitations on privacy protection. Specifically, to maintain the energy-efficiency of the WSN, the length of each path along which fake data is forwarded is only one hop, therefore, an adversary is able to quickly identify fake paths and eliminate them from consideration.

Another aspect of privacy preservation is anonymity, where the identity of the origin and/or the destination of a conversation is hidden from adversaries unless it is intentionally disclosed by the user. Ring signature (Rivest, et al. (2001)) is a signer-ambiguous signature scheme, first introduced by Cramer et al in 1994. With ring signature, a set of possible users (signers) should be specified and each user should be associated with the public key of some standard signature scheme such as RSA. To generate a ring signature, the actual signer declares an arbitrary set of possible signers that must include himself, and computes the signature of any message by himself using only his secret key and the other's public keys. Ring signatures can be verified by the intended recipient as a valid signature from one of the declared signers, without revealing exactly which signer actually produced the signature. Ring signatures provide an elegant way to leak authoritative secrets in an anonymous way and can be used to solve multiparty computation problems. In the case of anonymous access authentication, ring signatures allow a legitimate user to hide his true identity among an arbitrarily selected set of other users. The non-linkability of multiple transactions of the same user is also well protected.

## 4. Conclusion

In this chapter, we present on the issues of security and privacy in WSN. We provide a comprehensive study regarding the requirements, different kind of well-known attacks and some of the proposed solution to counter the security attacks on WSN. We also emphasise on the embedded device security where industry has recently given a lot of attention. We have touched upon the concept of trust and reputation based security analysis in WSN. In fact, we attempt to make the main focus of this chapter on privacy preservation aspects of WSN. It is found that WSN security is well-researched compared to the privacy preserving issues. So, our endeavour was to bring that privacy protection problem in WSN. In that regard, we have provided detailed description of some of the important schemes and present the privacy preservation of WSN both from functional and requirement perspectives.

## 5. References

Chan, H.; Perrig, A. & Song, D. (2003). *Random key predistribution schemes for sensor networks*, Proceedings IEEE Symposium on Security and Privacy, pp. 197 - 213. IEEE Computer Society.

Liu, D.; Ning, P. & Li, R. (2005). *Establishing pairwise keys in distributed sensor networks*, ACM Trans.Inf. Syst. Secur., vol. 8, no. 1, pp. 41–77.

Newsome, J.; Shi, E.; Song, d. & Perrig, A. (2004). *The Sybil Attack in Sensor Networks: Analysis & Defenses*, IEEE International Workshop on Information Processing in Sensor Networks (IPSN'04), Berkeley, USA.

Weiser, M. (1991). *The Computer for the Twenty First Century*, Scientific American, pp. 94-104, September, 1991.

Karlof, C. & Wagner, D. (2003). *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasure*, Ad-Hoc Networks, vol. 1, no. 2-3, pp. 293-315, Elsevier, September 2003.

Law, Y. W.; Doumen, J. & Hartel, P. (2006). *Survey and Benchmark of Block Ciphers for Wireless Sensor Networks*, ACM Transactions on Sensor Networks, vol. 2, no. 1, pp. 65-93, February, 2006.

Alarifi, A. & Du, W. (2006). *Diversifying Sensor Nodes to Improve Resilience against Node Compromise*, 2006 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06),Alexandria, USA, October 2006.

Gaubatz, G.; Kaps, J.P.; Öztürk,E. & Sunar, B. (2005*). State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks*, IEEE International Workshop on Pervasive Computing and Communication Security (PerSec'05), Hawaii, USA, March 2005.

Shi, E. & Perrig, A. (2004). *Designing secure sensor networks*, Wireless Communication Magazine, vol. 11, no. 6, pp. 38-43, December 2004.

Wang, X.; et al. (2005). *Search-based physical attacks in sensor networks: modeling and defense*, Technical report, Department of Computer Science and Engineering, Ohio State University, February 2005.

Wang, X.; et al. (2004). *Sensor network configuration under physical attacks*, Technical report (OSU-CISRC-7/04-TR45), Department of Computer Science and Engineering, Ohio State University, July 2004.

Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y. & Cayirci,E. (2002). *A survey on sensor networks*, IEEE Communications Magazine, vol. 40,no. 8, pp. 102-114, August 2002.

Wood, A.D. & Stankovic,J.A. (2002). *Denial of service in sensor networks*, IEEE Computer, vol. 35, no. 10, pp. 54-62.

Hu, Y.; Perrig,A. & Johnson,D.B. (2003). *Packet Leashes: A defence Against Wormhole Attacks in Wireless adhoc Networks*, IEEE INFOCOM, vol. 3, pp. 1976 – 1986.

Newsome, J.; Shi, E.; Song, D. & Perrig, A. (2004). *The sybil attack in sensor networks: analysis & defenses*, Proceedings of the third international symposium on Information processing in sensor networks, pp. 259–268. ACM Press.

Douceur, J. (2002). *The sybil attack*, Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), February 2002.

Deng, J.; Han, R. & Mishra, S. (2004). *Countermeasuers against traffic analysis in wireless sensor networks*, Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

Awerbuch, B.; et al. (2004). *Mitigating Byzantine Attacks in Ad HocWireless Networks*, Technical Report version 1, March 2004.

Hu, Y.; Perrig, A. & Johnson, D.B. (2003). *Rushing Attacks and Defense in Wireless ad Hoc network Routing protocols*, ACM workshop on Wireless Security, pp. 30 – 40, 2003.

Raymond, D.; et al. (2006). *Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocols*, Proceedings of 7th Annual IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop (IAW), pp. 297–304.

Karlof, C. & Wagner, D. (2003). *Secure routing in wireless sensor networks: Attacks and countermeasures*, Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, May 2003, pp. 113-127.

B. Schneier. (1996). *Applied Cryptography*, Second Edition, John Wiley & Sons.

Kobiltz, N. (1987). *Elliptic curve cryptosystems*, Mathematics of Computation, vol. 48, pp. 203-209.

Liu, A. & Ning, P. (2005). *TinyECC: Elliptic Curve Cryptography for Sensor Networks* (version 0.1), September 2005.

Eschenauer, L. & Gligor, V.D. (2002). *A key-management scheme for distributed sensor networks*, 9th ACM Conference on Computer and Communication Security, pp. 41–47.

Merkle, R. (1978). *Secure communication over insecure channels*, Communications of the ACM, vol. 21, no.4, pp. 294–299.

Spencer, J. (2000). *The Strange Logic of Random Graphs*, Algorithms and Combinatorics, no.22, 2000.

Zhu, S.; Setia, S. & Jajodia, S. (2003). *LEAP: Efficient security mechanism for large –scale distributed sensor networks*, Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 62-72, New York, NY, USA, ACM Press.

www.atmel.com

www.arm.com

https://www.trustedcomputinggroup.org

Sweeney, L. (2005). *Privacy Technologies for Homeland Security, Testimony before the Privacy and Integrity Advisory Committee of the Department of Homeland Security*, Boston, MA, Sep. 28, 2005.

Agrawal, R. & Srikant, R. (2000). *Privacy-Preserving Data Mining*, ACM Sigmod, pp. 439–450.

Kargupta, H.; Dutta, S.; Wang, Q. & Sivakumar, K. (2005). *Random-data perturbation techniques and privacy-preserving data mining*, Knowledge and Information Systems, vol. 7, no. 4, pp. 387–414.

Goldwasser, S. (1997). *Multi-party computations: Past and present*, 16th Annual ACM symposium on Principles of distributed computing, pp. 1–6.

Conti, M.; et al. (2009). *Privacy-preserving robust data aggregation in wireless sensor networks*, Security and Communication Networks (Wiley), vol. 2, pp. 195–213.

Wright, M.; Adler, M.; Levine, B.N. & Shields, C. (2003). *Defending anonymous communications against passive logging attacks*, IEEE Symposium on Security and Privacy, pp. 28–41.

Eschenauer, L. & Gligor, V.D. (2002). *A key-management scheme for distributed sensor networks*, 9th ACM Conference on Computer and Communication Security, pp. 41–47.

Goldreich, O. (2002). *Secure multi-party computation*, Working Draft, First version posted in June, 1998 and final revision posted in Oct, 2002.

Yao, A. (1982). *Protocols for secure computations*, 23rd Annual Symposium on Foundations of Computer Science, pp. 160–164.

He, W.; Liu, X.; Nguyen, H.; Nahrstedt, K. & Abdelzaher, T. (2007). *PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks*, IEEE Infocom, pp. 2045–2053.

Rivest, R.; Shamir, A. & Tauman, Y. (2001). *How to leak a secret*, Advances in Cryptology - ASIACRYPT 2001.

Conti, M.; Zhang, L.; Roy, S.; Pietro, R.D.; Jajodia, S. & Mancini, L.V. (2009). *Privacy-preserving robust data aggregation in wireless sensor networks*, Secur. Commun. Netw, no. 2, pp.195–213.

Yao, J.; & Wen, G. (2008). *Protecting classification privacy data aggregation in wireless sensor networks,* Proceedings of the 4th International Conference on Wireless Communication, Networking and Mobile Computing, WiCOM, Dalian, China, October 12–14, 2008; pp. 1–5.

Shao, M.; Zhu, S.; Zhang, W. & Cao, G. (2007). *Pdcs: Security and privacy support for data-centric sensor networks*, Proceeding of 26th IEEE International Conference on Computer Communications, INFOCOM, Anchorage, AK, USA, May 6–12, 2007; pp. 1298–1306.

Zhang, W.S.; Wang, C. &Feng, T.M. (2008). *GP2S: Generic privacy-preservation solutions for approximate aggregation of sensor data, concise contribution*, Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom, Hong Kong, China, March 17–21, 2008; pp.179–184.

Taban, G. & Gligor, V.D. (2009). *Privacy-preserving integrity-assured data aggregation in sensor networks*, Proceeding of International Symposium on Secure Computing, SecureCom, Vancouver, Canada, August 29–31, 2009; pp. 168–175.

Ukil, A. & Sen, J. (2010). *Secure Multiparty Privacy Preserving Data Aggregation by Modular Arithmetic*, International conference on parallel, distributed, and Grid Computing, pp. 329 - 334, Oct, 2010.

Sen, J. (2009). *A Survey on Wireless Sensor Network Security*, International Journal of Communication Networks and Information Security (IJCNIS), vol. 1, no. 2, pp.55 - 78 , Aug. 2009.

Girao, J.; Westhoff, D. & Schneider, M. (2005). *CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks*, In Proceedings of IEEE International Conference on Communications, ICC, Seoul, Korea, May 16–20, 2005; volume 5, pp. 3044–3049.

Domingo-Ferrer J. (2002). *A provably secure additive and multiplicative privacy homomorphism*, Proceedings of the 5th International Conference on Information Security, Sao Paulo, Brazil, September 30–October 2, 2002; pp. 471–483.

Deng, J.; Han, R. & Mishra, S. (2006). *Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks*, Pervasive and Mobile Computing Elsevier, vol. 2, no. 2, pp.159–186.

Xi, Y.; Schwiebert, L. & Shi, W.S. (2006). *Preserving source location privacy in monitoring-based wireless sensor networks*, Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), April 2006.

Kamat, P.; Xu, W.Y.; Trappe, W. & Zhang, Y.Y. (2007). *Temporal privacy in wireless sensor networks*, Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS 2007), June 2007, pp. 23–23.

Mehta, K.; Liu, D.G. & Wright, M.(2007). *Location privacy in sensor networks against a global eavesdropper*, Proceedings of the IEEE International Conference on Network Protocols (ICNP 2007), October 2007, pp. 314–323.

Jian, Y.; Chen, S.G.; Zhang, Z. & Zhang, L. (2007). *Protecting receiver-location privacy in wireless sensor networks*, Proceedings of the 26th International Conference on Computer Communications (INFOCOM 2007), May 2007, pp. 1955–1963.

Kamat, P. Zhang, Y.Y.; Trappe, W. & Ozturk, C. (2005*). Enhancing source location privacy in sensor network routing*, Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005), June 2005, pp. 599–608.

Grandison, T. & Sloman, M. (2000). *A Survey of Trust in Internet Applications*, IEEE Communications Surveys and Tutorials, vol. 3, no. 4, September 2000.

Jøsang, A.; Ismail, R. & Boyd, C.  (2007). *A survey of trust and reputation systems for online service provision*, Decision Support Systems, vol. 43, no. 2, pp.618–644, March 2007.

Blaze, M. Feigenbaum, J. & Lacy, J. (1996). Decentralized trust management, In Proceedings of  IEEE Conference on Security and Privacy.

Grandison T. & Sloman, M. (2002). *Specifying and analysing trust for internet applications; Towards The Knowledge Society: eCommerce, eBusiness, and eGovernment*, The Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002), IFIP Conference pp. 145–157.

Li, N. & Mitchell, J.C. (2003). *Datalog with Constraints: A Foundation for Trust-management Languages*, Proceedings of the Fifth International Symposium on Practical Aspects of Declarative Languages pp. 58–73, January 2003.

Abdul-Rahman, A. & Hailes, S. (1997).  *A distributed trust model,* Proceedings of New Security Paradigms Workshop, ACM, pp. 48 – 60, 1997.

*www.ebay.com.*

Staab, S.; et al. (2004). *The pudding of trust*, IEEE Intelligent Systems, vol. 19, no. 5, pp.74–88.

Davis, C. (2004). *A localized trust management scheme for ad-hoc networks*, 3rd international conference on Networking.

Duma, C.; Shahmehri, N. & Caronni, G. (2005). *Dynamic trust metrics for peer-to-peer systems*, Proc. of 2nd IEEE Workshop on P2P Data Management, Security and Trust, August 2005.

Boukerch, A.; Xu, L. & EL-Khatib,K. (2007). *Trust-based Security for Wireless Ad Hoc and Sensor Networks*, Computer Communication, vol. 30, pp. 2413-2427.

Xiong, L. & Liu, L. (2004).  *PeerTrust: Supporting reputation based trust in peer to peer communities*, IEEE Transactions on Data and Knowledge Engineering, Special Issue on Peer to Peer Based Data Management, vol. 16, no. 7, pp. 843–857, July 2004.
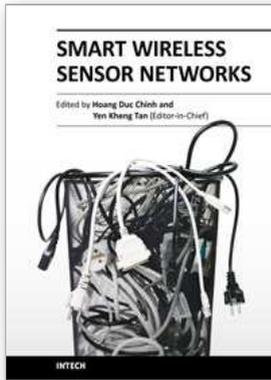
**Smart Wireless Sensor Networks**

Edited by Yen Kheng Tan

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area â€" wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodesâ€™ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of networkâ€™s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

# INTECH
open science | open minds

www.intechopen.com