2009

Message integrity model for wireless sensor networks

Haider Qleibo
Iowa State University

Recommended Citation

Qleibo, Haider, "Message integrity model for wireless sensor networks" (2009). *Graduate Theses and Dissertations*. 10637.
https://lib.dr.iastate.edu/etd/10637

Message integrity model for wireless sensor networks

by

Haïder W. Qleibo

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Doug Jacobson, Major Professor
Jim Davis
Tom Daniels
Mani Mina
William Jenks

Iowa State University

Ames, Iowa

2009

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my parents Samira and Wahib, who raised me to understand the importance of ambition and enthusiasm without which I would not have been able to complete my work. They have provided me with unconditional love, and instilled in me the values of strong work ethic. They always encouraged me to strive for the best. As for my best friend and lovely sister Mounira, I would like to extend my hand of appreciation for her continuous support in achieving my goals and dreams, and for never doubting me when I doubted myself. To Betsy and John Mayfield, who treated me like a son of their own and being there for me throughout the entire time of graduate studies.

I am sincerely grateful for my major professor Prof Doug Jacobson. It has been a great honor and privilege to work with such talented engineer, researcher, professor and a friend. He has opened doors of opportunity for me that I would not have known existed and has helped me define my career and future plans. I will forever remember him as a having a key role in shaping my future. I am also indebted to Prof Jim Davis; he had numerous obligations and commitments but always made time assist me in all my academic, professional and personal life. He had his door open to me whenever I needed someone to listen to me and bounce ideas with. I will never forget Dr. Tom Daniels, his sense of humor and broad knowledge helped me through my graduate program. I would like to express my sincere appreciation to both Prof William Jenks and Dr. Mani Mina for serving on my committee.

This journey would not have been easy without the continuous support, encouragement and understanding of Mike Bowman, Jeff Franklin and Rick Hindman. They gave me the time I needed to complete my work.

During my journey at ISU I made a number of friends that became part of my daily life; this work wouldn’t have been completed without their support. My sincere thanks go to both Abullatif Ibdah and Mohammad Mekkawy (aka MEX) who accepted me as their roommate they never had. I am grateful to my old time friends Diyaa Nammari and Mohammad Tina for constantly staying in touch with me even when I am thousands of miles away.

Last but not least, this work would not have been achieved without the support and understanding of my long lasting source of energy, Ray McCormick. Her dedication and best friendship qualities made the exhaustive times tolerable and her sound advice kept me safe.

ABSTRACT

WSNs are susceptible to a variety of attacks. These attacks vary in the way they are performed and executed; they include but not limited to node capture, physical tampering, denial of service, and message alteration. It is of paramount importance to protect gathered data by WSNs and defend the network against illegal access and malicious insertion of data that would alter the entire integrity of the system. The severe resource constraints in each sensor make it a challenge to secure the network. The need for new security ideas was the main inspiration and motivation of this research. While there has been a remarkable progress in many enabling technologies for sensor networking, the integrity of information received by the system has received less attention [50]. For instance, while many methods have been developed for self-organizing the network functions, less attention was paid to ensure high data integrity.

While most the security mechanisms focus on confidentiality, we focus on integrity and freshness of the message. In order to achieve a secure system, security has to be integrated into every component. Sometimes security is viewed as a standalone component of a system's architecture, where a separate module provides security. This separation is a very flawed approach to network security. Components designed without security in mind can become a point of attack. The integrity model describes how data items in the system should be kept valid from one state of the system to the next. To define a security model, it requires specifying both the security requirements and the threat model.

CHAPTER 1. INTRODUCTION

1.1 Wireless Sensor Networks

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it" [49]. One of the main reasons that wireless sensor networks (WSNs) have come to prominence in the past couple of years is because they hold the potential to revolutionize many segments of our economy and daily life. They are transforming the way we live, work and interact with the physical environment. WSNs are very small, inexpensive and smart. The magnitude of their impact has a wide spectrum that goes beyond environmental monitoring and conservation, but also impacts business asset management, health care and military. All this would not be possible without the recent technological advances that made it feasible to deploy small, low-power, low-bandwidth, and multifunctional wireless sensor nodes to monitor and report the conditions and events of their local environment. This kind of network could be deployed in large-scale and complex environments collecting and aggregating data in real time, transforming it to meaningful information, and keeping the end user aware of the events witnessed and prepared to take proper actions if needed. A major benefit of WSNs is helping us understand and better manage our increasingly interconnected physical world [3]. Mark Weiser, the father of Ubiquitous computing, mentioned in his article published in 1991 that computers will merge with the environment more and more until they become completely invisible to the user [4]; as a result, this is putting WSN in the front seat of Ubiquitous Computing [5].

WSN set a new paradigm for large-scale distributed systems and information gathering based on the collaborative efforts of a large number of self-organized nodes [48]. With their limited energy, computation and communication capabilities they pose unique security challenges that make current existing security mechanisms inadequate. Moreover, theses nodes are deployed in accessible terrain adding the risk of a physical attack. It only takes one compromised node to jeopardize the entire network. We can see the importance of this new field by the number of recent funding initiatives including Defense Advanced Research Projects Agency (DARPA) SENSIT program, military programs, and NSF Program Announcements. If WSNs are going to be the eyes and ears of our future society, then there is a need in asking “how can we trust the information provided by the sensor networks?” For these networks to be useful, the information they provide must be of a high integrity. The decision making process can go askew if the network provides misleading picture of the physical world and what it is sensing and reporting.

WSNs are susceptible to a variety of attacks. These attacks vary in the way they are performed and executed; they include but not limited to node capture, physical tampering, denial of service, and message alteration. It is of paramount importance to protect gathered data by WSNs and defend the network against illegal access and malicious insertion of data that would alter the entire integrity of the system. The severe resource constraints in each sensor make it a challenge to secure the network. The need for new security ideas was the main inspiration and motivation of this research. While there has been a remarkable progress in many enabling technologies for sensor networking, the integrity of information received by the system has received less attention [50]. For instance, while many methods have been developed for self-organizing the network functions, less attention was paid to ensure high data integrity.

Our research focuses on studying the characteristics of sensor networks and their behavior. We compare the current known threats and attacks facing this kind of network and look at the challenges facing the development of new security models. We discuss confidentiality of a message versus its integrity. For some application, confidentiality is not of a great importance as long as the message obtained from the sensor node maintains a certain level of integrity. Information integrity is maintained by preventing corruption of data items in a system due to either error or malicious intent. Our integrity model is based on David D. Clark and David D. Wilson integrity model [44] with primary concern of formalizing the notion of information integrity. While most the security mechanisms focus on confidentiality, we focus on integrity and freshness of the message. In order to achieve a secure system, security has to be integrated into every component. Sometimes security is viewed as a standalone component of a system's architecture, where a separate module provides security. This separation is a very flawed approach to network security. Components designed without security in mind can become a point of attack. The integrity model describes how data items in the system should be kept valid from one state of the system to the next. To define a security model, it requires specifying both the security requirements and the threat model. Our model does not discuss mobile sensors nodes or physical interference with the message like noise in the environment, congestion, or climate change.

CHAPTER 2. LITERATURE REVIEW

2.1 Definitions

Confidentiality: ensures that a transmitted message cannot be understood by anyone else and can be accessed by the intended nodes.

Data Integrity: ensures that the transmitted message is original and has not been altered throughout the transmission.

Authentication: allows for communicating parties to know the identities of each other in order to make sure they are genuine.

Availability: ensures the availability of network services whenever it is required by the intended parties.

Ubiquitous Computing: all models of ubiquitous computing (also called pervasive computing) share a vision of small, inexpensive, robust networked processing devices, distributed at all scales throughout everyday life and generally turned to distinctly common-place ends.

2.2 Wireless Sensor Networks: Design and Architecture

Sensor networks are defined as a large number of self organizing, low power, low cost wireless nodes that are deployed en masse monitoring a certain phenomenon; they could be deployed inside the phenomenon or very close to it [7]. Figure 1 shows a typical sensor network and its major components:

1. Sensor field
2. Sensor nodes
3. Sink
4. Task manager

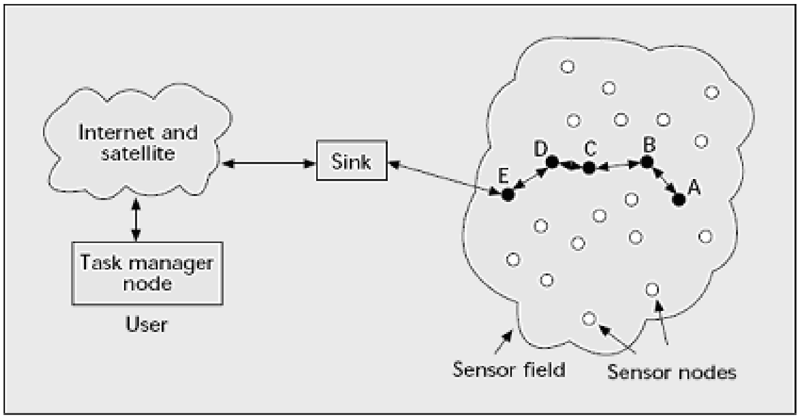

Figure 1: Sensor nodes scattered in a sensor field [9]

Each of these nodes is equipped with its own sensor, processor, memory, power source, and radio transceiver. Sometimes, depending on the application that the network is serving, a GPS serves as the 6th element (figure 2). These nodes are not tamper-proof; it is infeasible to keep them low-cost while packaging them in a tamper resilient package. Such networks are exposed to internal and external attacks [8]. The position of the sensor nodes need not to be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. Each of these nodes has the capability to collect data and route it back to a more powerful resource, referred to as sink or base station [7]. The sink is a more powerful node and it acts as a gateway to another network; it has a powerful processor with significant storage space as well as an unlimited power source that makes it outlive other nodes. All nodes transmit their aggregated data to the sink, which in return relays it to other reliable communication means and on to the task manager where user intervention is needed.

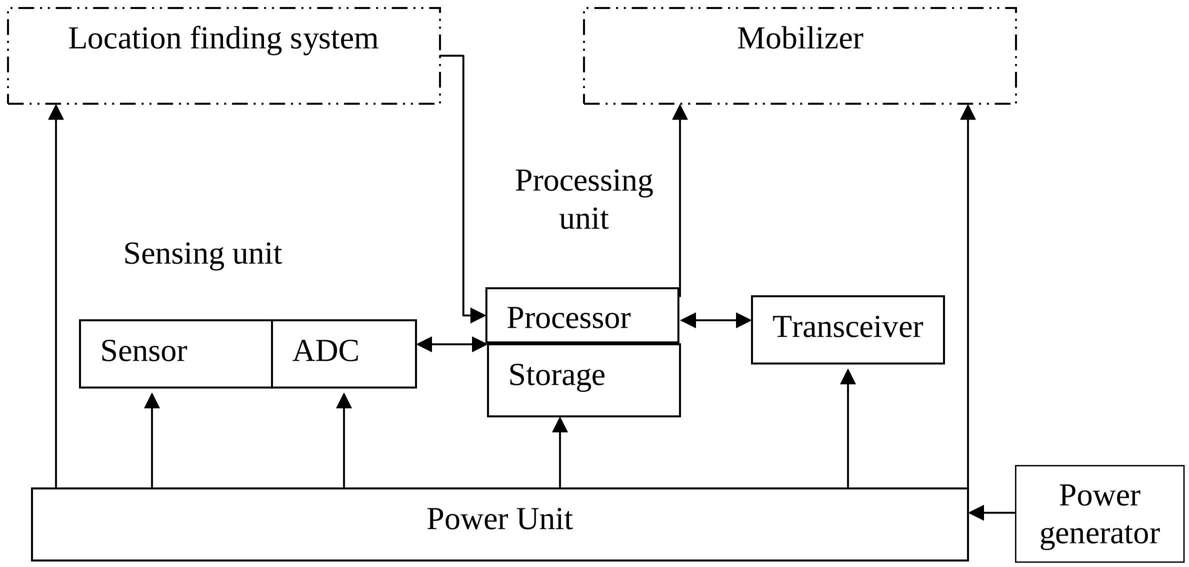

Figure 2: Sensor node components

Network topology in WSN can be configured in different topologies:

1. Single-hop star
2. Multi-hop mesh and grid
3. Tier hierarchical cluster

Single-hop star topology: Each node communicates directly with the sink. This is the simplest design for a WSN as the networking concerns are reduced to the minimum. This topology has many drawbacks; it's neither scalable nor robust. For instance, in large networks, distant nodes from the sink will have a big wireless link disadvantage.

Multi-hop mesh and grid: A larger area network, multi-hop routing is necessary as shown in figure -. Nodes can form a mesh graph and communicate with the sink. A major weakness of this topology is that for very large networks, each node has to keep a routing table in its memory of all the nodes.

Hierarchical cluster: This is the kind of network we see deployed in large areas where a group of nodes within a certain region in the network report to different cluster-heads. There are several ways to implement this kind of topology; one is that the cluster-head will be known in advance to the network and has power that would surpass the life of the nodes in its region. The other approach is to change the cluster-head for each region periodically, giving the task to each node. This has a great advantage for scalability and management; it divides a big network into separate zones where each zone aggregates its data locally before sending it to the cluster-head. Then the cluster-head will send the data to either another cluster-head or directly to the sink.

2.2.1 Protocol Stack

Like any telecommunication device, sensor nodes has a specific network stack; research is still being conducted to determine how to optimize the protocol stack [9] (figure 3).

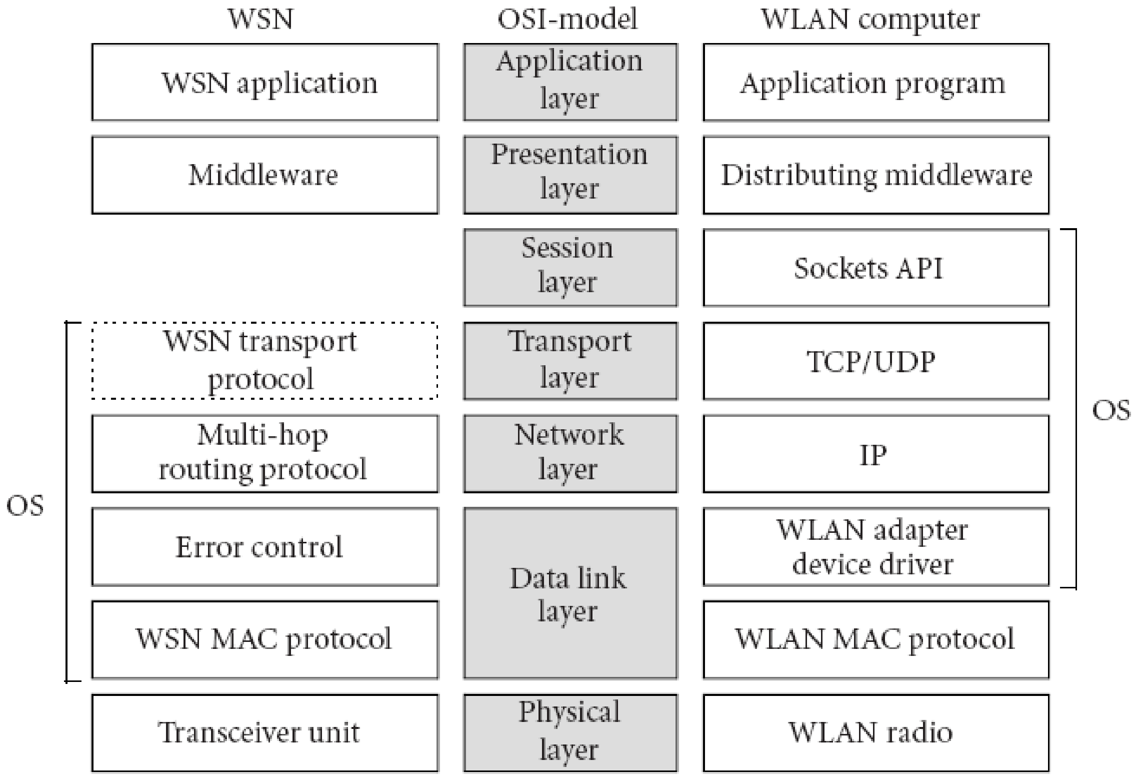

Figure 3: OSI model, WSN, and distributed system in WLAN protocol layers

Physical Layer: Responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption. Thus far, the 915 MHz ISM band has been widely suggested for sensor networks. This layer is a fertile research area and is widely unexplored; current open research issues range from power-efficient transceiver design to modulation schemes.

Data link Layer: Ensures reliable point-to-point and point-to-multi-hop connections in the network. It is responsible for medium access, error control, multiplexing of data streams and data frame detection. Conventional MAC protocols are not suited to sensor networks due to the network constraints. The MAC protocol in a wireless multi-hop self-organizing sensor network should satisfy two objectives [7]: one is to create the network infrastructure and the other is to fairly and efficiently share communication resources between other sensor nodes. Data link layer protocols include: Eavesdrop and Register (EAR) [10], CSMA-Based medium Access Protocols [11], and Self-Organized Medium Access Control for Sensor Networks (SMACS) [10]. There is considerable work and research to be conducted in this layer; some of the obvious include power saving modes of operation, error control coding schemes and MAC for mobile sensor networks. Table 1 gives a brief qualitative overview of MAC protocols.

MAC protocol	**Channel access mode**	**Sensor network specifics**	**Power conservation**
SMACS and EAR	Fixed allocation of duplex time slots at fixed frequency	Exploitation of large available bandwidth compared to sensor data rate	Random wake up during setup and turning radio off while idle

Hybrid TDMA/FDMA	Centralized frequency and time division	Optimum number of channels calculated for minimum system energy	Hardware-based approach for system energy minimization
CSMA-based	Contention-based random access	Application phase shift and pre-transmit delay	Constant listening time for energy efficiency

Table 1: A qualitative overview of MAC protocols for sensor networks [7]

Network Layer: Responsible for routing information through the sensor network and finding the most efficient path for the packet to travel on its way to the destination. Most protocols can be categorized under one of the following techniques: gossiping, flooding, SMECN (Small Minimum Energy Communication Network) [12], SPIN (Sensor Protocols for Information via Negotiation) [13], SAR (Sequential Assignment Routing) [10], LEACH (Low Energy Adaptive Clustering Hierarchy) [14] and Directed Diffusion [15]. Table 2 gives some details about each of these protocols.

Network layer scheme	**Description**
SMECN	Creates a subgraph of the sensor network that contains the minimum energy path
Flooding	Broadcasts data to all neighbor nodes without regard to if they have received it before (or not)
Gossiping	Sends data to one randomly selected neighbor

SPIN	Sends data to sensor nodes only if they are interested; has three types of messages (i.e., ADV, REQ, and DATA)
SAR	Creates multiple trees where the root of each tree is one hop neighbor from the sink; selects a tree for data to be routed back to the sink according to the energy resources and additive QoS metric
LEACH	Forms clusters to minimize energy dissipation
Directed diffusion	Sets up gradients for data to flow from source to sink during interest dissemination

Table 2: An overview of network layer schemes

Transport Layer: Very little literature was found on this layer; protocols of this layer are yet to be explored. In general, these protocols are needed when the sensor network needs to be accessed through the Internet. The proper approach, since nodes are limited in power, is to suggest a UDP-type of protocol.

Application Layer: Responsible for representing required information to the application used in the network and propagate requests from the application down to the lower layers.

2.3 WSN vs. MANETs

Probably the closest technology to WSN is Mobile Ad hoc Networks (MANETs); the two share many characteristics. To mention just the obvious, they both don’t have a fixed network topology, the nodes are connected wirelessly, and power is an expensive resource [4]. Still both networks vary in many respects [16]

1. The number of sensor nodes in a sensor network is extremely bigger than the one of ad hoc network.

2. Sensor nodes are densely deployed.
3. Sensor nodes are prone to failure.
4. The topology of a sensor network changes frequently.
5. Radio range of sensor nodes is much less than MANETs.
6. Ad hoc networks use a point-to-point communication paradigm, while sensor networks rely on broadcast.
7. Sensor nodes are limited in power, computational capacities and memory.
8. Because of the large number of nodes, sensor nodes may not have global identification.
9. Sensor nodes should have a trust relationship with other nodes; this is not assumed with ad hoc networks.

An ad hoc network is set up to meet an immediate communication need instantly. While WSN have to interact with the environment, their traffic characteristics are different than other human-driven forms of networks [17]. Furthermore, MANETs are associated with different applications and different equipment; a node in ad hoc networks could be a laptop or PDA with plenty of battery and processing powers. Another major difference is the human interaction in MANETs; there is a constant monitoring of the network. Both are required to self-organize once deployed, but the difference is in the traffic load and routing protocols that save energy [17]. Power conservation is an important issue to extend the life of WSN and neither MANET nor Bluetooth protocols can be used.

MANETs support routing between any pair of nodes [18, 19, 20, 21]. Most traffic in WSN can be classified into one of three categories [9]:

1. *Many-to-one:* Multiple sensor nodes send their reading to the sink or an aggregation point.

2. *One-to-many:* The sink multicasts a query or control information to many nodes.
3. *Local communication:* Nodes send each other messages to localize themselves and discover their neighbors to coordinate with each other.

Since nodes sense the same phenomenon, they do not have to report the same reading to the sink multiple times. Instead, they process and aggregate the data among each other, discard any duplication, and then send it to the sink; this will reduce traffic and save energy.

Some security issues related to ad hoc networks are similar to those found in sensor networks, but the defense mechanisms developed for ad hoc networks are not applicable to sensor networks [22]. For instance, ad hoc networks use encryption to ensure authentication based on public key cryptography [23, 24, 25, 26], which is too expensive for sensor nodes to process.

2.4 WSN Security Barriers

Limitations set by WSN lead to a very demanding environment to provide security [8]. Security techniques used in traditional networks cannot be applied directly. Here, it is more than just message encryption. In fact, in many applications, encryption is not an important security goal of wireless sensor networks. The most important security goal is to ensure that any message received has not been modified in any way and is from the sender which it claims to be. There are other applications where security is of an ample importance, such as in disaster relief, public safety, home healthcare and military [27]. The network should be resilient to individual node failure, which could be a result of battery exhaustion, node physical destruction, or potential imperfection in large-scale production. Network continuity and functionality, even with disruption, is a critical challenge facing WSN.

1. Unreliable Transfer: This is a major threat to WSN. The entire network relies on defined protocols and communication for security. Packets between nodes are transmitted in a connectionless manner, which can be unreliable. These packets may be dropped at highly congested nodes. Some of the code is used for handling errors; if the protocol lacks this feature, a security packet might be lost in transition (e.g., a cryptographic key) [28].

2. Conflicts: With the broadcast nature of WSN, packets may collide in the middle of a transfer even with existing reliable channels. If packets meet in the middle, conflict will occur thus causing a failure of the transfer. This could be a significant problem in highly populated networks [7, 28].

3. Latency: Three major factors stand behind latency in the network and lack of synchronization between nodes: multi-hop routing, network congestion, and node processing. This is very important in networks that rely on cryptographic keys for synchronization [29, 7, 28].

4. Physical attack exposure: In many applications, sensor nodes are left unattended for long periods of time. The sensed field could be vulnerable to adversaries and inclement weather, thus leaving nodes exposed for physical attacks [28, 30, 7, 17].

5. Remote management: It is nearly impossible to manage the network physically; all management should be done remotely. In particular, when nodes are deployed in a hostile environment, this makes it impossible to have physical contact with the network.

6. Lack of central management point: If the network was not well thought out in the design phase, it could be fragile, inefficient and unorganized. Once deployed, it's very difficult to correct these errors. [30, 28]

7. Key establishment: Cryptographic keys are essential in the setup phase of a WSN for future use. This is a classical problem that researchers have been studying for decades now; several protocols have been proposed to address this problem. As mentioned earlier, WSN pose new challenges that render these protocols impractical.

2.5 WSN Security Requirements

Securing communication in a very large network constructed of thousands of unattended constrained nodes makes security a challenging task with the unique requirements. Various factors are worth mentioning at this point:

- Nodes depend on each other for correct operation.
- Messages have to be transmitted over several hops, since direct communication between arbitrary nodes is impossible due to limited radio range.
- Nodes have little knowledge of other distant nodes [26].

It is unfortunate that security is still looked at a separate component or a module that could be patched after the design is completed. To achieve a secure system, security mechanisms should be integrated into every component [30]; no component should be designed without security.

The major requirements for security in WSN could be summarized in the following domains:

Authentication
Data Confidentiality
Data Freshness
Robustness
Availability
Data Integrity
Secrecy
Privacy

Each item in the above list is a complete domain by itself, and is a research topic evolving as we speak. This research interest is focused on data integrity.

2.6 Cluster-Based Routing Protocols

This section will give a brief introduction about cluster-based routing protocol, which is the most popular research area in routing protocols for WSN.

Global goals of the network could be achieved via routing protocols; these protocols coordinate the activities of individual nodes in an efficient manner. In a cluster-based routing protocol, nodes are grouped in an efficient way to relay messages to the sink. Each group has a cluster head that acts like a gateway. In some protocols, these cluster heads suffer less from energy constraints, while in other protocols a cluster head could be any regular node in the group. The main objective of a cluster head is to aggregate data received from the group, check its validity, and then send it to the sink as a group representative.

The main advantage of cluster-based protocols is energy conservancy and minimizing latency. Current research is being conducted on factors affecting cluster formation, cluster head communication and data aggregation. Several protocols have been proposed [31], but the most promising ones are LEACH and PEGASIS.

Challenges facing a cluster-based routing protocol can vary depending on the application it addresses. Mobility and self-configuration/reconfiguration is a huge challenge.

2.7 Attacks and Countermeasures

Many sensor network routing protocols have been proposed, but none of them have been designed with security as a goal. [9] These protocols are simple in nature; this is why they are susceptible to attacks. It is important to mention that attackers can have much more energy at their disposal than the sensor nodes and radio signal could be much higher than the one operating on the node. All security measures carried out by a sensor node require energy, so stressing the network with attacks at a constant level can cause premature power depletion.

2.7.1 Attacks on Protocol Stack

Many applications served by WSN can be security-sensitive, and attacks against these networks might cause real-world damage to the health and safety of people.

2.7.1.1 Physical Layer: Jamming and tampering are the most common attacks on this layer.

1. Jamming interference with radio frequencies the nodes are using. Jamming a certain percentage of nodes could disrupt the network's operation. If the adversary has powerful machine, s/he could jam the entire network.

2. Physical attack and node capture. It is economically infeasible to make nodes tamper-proof because of the cost increase. An adversary could capture a node and replace parts of its hardware or try to gain access to data and cryptographic keys.

2.7.1.2 Data Link Layer: Attacks relevant to this layer could take different forms: collision, unfairness and battery exhaustion.

1. Collisions: It is similar to link layer jamming. If the adversary was able to change or corrupt an octet of transmission creating a mismatch in checksum, then the entire packet is disrupted.

2. Unfairness: This attack could be launched by abusing MAC priority schemes leading to missing real-time deadlines, resulting in service degradation.

3. Battery exhaustion: Could be as a result of naïve link layer implementation's attempt to repeatedly retransmit a packet after late collision.

2.7.1.3 Network Layer: Attacks against different routing protocols could fall in [9] one of the following categories: 1. spoofed, altered, or replayed routing information, 2. selective forwarding, 3. sinkhole attacks, 4. sybil attacks, 5. wormholes, 6. HELLO flood, and 7. acknowledgement spoofing.

1. Spoofed, altered, or replayed routing information: This is a direct attack where the attacker can complicate the network by creating routing loops by spoofing, altering or even replaying routing information. This attack could be carried out by targeting routing information

exchanged between nodes partitioning the network, creating false-error messages, and by increasing latency from end-to-end.

2. Sinkhole: If this attack is launched successfully, traffic will be lured from a particular area through the adversary’s node. One objective of this attack is to make the compromised node look attractive to the surrounding nodes with respect to their routing algorithms. Another objective is that it lays a path for launching selective forwarding attacks.

3. Selective forwarding: In a multi-hop network, the assumption is that participating nodes will faithfully forward messages received. In this kind of attack, the adversary includes himself/herself in the path of data flow to be effective. Malicious nodes may refuse to forward certain packets or simply drop them, acting like a black hole.

4. Sybil attack: The objective of this attack is to have the malicious node advertise multiple identities confusing the nodes around it. Sybil attacks targets fault-tolerant schemes such as distributed storage [32], disparity [33] and multi-path routing [35]; furthermore, it poses a big threat to geographic routing.

5. Wormhole: The adversary tunnels messages received in one part of the network over a low latency link to another part of the network where the messages are then replayed. An adversary could convince far nodes in the network, which are typically multiple hops away from sink, that they are one or two hops close to the sink through the wormhole. These attacks are commonly set up to appear through two colluding malicious nodes. A smart way to launch these attacks and conduct them without being noticed is to couple them with sybil and selective forwarding attacks.

6. HELLO flood attack: In many protocols, broadcasting a HELLO packet to the neighboring nodes announcing its location is very common. An attacker with a powerful

machine and antenna could convince the entire network that s/he is their neighbor. Nodes placed at a large distance from the attacker will be sending their messages into oblivion, leaving the network in a state of confusion.

7. Acknowledgement spoofing: Using the fact that some routing algorithms choose the next hop based on reliability issues, an adversary could advertise a weak or even a dead link as a strong reliable one.

Table 3 below lists various protocols with the relevant attacks associated with them.

Protocol	**Relevant attacks**
TinyOS beaconing	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes, HELLO floods
Directed diffusion and its multipath variant	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes, HELLO floods
Geographic routing (GPSR, GEAR)	Bogus routing information, selective forwarding, Sybil
Clustering based protocols (LEACH, TEEN, PEGASIS)	Selective forwarding, HELLO floods
Rumor routing	Bogus routing information, selective forwarding, sinkholes, Sybil, wormholes
Energy conserving topology maintenance (SPAN, GAF, CEC, AFECA)	Bogus routing information, Sybil, HELLO floods

Table 3: Summary of attacks against sensor networks protocols

Another building block in sensor networks is time synchronization between nodes. Traditional time synchronization protocols cannot be used in sensor networks due to the same constrains mentioned above. Several protocols are proposed to tackle this unique problem; unfortunately, none of them have been designed with security in mind. In addition to the attacks listed in the previous table, an additional threat is posed against time-synchronization protocols. All attacks of this nature have the same goal in mind, which is to convince some nodes that their neighbors' clocks are at a different time than they actually are. Since global time synchronization is built upon synchronization at the neighborhood level, this will disrupt the mechanisms by which the protocols maintain global time in the network or allow events at distant points in the network to be given time stamps that reflect the actual difference between their times of occurrence [34]. The most widely used protocols include Reference Broadcast Synchronization (RBS), Time-sync Protocol for Sensor Networks (TPSN), and Flooding Time Synchronization Protocol (FTSP). Designing a secure time synchronization protocols is a vital task for a proper functionality of the entire network, especially time-dependant applications.

2.8.2 Countermeasures

Many countermeasures have been proposed to defend against these attacks, yet none of these countermeasures have proved to be a good solution since they haven't been implemented on either the software or the hardware level. A detailed discussion about countermeasures could be found in [7, 9, 36, 37, and 38].

2.8.2.1 Jamming: There are a few techniques to defend against this attack by using spread spectrum communication. Another proposed solution is the use of code spreading similar to the ones used in mobile phones, but this kind of solution requires more design, power and cost.

2.8.2.2 Tampering: Motes may be programmed to delete sensitive information upon tampering since it is not economically feasible to tamper-proof packaging. Another approach proposed in [39] is to increase the effort of the adversary to run a successful attack. For example, data may be stored at a subset of nodes in the network and continuously be moved around by the sensors to evade possible access by the adversary. This makes it harder for the adversary to choose which node to capture. In the worst case, the adversary must capture much more than t out of n nodes to access the data in question.

2.8.2.3 Collision: Adding collision detection to the protocol could solve this problem, but it hasn’t been proved fully effective.

2.8.2.4 Spoofed, altered or replayed data: By introducing link layer encryption and authentication, we can prevent outside attacks. The problem with the proposed solution is the use of a global key; once this key has been compromised, the entire network is at stake. And more so, this technique proves useless if the attack was launched from inside.

2.8.2.5 Selective forwarding: Introducing multi-path routing to the network could prevent this attack from occurring; this will give the message an opportunity to reach the destination using several paths with the hope that one of them will reach it, given that not all nodes are compromised. This is an expensive approach and would overwhelm nodes with the same message.

2.8.2.6 Sybil attack: An insider node cannot be prevented from participating in the network, but then the attacker is restricted to use identities from the same network. Therefore, identities must be verified and using public key cryptography, and checking digital signatures is beyond the scope of sensor networks. A good approach is to have every node share a unique key with the base station, and then nodes could verify one another using a Needham-Schroeder-like protocol and establish a shared key.

2.8.2.7 HELLO flood attack: These kinds of attacks can be easily avoided by verifying bi-directionality of a link between two nodes before taking meaningful action based on a message received over that link.

2.8.2.8 Wormhole and sinkhole attack: These are one of the most difficult attacks to prevent, especially when the two are coupled together. Wormholes use invisible channels to the network and the advertised routes of sinkhole attacks are extremely hard to verify. Geographic routing is immune from these attacks because messages are routed to the physical location of the sink; false links are easily detected by neighbors once they figure out that the physical distance of the advertised route exceeds the signal range of the motes. Another proposed solution is tight time synchronization but this is extremely expensive and requires a new protocol to handle it.

Table 4 below summarizes attacks and proposed countermeasures.

Attacks	Countermeasures
Bogus routing information, Sybil attacks, HELLO floods, and acknowledgement spoofing,	Link-layer encryption and authentication, Multi-path routing, identity verification, bidirectional link verification, and authenticated broadcast
Sinkhole attacks and wormholes	They pose significant challenges to secure routing protocol design, and it is unlikely there exists effective countermeasures against these attacks that can be applied after the design of a protocol has completed. Geographic routing protocols are one class of protocols

	that holds promise.

Table 4: Attacks and suggested countermeasures

2.9 Cryptographic Primitives

Wireless sensor networks operating over insecure wireless channels and nodes are deployed in the public, which makes them an easy target by an adversary. The standard approach for keeping sensitive data from leaking out is to employ encryption and encrypt the data with a key known only by the receiver [8]. Since sensor nodes are constrained with computational power, memory and energy, using asymmetric cryptography like RSA signature algorithm or the Diffie-Hellman key agreement protocol is too expensive. On the other hand, a better approach is to use symmetric cryptographic alternatives like AES block cipher or the HMAC-SHA-1 message authentication code. A major drawback of symmetric cryptography is that it is not as versatile as public key techniques which complicate the design of secure applications [40]. Another important issue is scalability; security techniques should be capable of scaling to large-scale deployments. Problems occur at the early setup phase of the network, where shared secrets need to be distributed either by the manufacturer at production time or by clever protocols at deployment time [41, 42, 39].

2.9.1 SPINS: Security Protocols for Sensor Networks

SPINS is a suite of security building blocks proposed by A. Perrig et al.; highly-constrained sensor nodes [43]. The first design was for the first generation of sensor nodes Rene, which had very scarce resources. Later it was applied on Mic2 motes. SPINS has two blocks to it: SNEP and μTESLA.

SNEP: Sensor network encryption protocol; it provides data confidentiality, two-party data authentication, and data freshness.

μ**TESLA:** Micro version of the Timed, Efficient, Streaming, Loss-tolerant Authentication; is an authenticated broadcast protocol; it relies on asymmetric mechanism to allow nodes receiving a broadcast message to verify the authenticity of the message [34, 36, 43]. The mechanism is based on the arrival time of the messages.

A major drawback in this protocol is that it assumes that nodes have a global identifier; this is not applicable in sensor nodes lacking identifiers.

2.9.2 TinySec

Link layer security architecture based on TinyOS. TinySec claims to protect authenticity, integrity and confidentiality of messages between neighbor nodes. It provides two security operations: authenticated encryption (TinySec-AE) and authentication-only (TinySec-Auth). It uses Skipjack and RC5 cryptographic algorithms. Although this protocol shows much strength, it fails to address the attacks of jamming, key compromise, replay and denial of service.

CHAPTER 3. MESSAGE INTEGRITY

Message integrity addresses the threat of unauthorized manipulation of data. Given the challenges we mentioned earlier, there is couple of frequent questions asked that needs to be addressed: “When the data is kept confidential, does it mean it is safe from tampering?” Another common question is “How can the end user rely on the information provided by the sensor network?” With the implementation of confidentiality, it will be difficult for the adversary to steal it; however, this doesn’t mean its safety. An attacker can change the data and send the sensor network into disarray [28]. For instance, a malicious node may add some fragments or manipulate the data within the packet; this new packet can be sent to the original receiver. Data loss or damage can occur from harsh communication environments. Hence, data integrity ensures any received data has not been altered in transit. Cryptographic and authentication mechanisms alone cannot be used to solve this problem as internal adversarial nodes will have access to valid keys [2]. Besides, sensor nodes are also vulnerable to system faults and non-malicious malfunctioning of transceiver due to harsh communication generating faulty data. Such behavior is outside the realm of cryptography [2].

Current proposals for routing protocols in sensor networks optimize for the limited capabilities of the nodes and the application-specific nature of the networks, but do not consider security. These protocols were designed without security as a goal [9]. It is unlikely a sensor network routing protocol can be made secure by incorporating security mechanisms after design have been completed.

3.2 Clark-Wilson Integrity Model

What is so unique about the Clark-Wilson Integrity Model [44, 45] is that it uses transactions as the basic operation, which models very well with WSNs. Each node has to relay sensed data from the environment to its neighboring node or to the cluster head if it is near.

The distinctive property of *well-formed transaction* transitions the system from one consistent state to another consistent state through a well defined series of operations. In this model, a secure system should have the following characteristics:

- prevent unauthorized disclosure or theft of information
- prevent unauthorized modification of information
- prevent denial of service attacks

Traditional threats the system should have countermeasures for:

- system penetration by unauthorized user
- unauthorized actions by authorized user
- abuse of special privileges by systems programmers and facility operators

In most of WSN applications, preventing unauthorized data modification is of a greater importance than preventing disclosure. That is the core idea of this model.

It is important to mention what is needed from this model; mainly, there are two main points to defend:

1. There is a distinct set of security policies related to integrity rather than disclosure, which are often of highest priority in most WSN applications.
2. Some separate mechanisms are required for enforcement of these policies.

Two mechanisms at the heart of fraud and error control:

1. The well formed transaction
2. Separation of duty

Since this model was developed for commercial use, the simple example that accompanies it is drawn from the same environment. Let D be today's deposits, W is the amount of money withdrawn today so far, YB is the amount of money in all accounts at the end of yesterday, and TB is the amount in all accounts so far today. With a well-formed transaction system, the bank accounts should be balanced satisfying the integrity constraint:

$$D + YB - W = TB$$

3.2.1 The Formal Model

The first step is to label data items in the system to which the integrity model should be applied; these data items are called constrained data items (CDIs). The desired integrity policy is defined by two classes of procedures [44, 45]: integrity verification procedures (IVPs) and transformation procedures (TPs). IVPs confirm that all the CDIs in the system conform to the integrity specification at the time the IVP was executed. TPs correspond to well-formed transactions; it takes the set of CDIs from one valid state to another other. A valid state maintains its integrity when the system ensures that only TPs can manipulate and handle the CDIs. It is a valid assumption that the system is in a current valid state because an IVP was executed to verify it. If we return to the bank example:

CDIs = Balances in the accounts

IVP = Checking if the accounts are balanced

TPs = Depositing, withdrawing and transferring money

To ensure that the bank is managing the accounts correctly, a bank examiner must certify that the bank is using proper procedures to check that the account is balanced.

The integrity assurance is a two-part process:

1. Certification: Done by the security officer, system owner with respect to an integrity policy.
2. Enforcement: Done by the system.

3.2.2 Clark-Wilson Rules

The model [44, 45] could be summarized in certification rules (CRs) and enforcement rules (ERs)

CR1: When an IVP is run, it must ensure that all CDIs are in a valid state.

CR2: For some associated set of CDIs, a TP must transform those CDIs in a valid state into a (possibly different) valid state.

Note that a TP may corrupt a CDI if it is not certified to work on that CDI.

Access control rules:

ER1: The system must maintain the certified relations and must ensure that the only TPs certified to run a CDI manipulate that CDI. This means that if *f* operates on *o* the (*f*, *o*) ? *C* where C is the set of certified relations.

- This defines a set of triples (user, TP, {CDI set}) to capture the association of users, TPs and CDIs.
- These triples define a relationship called "*allowed* relation".

CR1, CR2 and ER1 provide the basic framework to ensure internal consistency of the CDIs. To provide a mechanism for external consistency, we need additional rules:

- ER2: The system must associate a user with each TP and set of CDIs. The TP may access those CDIs on behalf of the associated user. The TP cannot access that CDI on behalf of a user not associated with that TP and CDI. The relation could be in the form of (UserID, TPi, (CDIa, CDIb, CDIc, …)).

- CR3: List of relations in ER2 must be certified to meet the separation of duty requirement, i.e. allowed relations must be certified.
- ER3: The system must authenticate the identity of each user attempting to execute a TP.

All TP execution should be logged to provide audit trail. To implement this, another CDI is used with an associated TP that only appends to the existing CDI value.

- CR4: All TPs must be certified to write an append-only CDI logging enough information necessary to permit the nature of the operation to be reconstructed.

There is one more component to the integrity mode in that not all data is constrained data. In addition to CDIs, most systems contain data items not covered by the integrity policy. These data items are called unconstrained data items UDIs. Example of UDIs could fall under data entered from a keyboard.

-CR5: Any TP that takes a UDI as an input value must be certified to perform only valid transformations, or else no transformations for any possible value of the UDI. The transformation should take the input from a UDI to a CDI or the UDI is rejected.

For the model to be effective, the various certification rules must not be bypassed leading to separation of duty section.

- ER4: Only the certifier of a TP may change the list of entities associated with a TP. No certifier of a TP, or of any entity associated with that TP, may ever have execute permission with respect to that entity.

3.3 Clark-Wilson vs. Biba

Another model, as important is as Clark-Wilson is Biba integrity model [BIB77]. Biba constructed a model for preventing inappropriate modification of data. It is dual to Bell-La

Padula model. The reason why Biba wasn’t considered in this research as the foundation like Clark-Wilson is the fact that Biba defines integrity levels, which are analogous to the sensitivity levels of the Bell-la Padula model. This property makes it an inadequate approach since in a cluster-head network, each node can become a cluster-head at any time destroying the integrity levels constructed by Biba. Other reasons worth mentioning, it is hard to determine the integrity labels. While it is hard to implement in real systems, Biba provides no mechanism to support data consistency.

Instead of data and user level classification, Clark Wilson model places strict controls on what programs have permission to manipulate certain data, and what users have access to these various programs. This feature makes it a reasonable approach for WSN.

CHAPTER 4. WSN MESSAGE INTEGRITY MODEL

Based on our research and literature provided, there is no well founded integrity model developed for WSN. Although Clark-Wilson encompasses all objectives regarding integrity: user integrity, data integrity and process integrity; we cannot apply it the as it is. One of the main limitations that Clark-Wilson has is that it is hard to formalize it; which works to our advantage since we will modify it to serve our research goals. Its strengths lie in its well-formed transaction preserving data integrity and separation of duty. We are concerned in how we can apply these strengths while designing a system with high integrity.

Our model is based on these requirements:

1. Trusted subjects: initial nodes.
2. Trusted Code: all application codes and static data for any trusted subject must correspond to known and trusted hashes.
3. Information Flow: all information flowing to a trusted subject must come from another trusted subject.
4. Initial Verification: initial verification procedure code must be of high integrity.
5. Message payload: the message payload must be the same en route from its origin to destination.

In this model, we assume WSNs are homogenous (they contain the same hardware and software configuration) and static (nodes do not move after deployment). Moreover, we assume that data confidentiality is not important for that specific application. The goal is verifying the exchanged content between communicating participants of the WSN have been altered. Figure 3 shows system architecture for habitat monitoring.

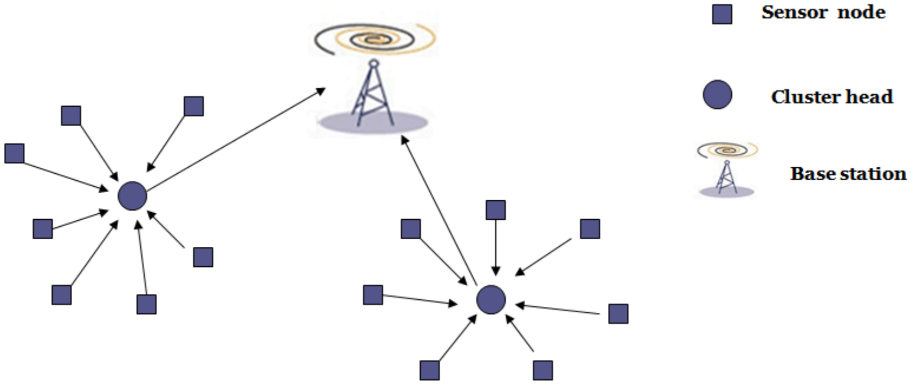

Figure 3. Wireless sensor network monitoring habitat

Once the nodes are in place, a cluster based routing protocol groups the sensor nodes to relay sensed data to the base station efficiently. Each cluster of nodes has a cluster head. In [47], a proposed election mechanism is introduced of cluster heads. This method will reduce the likelihood of a compromised and malicious node from being selected as a cluster head. This approach though does not scale to all nodes, but is a mechanism that addresses a potential breach from the early distribution of the nodes.

Clustering facilitates data aggregation and is an energy efficient technique where nodes forward data to cluster heads for processing and then transmit the findings to the base station. The designer could use broadcast, multicast or anycast in communications; this method is very effective. Figure 3 shows how a hierarchical approach breaks the network to many segments and layers. Aggregated data travel from a lower clustered layer to a higher one until it reaches the sink. A hierarchical based cluster based moves data faster to the base station; this reduces latency and is more power efficient.

Environmental and habitat monitoring is a great example for an application where confidentiality is not of a great importance but data integrity is paramont. We have mentioned earlier many examples of wireless sensor networks, but we believe that environmental monitoring is a domain where they have a huge impact. Sensor nodes could sense events like pollution or ice melting, and at the same time reporting pressure, temperature and wind speed.

Since WSN consists of multiple identical nodes collaborating in a cluster set to achieve one goal, which is sensing the surrounding area. For each sensed phenomena, there will be a service initiated at node level to communicate with the cluster head. Note, other nodes in the same cluster layer and proximity will sense the same phenomena and report. Depending on the network design, nodes could sense once every second or minute; this is something set by the user during application development cycle. The initiation of sensing is referred to as *initiation service.* As we have explained in chapter 3, all data items used in this model can either be constrained data items (CDIs) or unconstrained data items (UDIs). If the nodes are sensing multiple items, each data group should be handled with a different initiation service.

After the nodes are in place and cluster layers are formed, the first sensed reading occurs. At this time, the data being sensed is dealt with as UDI by all nodes. Once the reading is captured, this data becomes a CDI. To maintain the integrity of the CDIs, we have the integrity verification procedures (IVPs) and transformation procedures (TP). It is worth remembering here that IVPs ensures that all CDIs meet the integrity constraints of the system before the second round of sensation. This guarantees that the system is in a valid state.

A major advantage for this state machine kind of process is that after the initiation services has been executed, and TP ensured that all CDIs are consistent, the data is preserved from any alteration to that state even if a malicious node was introduced.

Each node could be a sensing node or a cluster head, depending on the characteristics described earlier. This means that each node is equipped with a node transformation procedures and cluster-head transformation procedures. The node transformation procedures are executed between nodes while data gathering and the cluster-head transformation procedures are executed with cluster heads communicate aggregated data. This means that for the system to stay in a valid state, all procedures should complete successfully and all services moved to another valid state.

Certification Rule 1: when IVP is run, it turns all UDIs to CDIs ensuring that all CDIs in a valid state.

Certification Rule 2: all CDIs are associated with an initiation service. Each initiation service is responsible for its own CDIs enforcing separation of duty.

Since we are dealing with multiple CDIs and multiple services, there might be inconsistency between nodes. Cluster head must be to ensure that data gathered by different services from different nodes are consistent and should rule out any inconsistencies.

Certification Rule 3: cluster-head must be able to ensure data consistency from nodes

Enforcement Rule 1: the system must maintain the certified relations and must ensure that the only TPs certified to run a CDI manipulate that CDI.

The formal model of Clark Wilson assigns users to TPs. In our framework, nodes and cluster-heads could act as users; but each TP can manipulate only a certain set of CDIs.

Enforcement Rule 2: the system must associate a nodes with each TP and CDIs.

In this model, taking Clark Wilson as the base; we took advantage of the major two principles in it: separation of duties and well formed transaction to ensure integrity. While we focus on the latter, the first principle is important but it is beyond the scope of this research. This model defines a higher abstract notion of transaction.

CHAPTER 5. CONCLUSION

The main contribution of this work is to establish an integrity model for WSN and help design a network that sustain high information integrity for a long duration [50]. In this research we have presented a mechanism for message integrity in WSN that is based on Clark Wilson integrity model. The impact this model is very broad, it will benefit the scientific that are using sensor networks as well as the environmental ones where integrity is more of a concern than confidentiality. Moreover, it gives the designer the freedom to use the proper way of implementation. Our approach is a top-down one and is capable of describe sufficient conditions to protect and preserve the integrity of sensed data.

In this research, we presented a new way to preserve message integrity. Our approach is based on Clark Wilson integrity model in which each cluster head verifies if the previous cluster head has preserved the integrity of the message using a set of rules.

Integrity in WSN is still a new research topic and there is a lot of room for improvement. Our model needs to be tested for design, robustness and scalability. A major research topic is physical security and non-malicious behavior (like channel noise, snow, rain, dust etc). Improving in this field takes a lot of effort and collective work. A researcher team should have access to expertise in spanning statistics, networking, signal processing, hardware and software platforms, and information and security.

When confidentiality is not a major requirement in a network, our model ensures message integrity especially in environmental monitoring which would reveal previously unobservable phenomena in the physical world [46].

This model fails to address sensor mobility and heterogeneity of nodes; if the network is comprised of two or more types of sensor nodes then the model must be adapted on both application and node levels. Tests need to be implemented on the robustness and scalability of this model. Another major research topic that could help our model is building reputation between nodes and ensures trustworthiness among them after a certain period of deployment.

A routing protocol lacking security from early design stages leaves us with a vulnerable network that could be easily compromised.

BIBLIOGRAPHY

[1] MIT Technology Review, “10 Emerging Technologies That Will Change the World,” http://www.technologyreview.com/Infotech/13060/

[2] Saurabh Ganeriwal and Mani B. Srivastava, “Reputation-based Framework for High Integrity Sensor Networks,” 2nd ACM workshop on Security of ad hoc and sensor networks, 2004

[3] Nirupama Bulusu and Sanjay Jha, Introduction to Wireless Sensor Networks, Artech House, Aug. 2005

[4] Mark Weiser, “The computer for the 21st century,” Scientific American, Sep. 1991

[5] B. Warneke, M. Last, B. Liebowitz, K. Pister, “Smart Dust: communicating with a cubic-millimeter computer,” IEEE Computer, Jan 2001

[6] David Wagner, Naveen Sastry, “Security Considerations for IEEE 802.15.4 Networks,” WiSE 04, Oct. 2004

[7] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, “A Survey on Sensor Networks,”, IEEE Communications Magazine, Aug. 2002

[8] Mayank Saraogi ,”Security in Wireless Sensor Networks,” Department of Computer Science University of Tennessee, Knoxville

[9] Chris Karlof and David Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” IEEE Sensor Network Protocols and Applications, 2003

[10] E. e. a. Sohrabi, “Protocols for self-organization of a wireless sesnor network,” pp. 16–27, October 2000.

[11] A. Woo and D. Culler, “A transmission control scheme for media access in sensor networks.”

[12] L. Li and J. Halpern, “Minimum- energy module wireless networks revisited,” June 2001

[13] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” pp. 174–185, 1999.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in The 33rd Annual Hawaii International Conference on System Siences (HICSS-33)

[15] C. Intanagowiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” 2000.

[16] Chris Karlof , Naveen Sastry and David Wagner, “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks”, SenSys’04, November, 2004

[17] Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks,” WILEY 2005

[18] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in IEEE INFOCOM ’97, 1997

[19] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,”, MILCOM ’97 panel on Ad Hoc Networks, 1997

[20] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” Mobile Computing, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[21] C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” ACM/SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications, 1994

[16] = [22] Arjan Durresi, Vamsi Paruchuri, Rajgopal Kannan and S.S. Iyengar, “A Lightweight Protocol for Data Integrity in Sensor Networks,” ISSNIP 04

[23] L. Zhou and Z. Haas, “Securing Ad-Hoc Networks, IEEE Network Magazine, Vol. 13, No. 6, 1999.

[24] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, ”Providing robust and ubiquitous security support for mobile ad-hoc networks, in ICNP, 2001

[25] M. G. Zapata, ”Secure ad-hoc on-demand distance vector (SAODV) routing”, IETF MANET Mailing List, Message-ID: 3BC17B40.BBF52E09@nokia.com, Available at ftp://manet.itd.nrl.navy.mil/pub/manet/2001-10.mail, October 8, 2001.

[26] H. Luo, P. Zefros, J. Kong, S. Lu, and L. Zhang, ”Self-securing ad hoc wireless networks”, in Seventh IEEE Symposium on Computers and Communications (ISCC 02), 2002.

[27] Anthony D. and Wood John A. Stankovic, “Denial of Service in Sensor Networks,” IEEE Computer 2002

[28] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, “Security in Distributed, Grid, and Pervasive Computing,” CRC Press, 2006

[29] J. A. Stankovic et al. Real-time communication and coordination in embedded sensor networks. Proceedings of the IEEE, 91(7):1002–1022, July 2003

[30] Adrian Perrig, John Stankovic, and David Wagner, “Security in Wireless Sensor Networks,” ACM June 2004

[31] Jamil Ibriq and Imad Mahgoub, “Cluster-based Routing in Wireless Sensor Networks: Issues and Challenges,” SPECTS ‘04

[32] Castro and Liskov, “Practical byzantine fault tolerance,” in OSDI: Symposium on Operating Systems Design and Implementation. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, 1999.

[33] A. Banerjea, “A taxonomy of dispersity routing schemes for fault tolerant real-time channels,” in Proceedings of ECMAST, vol. 26, May 1996

[34] Michael Manzo, Tanya Roosta and Shankar Sastry, “Time Synchronization Attacks in Sensor Networks,” SASN’05, Nov. 2005

[35] K. Ishida, Y. Kakuda, and T. Kikuno, “A routing protocol for finding two node-disjoint paths in computer networks,” in Internation Conference on Network Protocols, November 1992

[36] Nirupama Bulusu and Sanjay Jha, Wireless Sensor Networks, ARTECH HOUSE 2005

[37] Holger Karl and Andreas Wilig, Protocols and Architecture for Wireless Sensor Networks, WILEY 205

[38] Sophia Kaplantiz, “Security Models for Wireless Sensor Networks”, University of Monach, 2005

[39] Zinaida Benenson and Felix C. Freiling, “On the Feasibility and Meaning of Security in Sensor Networks,” Department of Computer Science, RWTH Aachen University, Germany

[40] Elaine Shi and Adrian Perrig, “Designing Secure Sensor Networks,” IEEE Wireless Communications, Dec. 2004

[41] R. J. Anderson, H. Chan, and A. Perrig. Key infection: Smart trust for smart dust. In ICNP, 2004.

[42] J. Hwang and Y. Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In SASN ’04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. ACM Press, 2004

[43] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, “SPINS: Security Protocols for Sensor Networks,” Mobile computing and networking, 2001

[44] David D. Clark and David D. Wilson, “A comparison of Commercial and Military Computer Security Policies,” IEEE 1987

[45] Matt Bishop, Computer Security Art and Science, Addison-Wesley 2002

[46] Jeremy Elson and Deborah Estrin, “Sensor Networks: A bridge to the physical world”, Wireless Sensor Networks, Chapter 1

[47] Garth V. Crosby, Niki Pissinou, James Gadze, “A Framework for Trust-based Cluster Head Election in Wireless Sensor Networks”, Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems

[48] Wireless Sensor Networks: An Information Processing Approach by Feng Zhao and Leonidas Guibas

[49] - Mark Weiser, head of the Xerox PARC Computer Science Laboratory, in Beekman, Computer Confluence: Exploring Tomorrow's Technology, P. 260, Chapter 9, Copyright 2001, 572 pp. Paper format ISBN 0-13-088237-2

[50] Mark Hansen, Greg Pottie, Songwu Lu, Mani Srivastava, “NeTS-NOSS: Algorithms & System Support for Data Integrity in Wireless Sensor Networks”, last accessed April 2009 at http://nesl.ee.ucla.edu/fw/integrity/noss-cal-all-20050121.pdf