DEVELOPMENT OF SECURITY IN WIRELESS SENSOR NETWORK USING IPv6

VIKNESWARY JAYAPAL

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical-Electronics & Telecommunications)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MAY 2009

ABSTRACT

The Wireless Sensor Networks (WSNs) is one of core technologies expected to become a potential basis of future ever-present networks. WSN consists of multiple low cost sensor nodes, which could either, have a fixed location or randomly deployed that can communicate with each other for monitoring environments, medical systems, home network, industry automation and so on. However, most of the application scenarios require connectivity between WSNs and the Internet. Though WSN is typically not IP-enabled, connection to the IP network makes it easy to monitor sensors everywhere in the world. One of the purposes of the research work is to incorporate the IPv6 with WSNs, where IPv6 offers a larger address space. Therefore each of the sensor nodes will have their own IP address compare to IPv4, which has limited address space.The main objective of this research is to implement security in WSNs. Sensor networks are typically characterized by limited power supplies, low bandwidth, small memory sizes and limited energy. In addition, unlike traditional networks, sensor nodes are often deployed in accessible areas, presenting the added risk of physical attack. This leads to a very demanding environment in providing security. The research proposed XOR encryption algorithm that possesses built-in and enhanced security measure. The encryption and decryption of the payload secure the data's of the packet transfer. Other than that the XOR encryption is meant to realize the real time routing where the packets will be delivered within their end-to-end deadlines.XOR encryption security has been implemented in the 61owpan/IPv6 stack for TinyOS 2.1. TinyOS 2.1 an embedded operating system commonly used in wireless sensor networks. The hardware platforms used in this project, the TelosB motes, the 802.15.4 wireless communication standard and the TinyOS 2.1 operating system

ABSTRAK

Rangkaian peranti pengesan tanpa wayar(WSNs) adalah satu daripada teknologi teras yang dijangka akan berkembang menjadi rangkaian yang berpotensi. WSN terdiri daripada nod-nod peranti pengesan yang mempunyai tenaga dan ingatan terhad, di mana lokasinya diatur secara tetap atau rawak. Nod peranti pengesan boleh berkomunikasi antara satu sama lain untuk memantau persekitaran, sistem-sistem perubatan, rangkaian rumah, automasi industri dan sebagainya. Bagaimanapun, kebanyakan senario memerlukan keberkaitan antara WSNs dan Internet Walaupun WSN lazimnya bukan IP berkeupayaan, sambungan kepada rangkaian IP memudahkan ia untuk memantau pengesan-pengesan di serata dunia. Salah satu daripada tujuan kerja penyelidikan ini adalah, untuk menggabungkan IPv6 dengan WSNs, di mana IPv6 menawarkan ruang alamat yang besar. Oleh itu setiap nod peranti pengesan akan memiliki alamat IP yang tersendiri, berbanding dengan IPv4, yang telah menghadkan ruang alamat. Objektif utama penyelidikan ini merupakan untuk melaksanakan securiti di WSNs. Lazimnya pengesan jaringan berunsur bekalan kuasa yang terhad, rendah lebar jalur, saiz ingatan kecil dan tenaga terhad. Seperkara lagi, berbeza daripada rangkaian tradisional, nod peranti pengesan adalah sering digunakan di kawasan yang boleh, memberi risiko tambahan serangan fizikal. Penyelidikan ini mencadangkan algoritma enkripsi XOR yang memiliki securiti terbina dalaman dan dipertingkatkan. Enkripsi dan dekripsi *"payloacF* dapat menghantar paket data yang selamat. Selain daripada itu, enkripsi XOR adalah untuk penghalaan masa nyata di mana paket akan dihantar dalam tempoh tamat hujung ke hujung *"end to end deadlines"*. Enkripsi XOR securiti telah dilaksanakan dalam *"6lowpan"* / IPv6 susunan untuk *"TinyOS* 2.7". TinyOS 2.1 ialah satu sistem pengendalian tersirat biasanya digunakan dalam rangkaian peranti pengesan tanpa wayar. Perkakasan dan platform yang digunakan dalam projek ini adalah TelosB *"motes*", 802.15.4 standard komunikasi tanpa wayar dan TinyOS 2.1 sistem pengendalian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CPU	-	Central Processing Unit
DAD	-	Duplicate Address Detection
FFD	-	Full Function Device
GCC	-	GNU C compiler
HMAC	-	Hash Message Authentication Code
ICMPv6	-	Internet Control Message Protocol Version 6
IP	-	Internet Protocol
IPng	-	Internet Protocol Next Generation
IPv4	-	Internet Protocol Version 4
IPv6	-	Internet Protocol Version 6
LAN	-	Local Area Network
LFSR	-	Linear Feedback Shift Register
LoWPAN	-	Low Personal Area Network
LR-WPAN	-	Low Rate Wireless Personal Area Network
MAC	-	Media Access Control
MAC	-	Message Authentication Code
NAT	-	Network Address Translation
ND	-	Neighbour Discovery
NesC	-	Network Embedded System C
OS	-	Operating System
PRNG	-	Pseudorandom Number Generator
RAM	-	Random Access Memory
RFD	-	Reduced Function Device
ROM	-	Read Only Memory
SAA	-	Stateless Address Autoconfiguration
SHA1	-	Secure Hash Algorithm 1
TCP	-	Transmission Control Protocol

TEA	-	Tiny Encryption Algorithm
TinyOS	-	Tiny Operating System
UDP	-	User Datagram Protocol
USB	-	Universal Serial Bus
WSN	-	Wireless Sensor Network
XOR	-	Exclusive or

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Current technology in wireless communications and electronics have allowed the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances [1]. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks. Sensor networks represent a significant improvement over traditional sensors. Researchers see Wireless Sensor Networks (WSNs) as an "exciting emerging domain of deeply networked systems of low-power wireless motes with a tiny amount of CPU and memory, and large federated networks for high-resolution sensing of the environment" [2].

The WSNs is one of core technologies expected to become a potential basis of future ever-present networks. WSN consists of multiple low cost sensor nodes, which could either, have a fixed location or randomly deployed that can communicate with each other. Sensor nodes communication has a significant energy cost and, therefore, the development of energy-efficient hardware, software and protocols are very important from the point of view of the lifetime and performance of these networks. The size of a single sensor node can vary from shoebox-sized nodes down to size of a grain of dust and due to the size and cost constraints on sensor nodes lead to the corresponding limitations on resources such as energy, memory, computational speed and bandwidth [3].

1.1.1 Architecture WSNs

Figure 1.1 shows the communication architecture of a WSN. Sensor nodes are usually scattered in a sensor field and sensor nodes organize among themselves to produce high-quality information about the physical environment. Each sensor node makes its decisions based on its mission, the current information and its knowledge of its computing, communication, and energy resources.

Each of these scattered sensor nodes has the capability to collect and route data either to other sensors or back to an external base station(s). A base-station may be a fixed node or a mobile node capable of connecting the sensor network to an existing communications infrastructure or to the Internet where a user can have access to the reported data [4].

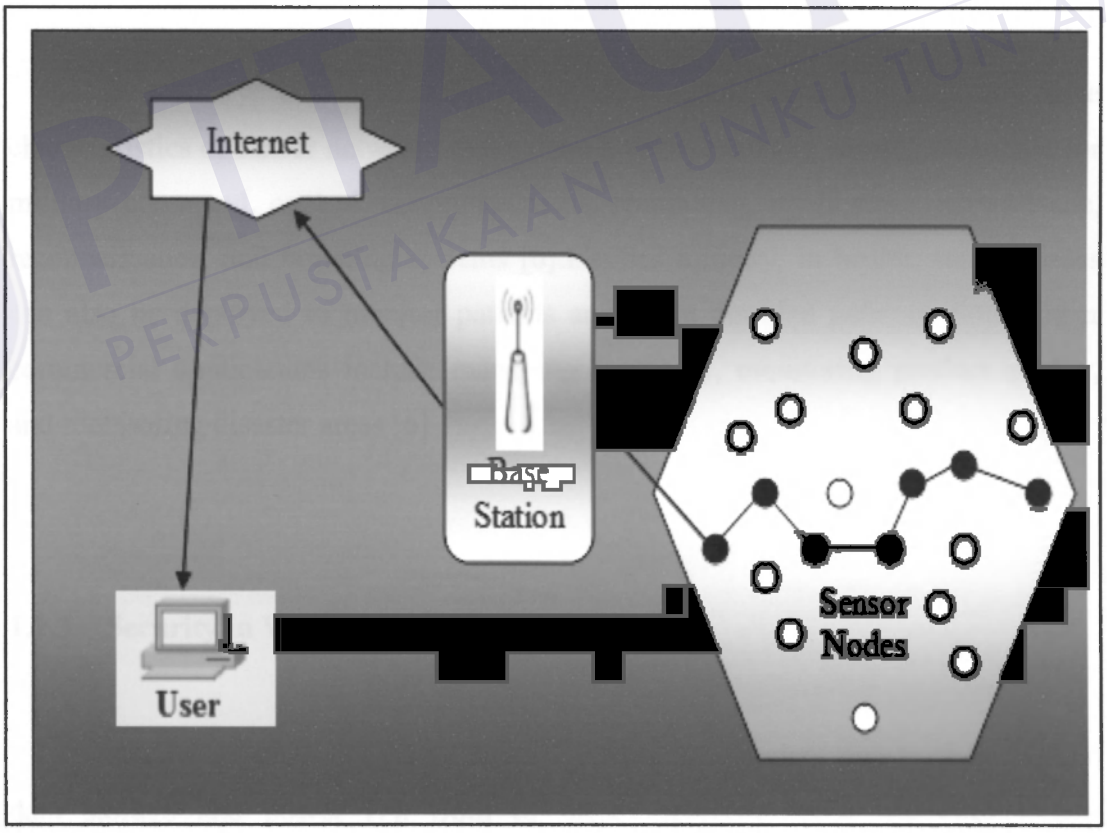

Figure 1.1 Communication Architecture in WSNs

1.1.2 Applications

Sensors in a WSN have a variety of purposes, functions, and capabilities. The field is now advancing under the push of recent technological advances and the pull of a myriad of potential applications. The radar networks used in air traffic control, the national electrical power grid, and nationwide weather stations deployed over a regular topographic mesh are all examples of early-deployment sensor network.

In recent technology, WSN may be able to monitor a wide variety of ambient conditions that include temperature, humidity, movement, lightning condition and pressure. It also can be used to monitor soil makeup, noise levels, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and the current characteristics such as speed, direction, and size of an object [5].

In military, the rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting systems [6].Besides military, in health, sensor nodes can also be deployed to monitor patients and assist disabled patients. Some other commercial applications include managing inventory, monitoring product quality, and monitoring disaster areas [6]

1.1.3 Security in WSNs

Sensor networks introduce severe resource constraints due to their lack of data storage and power [7]. Both of these represent major obstacles to the implementation of traditional computer security techniques in a wireless sensor network.

Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms while borrowing the ideas from the current security techniques, it is necessary to know security requirements in WSN [8, 9]. Following are a few requirements:

i. Data Confidentiality - In most applications nodes communicate very sensitive data such as surveillance information and industrial secrets. Such applications need to rely on confidentiality where the sensor network should not leak the readings to its neighbour. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, thus achieving confidentiality.

ii. Data Integrity - To ensure that information is not changed in transit, either due to malicious intent or by accident. 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

iii. Data Freshness - Even with data confidentiality and integrity does not mean the data is safe. Data also need to ensure the freshness of each message such that the data is recent, and to ensure that no old messages can be replayed. An adversary nodes might easily disrupt the normal work of the sensor, if the sensor unaware of recent or new data.

iv. Self-Organization - Distributed sensor networks must self-organize to support multihop routing. Such self organization is very hard to be done in a secure way. To do the multihop, 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.

v. Authentication - An adversary is not just limited to modifying the data packet. It can change the whole packet stream by injecting additional packets. So the receiver needs to ensure that the data used in any decision-making process originates from the correct source. On the other hand, when constructing the sensor network, authentication is necessary for many administrative tasks .Verifying that principals are who they claim to be can be achieved through appropriate proof of identity (i.e. encrypted signature).

1.2 Problem Statement

For WSNs to become truly ubiquitous, a number of challenges and hurdles must be overcome. One of the challenges on sensor nodes is to provide a security networks. Sensor networks are typically characterized by limited power supplies, low bandwidth, small memory sizes, limited energy and most of the WSNs will be deployed in a critical area.

Sensor networks may interact with sensitive data and/or operate in hostile unattended environments [7] therefore it leads to a very demanding environment to provide security. Furthermore, routing protocols do not address security, so it is easy for an adversary to exploit those routing protocols on a given WSNs. Unfortunately, security may be the most difficult problem to solve in WSNs [7, 10].

Since, most of the application scenarios require connectivity between WSNs and the Internet. Though WSN is typically not IP-enabled, connection to the IP network makes it easy to monitor sensors everywhere in the world but the use of the IP protocol has always been considered inadequate due to the fact that it does not minimize memory usage nor processing needs [11]. Sensor nodes have energy and

processing limitations, and the use of full TCP/IP protocol mechanisms requires resources that do not exist in these devices.

Other than that large numbers of sensor nodes that are densely deployed in a sensor network may not have a global identification due to the insufficient space address in IPv4. Extending IP to WSNs also considered impractical because these networks are highly constrained and must operate unattended for multiyear lifetimes on modest batteries [11].

1.3 Objectives

The objectives of the proposed research are:

1. To develop a security protocols for Wireless Sensor Networks using IPv6.
2. To develop security that can obtain optimum performance such as using limited memory and enable to transmit a secured packet within less than their deadline time.

1.4 Scopes

A few properties will be taken into account in order to achieve the objectives that have been mentioned earlier. The scope can be divided into three phases. The first phase will be the major phase, where development of a simple algorithm for security. TinyOS operating system and network embedded systems C (nesC) programming language will be used to develop the proposed security source code based on TelosB sensor node. The developed security will be doing encryption before sending and decryption after receiving the packet through the network. The second phases will focus on integrating the developed security and with the 61owpan stack. 61owpan defines the

specification of transmitting IPv6 packets over 802.15.4 network. The third phase involves in testing the developed code by uploading it into the real sensor nodes.

1.5 Contributions of Project

The contributions of this project can be listed as follows:

1. Development of a simple, accurate and reliable XOR encryption provides security for the packet transfer. The encryption and decryption require minimum processing time and small memory size to realize the real time routing where the packets will be delivered within their end-to-end deadlines.

2. Development of XOR encryption will eliminate the needs to have a complicated and complex security algorithm where processing time usually takes up a major part of the development.

1.6 Organization of Thesis

The thesis is arranged into six chapters. The first chapter introduces the project work by discussing the background of WSNs and Security in WSNs. Besides that, it also discusses the project objectives, scope of project, project contributions and thesis organizations.

Several relevant papers from the literature are reviewed to justify the research objective. Extensive literature review is presented in Chapter 2. In addition, types of security considerations and IPv6 theory and header compression are also presented.

In chapter 3, XOR encryption system design is presented and discussed. Pioneering knowledge of encryption and decryption were briefly explained. The key distribution and concept using pseudo random generator to generate the key is explained in detail. The workflow of the project is presented with a brief description of the design.

Development of the security coding in the UDPEcho application is presented in chapter 4. The method is outlined in depth to develop the XOR encryption. Finally the integration between the XOR Encryption and the UDPEcho application is discussed in detail.

Chapter 5 focuses on implementing the security protocol on real application testbed. The verification and validation of the security coding is tested through the real testbed. The results obtained from the development are presented in this chapter.

The final chapter, chapter 6 summarizes the research finding and suggested potential future work

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Sensor networks will play an essential role in the upcoming age of pervasive computing world. Many sensor networks have mission-critical tasks, so it is clear that security needs to be taken into account. Security is sometimes viewed as a standalone component of a system's architecture, where a separate module provides security. This separation is, however, usually a flawed approach to network security. To achieve a secure system, security must be integrated into every component, since components designed without security can become a point of attack. Consequently, security must pervade every aspect of system design. There are several important security challenges, including key establishment and trust setup, secrecy and authentication, privacy, robustness to denial-of-service attacks, secure routing, and node capture [12]. The combination of these factors demands security for sensor networks to ensure operation safety, secrecy of sensitive data, and privacy for people in sensor environments.

2.2 Obstacles of Sensor Security

A wireless sensor network is a special network which has many constraints compared to a traditional computer network. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor

networks. Therefore, before creating an appropriate security for existing system, it is essential to understand about all the restrains [2].

2.2.1 **Very Limited Resources**

All security approaches require a certain amount of resources for the implementation, including data memory, code space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor.

i. Limited Memory and Storage Space

A sensor is a tiny device with only a small amount of memory and storage space for the code. In order to build an effective security mechanism, it is necessary to limit the code size of the security algorithm. For example, one common sensor type (MICAz) has 8 MHz RISC CPU with only 4K RAM, 4K EEPROM, and 128K flash storage [5]. With such a limitation, the software built for the sensor must also be quite small. Therefore, the code size for the all security related code must also be small.

ii. Power Limitation

Energy is the biggest constraint to wireless sensor capabilities. We assume that once sensor nodes are deployed in a sensor network, they cannot be easily replaced (high operating cost) or recharged (high cost of sensors). Therefore, the battery charge taken with them to the field must be conserved to extend the life of the individual sensor node and the entire sensor network. When added security code, the energy of the system must be considered. When adding security to a sensor node, we are interested in the impact that security has on the lifespan of a sensor (i.e., its battery life). The extra power consumed by sensor nodes due to security is related to the processing required for security functions such as encryption, decryption, signing data and verifying signatures.

2.2.2 Unreliable Communication

Certainly, unreliable communication is another threat to sensor security. The security of the network relies heavily on a defined protocol, which in turn depends on communication.

i. Unreliable Transfer

The packet-based routing of the sensor network is connectionless and thus inherently unreliable. Packets may get damaged due to channel errors or dropped at highly congested nodes. The result is lost or missing packets. Furthermore, the unreliable wireless communication channel also results in damaged packets. Higher channel error rate also forces the software developer to devote resources to error handling. More importantly, if the protocol lacks the appropriate error handling it is possible to lose critical security packets [2].

ii. Conflicts

Even the channel is reliable, the communication may still be unreliable. This is due to the broadcast nature of the wireless sensor network. If packets meet in the middle of transfer, conflicts will occur and the transfer itself will fail especially in a high traffic density.

iii. Latency

The multi-hop routing, network congestion and node processing can lead to greater latency in the network, thus making it difficult to achieve synchronization among sensor nodes. The synchronization issues can be critical to sensor security where the security mechanism relies on critical event reports and cryptographic key distribution.

2.2.3 Unattended Operation

Depending on the function of the particular sensor network, the sensor nodes may be left deactivate for long periods of time. There are three main caveats to unattended sensor nodes:

i. Exposure to Physical Attacks

The sensor may be deployed in an environment open to adversaries, bad weather, and so on. The likelihood that a sensor suffers a physical attack in such an environment is therefore much higher than the typical PCs, which is located in a secure place and mainly faces attacks from a network.

ii. Managed Remotely Remote

Management of a sensor network makes it virtually impossible to detect physical tampering (i.e., through tamperproof seals) and physical maintenance issues (e.g., battery replacement). Perhaps the most extreme example of this is a sensor node used for remote reconnaissance missions behind enemy lines. In such a case, the node may not have any physical contact with friendly forces once deployed.

iii. No Central Management Point

A sensor network should be a distributed network without a central management point. This will increase the vitality of the sensor network. However, if designed incorrectly, it will make the network organization difficult, inefficient, and fragile. Mostly, once a sensor nodes is left unattended it is possible that it can be compromised by the adversary

2.3 Attacks on WSNs Routing

Many sensor network routing protocols are quite simple, and for this reason are sometimes even more susceptible to attacks against general ad-hoc routing protocols. Most network layer attacks against sensor networks fall into one of the following categories:

i. Spoofed, Altered, or Replayed Routing Information

The most direct attack against a routing protocol is to target the routing information that changed between nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency, etc [13].

ii. Selective Forwarding

In a selective forwarding attack, malicious nodes may refuse to forward certain messages and simply drop them [13]. The neighbouring nodes will conclude that the current route has failed and they decided to seek another route. Selective forwarding attacks are typically most effective when the attacker is explicitly included on the path of a data flow. However, it is conceivable that an adversary overhearing a flow passing through neighbouring nodes might be able to emulate selective forwarding by jamming or causing a collision on each forwarded packet of interest. Thus, an adversary who is launching a selective forwarding attack will likely follow the path of least resistance and attempt to include herself on the actual path of the data flow. In the next two sections, we discuss sinkhole attacks and the Sybil attack, two mechanisms by which an adversary can efficiently include herself on the path of the targeted data flow.

iii. Sinkhole Attacks

In a sinkhole attack [13, 14], the adversary goal is to attract almost all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the centre. Because nodes on or near the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (selective forwarding, for example). Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high quality route to a base station. Some protocols might actually try to verify the quality of route with end-to end acknowledgements containing reliability or latency information. In this case, a laptop-class adversary with a powerful transmitter can actually provide a high quality route by transmitting with enough power to reach the base station in a single hop, or by using a wormhole attack discussed in the following section. Due to either the real or imagined high quality route through the compromised node, it is likely each neighbouring node of the adversary will forward packets destined for a base station through the adversary, and propagate the attractiveness of the route to its neighbours. One motivation for mounting a sinkhole attack is that it makes selective forwarding trivial. By ensuring that all traffic in the targeted area flows through a compromised node, an adversary can selectively suppress or modify packets originating from any node in the area.

iv. The Sybil Attack

In a Sybil attack [15], a single node presents multiple identities to other nodes in the network. The Sybil attack can significantly reduce the effectiveness of fault tolerant schemes such as distributed storage, dispersity [15], multi-path routing, and topology maintenance. Replicas, storage partitions, or routes believed to be using disjoint nodes could in actuality be using a single adversary presenting multiple identities. Sybil attacks also pose a significant threat to geographic routing protocols. Location aware routing often requires nodes to exchange coordinate information with their neighbours to efficiently route geographically addressed packets. It is only

reasonable to expect a node to accept but a single set of coordinates from each of its neighbours, but by using the Sybil attack an adversary can "be in more than one place at once" [13].

v. Wormholes

In the wormhole attack [13, 16], an adversary tunnels messages received in one part of the network over a low latency link and replays them in a different part. The simplest instance of this attack is a single node located between two other nodes forwarding messages between the two of them. An adversary located close to a base station may be able to disturb routing by creating a well-placed wormhole. An adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole. This can create a sinkhole: since the adversary on the other side of the wormhole can artificially provide a high-quality route to the base station, potentially all traffic in the surrounding area will be drawn through her if alternate routes are significantly less attractive.

vi. HELLO Flood Attack

Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbours, and a node receiving such a packet may assume that 39 it is within radio range of the sender [13]. This assumption may be false; a laptop class attacker which is broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbour. For example, an adversary who is advertising a very high quality route to the base station to every node in the network could cause a large number of nodes to attempt to use this route, but those nodes sufficiently far away from the adversary would be sending packets into oblivion. The network is left in a state of confusion. A node realizing the link to the adversary is false could be left with few options: all its neighbours might be attempting to forward packets to the adversary as well. Protocols that depend on localized information exchange between neighbouring nodes are also subject to this attack. An adversary does not necessarily need to be

able to construct legitimate traffic in order to use the HELLO flood attack. This can simply re-broadcast overhead packets with enough power to be received by every node in the network. HELLO floods can also be thought of as one-way, broadcast wormholes. It is interesting to note that flooding usually used to denote propagation of a message to every node in the network over a multi-hop topology. In contrast, despite its name, the HELLO flood attack uses a single hop broadcast to transmit a message to a large number of receivers.

vii. Acknowledgement Spoofing

Several sensor network routing algorithms rely on implicit or explicit link layer acknowledgements. Due to the inherent broadcast medium, an adversary can spoof link layer acknowledgments for overhead packets addressed to neighbouring nodes [13]. Goals include convincing the sender that a weak link is strong or that a dead or disabled node is alive. For example, a routing protocol may select the next hop in a path using link reliability. Artificially reinforcing a weak or dead link is a clever way of manipulating such a scheme. Since packets sent along weak or dead links are lost, an adversary can effectively mount a selective forwarding attack using acknowledgement spoofing by encouraging the target node to transmit packets on those links.

2.4 Secure Routing in WSNs

In this section, several solutions that had been designed for securing WSNs will be discussed.

i. Secure Real-Time Routing Protocol For Wireless Sensor Networks by Dr. Adel Ali Ahmed (2008)

Secure Real time Load Distributions [17], SRTLD is a built in security, where it depends on random selection of next hop. Random selection of next hop

uses location aided routing and multipath forwarding contributes to build in security in WSNs. Random selection of next hop depends on PRR, packet velocity and remaining power. To enhance the security in SRTLD, it includes security power management module to ensure more security in WSN. Further supplement for secure the packet transfer, encryption and decryption with authentication of the packet header is done. The simulation evaluates the capability of SRTLD to overcome the HELLO flood attack and selective forwarding attack. A realistic simulation environment for SRTLD was created based on the physical characteristics of MICAz. SRTLD takes less than 150ms packet delay to forward a packet through 10 hops. For encrypt, decrypt and authenticate the header of the packet, it takes around 4.2 ms. WSNs with SRTLD routing doesn't affected by increasing the number of compromised nodes up to 20%.

ii. Tiny Sec : A Link Layer Security Architecture for Wireless Sensor Networks by C. Karlof, N.Sastry, D. Wagner.

TinySec by [18] presents two different security options: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). In TinySec-AE, TinySec encrypts the data payload and authenticates the packet with a message authentication code. The message authentication code is computed over the encrypted data and the packet header. In TinySec-Auth, TinySec authenticates the entire packet with a message authentication code, but the data payload is not encrypted. TinySec currently runs on the Mica, Mica2, and Mica2Dot platforms. TinySec-AE and TinySec-Auth increase packet latencies (compared to the current TinyOS stack) by 56.6 ms and 53.4 ms respectively. TinySec does not provide a secure localisation, secure routing mechanism while the proposed security algorithm addresses these weaknesses. It requires 728 bytes of RAM and 7146 bytes of program space.

iii. TEA: Tiny Encryption Algorithm by D.J Wheeler and R.M. Needham

Researchers in Cambridge University proposed an extremely simple encryption algorithm, the Tiny Encryption Algorithm (TEA) [19]. It is based on an

alternative application of a large number of iterations with XORs and additions operations, rather than on preset tables. Better performance was achieved with smaller code size and less complexity than standard encryption algorithms. The overall running time of the TEA algorithm per one hop is 28.176 ms and it requires 1140 bytes of memory size [20].

iv. Enhancing Base Station Security in Wireless Sensor Networks by Jing Deng etal.

Enhancing Base Station Security in WSNs is proposed in [21]. This security method devises two different solutions, one applicable during the route discovery phase and the other applicable after the route discovery phase. The authors in [21] made real experiment to estimate the required processing time to implement the security operation in MICA2 mote. They use nesC programming language, and they choose RC5 (with 12 rounds) as block cipher. The standard random number generator linear feedback shift register (LFSR) is used for hash function reversing. They do ID confusion for 32 bytes of data with 4 bytes as nodes addresses. The execution time in LFSR is 48 ms to forward a packet per one-hop. LFSR requires 1300 bytes in ROM and 110 bytes in RAM.

v. Implementation of Tiny Hash based on Hash Algorithm for Sensor Network by Hang Rok Lee

TinyHash based on hash algorithm for WSNs is a modified security method for TinySec [22]. It implements security components for message hash and authentication using Secure Hash Algorithm 1 (SHA1) function instead of using Cipher Block Chaining-Message Authentication Code (CBC-MAC) based on Skipjack, which is offered in TinySec. TinyHash uses Hash Message Authentication Code (HMAC) scheme for authentication and SHA1 hash algorithm for message digest. The security components were designed by a similar architecture with TinySec. In order to have compatibility with TinySec, TinyHash implements interfaces for wiring these components. Then execution time and memory size were

tested for SHA1 hash function implemented over TinyHash. TinyHash is implemented on Telos mote using SHA1 8bits version in order to have the compatibility with MICA mote series. SHA1 requires 140 bytes of RAM and 3504 bytes ROM. Also, it requires 35ms execution time for 160 bits data packet. HMAC has twice the size of SHA1 algorithm and has twice the execution time of SHA1.

vi. One-Time Pad by Major Joseph Mauborgne and AT & T's Gilbert Vernam

One-time pad was invented by [23]. It is a very simple security system and is unbreakable [26]. The pad is a block of random data equal in length to the original message and one copy of the pad is kept by each user. If the data on the pad is not truly random, the security of the pad is reduced. The pad is used by XORing every bit of the pad with every bit of the original message. Once the message is encoded with the pad, the pad is destroyed and the encoded message is sent. On the recipient side, the encoded message is XORed with the duplicate copy of the pad and the plaintext message is generated. The drawbacks of this mechanism are producing real random number is complicated and a one-time pad does not provide data authenticity and encryption scheme.

2.5 IPv6

IPv6 (Internet Protocol, version 6) also known as IPng (Internet Protocol, Next Generation) was proposed and is now a standard. In IPv6, the internet protocol was extensively modified to accommodate the unseen growth of the internet. (IPv6) [24] is a layer 3 best-effort transport protocol. It is the new version of the Internet Protocol, designed as the successor to IP version 4. Changes from IPv4 primarily include the expansion of the IP address size from 32 to 128 bits, header format simplification, improved support for extensions and options, flow labeling capability, authentication extensions and privacy extensions.

The new address space thus supports 2 (about 3.4x10) addresses. This expansion provides flexibility in allocating addresses and routing traffic and eliminates the need for network address translation (NAT). NAT gained widespread deployment as an effort to alleviate IPv4 address exhaustion.

On the other hand Core IPv6 components, such as Neighbour Discovery (ND), use link-local scoped multicast for address resolution, duplicate address detection (DAD), and router discovery. Stateless address autoconfiguration (SAA) simplifies configuration and management of IPv6 devices by enabling nodes to assign themselves meaningful addresses [11].

2.6 IEEE 802.15.4

IEEE 802.15.4 was designed specifically for long-lived application domains that require numerous low-cost nodes, and these constraints limit the capability of LoWPAN links and the microcontrollers to which they're attached. Throughput is limited to 250 kbps in the 2.4-GHz band and 20 or 40 kbps in other bands. The frame length is limited to 128 bytes to ensure reasonably low packet error when bit-error rates are non-negligible and reflect microcontrollers' limited buffering capabilities.

Additionally, the microcontrollers typically coupled with LoWPAN radios have limited memory and compute power. 802.15.4 defines short 16-bit link addresses, in addition to IEEE EUI-64 addresses, to reduce header overhead and memory requirements. IEEE 802.15.4 protocol specifies a global standard on physical and MAC layers for low data rate, low power, low cast and short range that make IEEE 802.15.4 suitable for WSNs [25, 2],

Two different device types can participate in an LR-WPAN network; a full-function device (FFD) and a reduced-function device (RFD). The IEEE 802.15.4 may operate either in the star topology or peer-to-peer topology.

2.7 61owpan

As mentioned before, 61owpan is a working group within the IETF concerned with the specification of transmitting IPv6 packets over IEEE 802.15.4 networks [26]. Creation of IPv6 link-local addresses and statelessly autoconfigured addresses on top of 802.15.4 networks. The followings are the characteristics of 6L0WPAN where it has a small packet size. It has 16-bit short or IEEE 64-bit extended media access control addresses [27, 28],

Besides the networks are ad-hoc and devices have limited accessibility and user interface. The IEEE 802.15.4 standard specifies an MTU of 128 bytes, (including the length byte) on a wireless link with a link throughput of 250 kbps or less. The IEEE 802.15.4 standard targets low-power wireless personal area networks (LoWPANs). Such LoWPANs consist of devices characterized by short range, low bit rate, low power and low cost. As a result, these devices typically are severely constrained and have only limited capabilities.

IPv6 is the workhorse for data delivery for wired networks- the Internet. Likewise, IEEE 802.15.4 devices provide sensing communication ability in the wireless domain.

In 61owpan, to avoid packet fragmentation and the overhead assembly, all the data to be transmitted should fit in a single IEEE 802.15.4 physical frame. The routing control packets are placed after the 61owpan dispatch where multiple routing protocols can be supported by the usage of different Dispatch bit sequences. Figure 2.1 shows the place of 61owpan mechanisms, such as mesh routing control packets, in a big picture of an IEEE 802.15.4 physical frame.

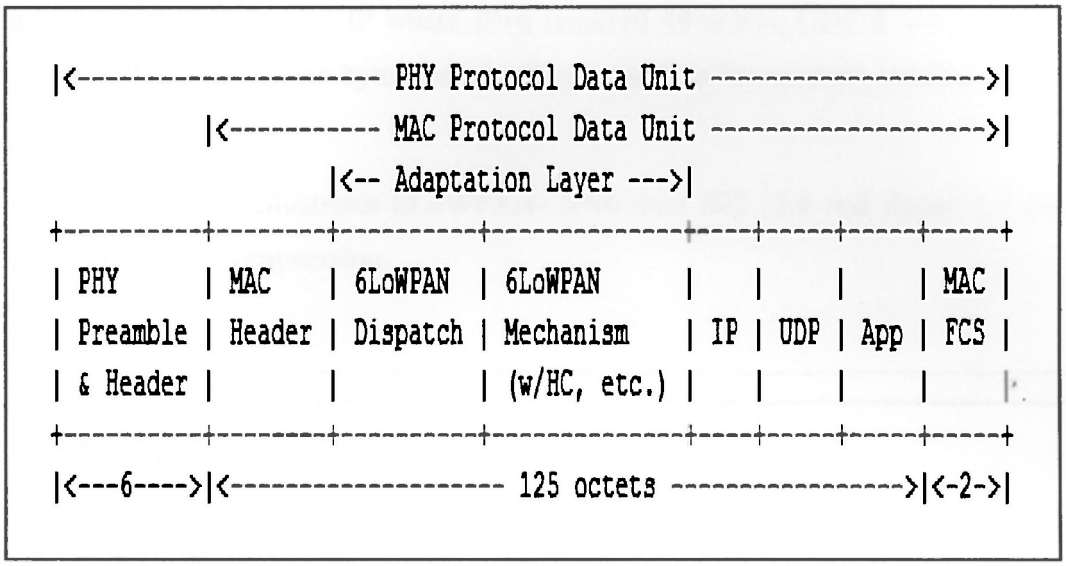

Figure 2.1: IEEE 802.15.4 Physical Frame

2.8 Challenges in 6LoWPAN

Challenges in 6LoWPAN can be divided into two categories:

i. Limited Packet Size

LoWPAN has limited RAM and storage. Due to it the protocols for low end LoWPAN should be designed such that control packets fit within a single 802.15.4 frame. The main constraint is where the transmission will be in one frame although all layers will be added for IP connectivity. Fragmentation and reassembly will be avoided As a result, headers for IPv6 and layers above must be compressed whenever IPv6 over IEEE 802.15.4.

ii. Packet Overhead

In fact, IEEE 802.15.4 can be best providing 102 bytes due to a maximum frame overhead of 25 octets are spares at the media access control. Contrariwise IPv6 requires 1280 bytes packet compare to entire IEEE 802.15.4's standard. Following

are sizes of the layers in IP where IPv6 required 40 octets, UDP 8 octets, TCP 20 octets and additional few bytes for other layers such as for security, routing and etc.

Figure 2.2 illustrates 6LoWPAN- IPv6 over 802.15.4 and figure 2.3 shows the IPv6 header compression.

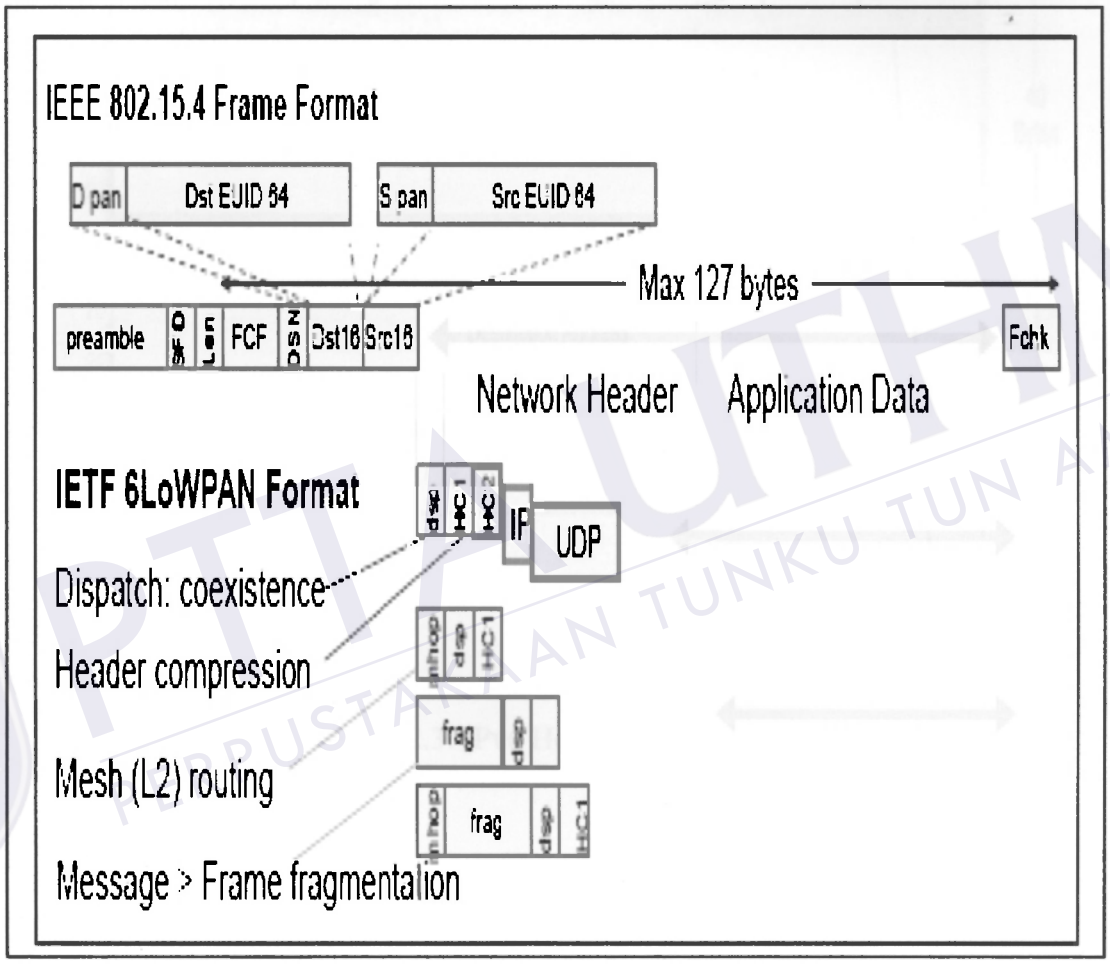

Figure 2.2: 6LoWPAN- IPv6 over 802.15.4

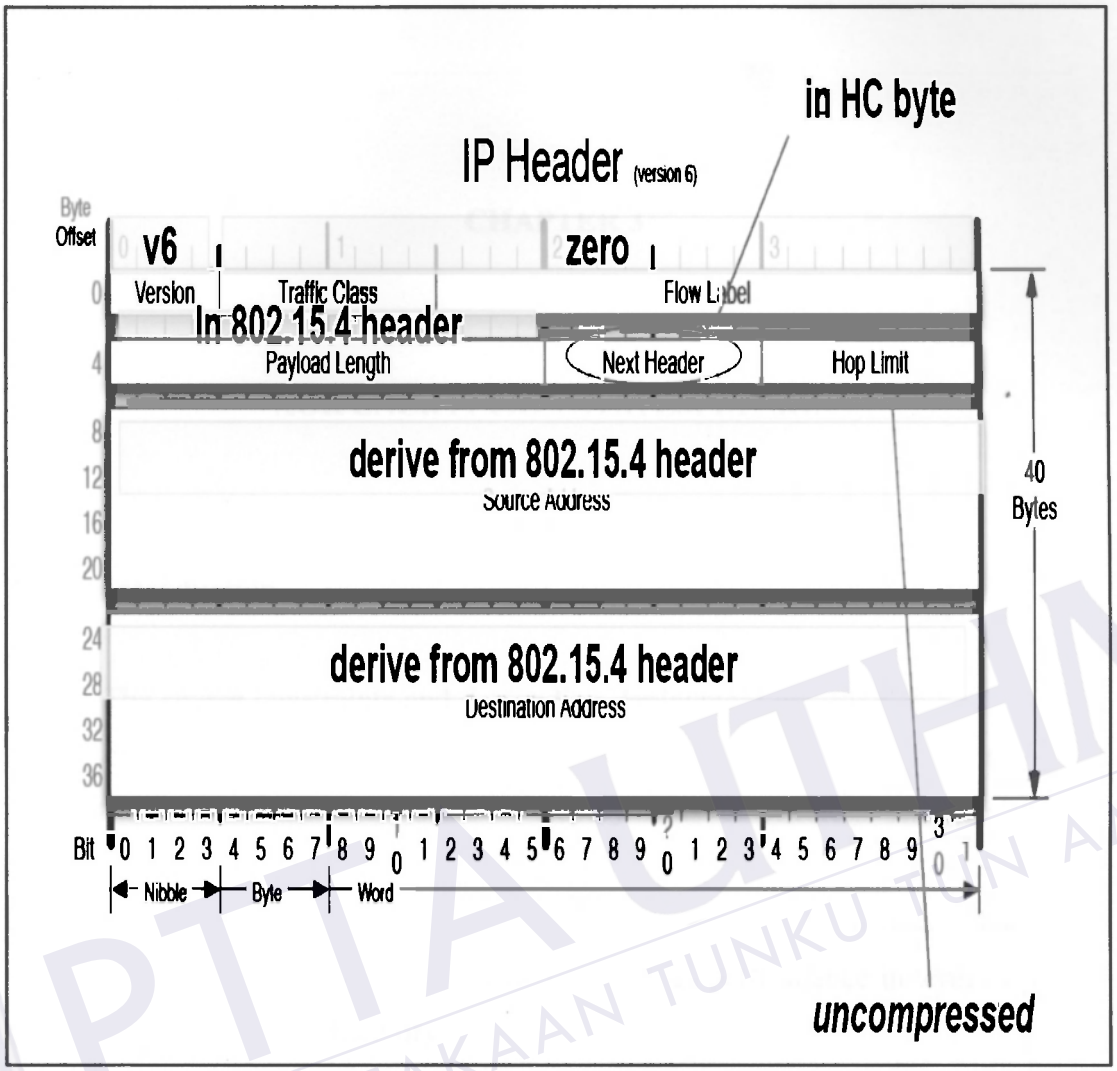

Figure 2.3: IPv6 Header Compression

2.9 Summary

Previous work that has been done in the area of security in WSNs, obstacles in security, IPv6 theory, 6lowpan theory, IEEE 802.15.4 theory and Ipv6 header compression has been reviewed in this chapter. Some of the ideas and methodology in the literature have proved to be useful for this project. The XOR method for encryption and decryption suggested in [23] and [19] for achieving minimum process time was implemented in this project. Other than that, the pseudo random generator suggested in [17] was used to find the key which is a prime number was also used in this project.

REFERENCES

1. F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *A Survey on Sensor Networks,* IEEE Communication Magazine, 2002.
2. K. Romer, F. Mattern. *The Design Space of Wireless Sensor Networks, IEEE Wireless Communications* Journal, Volume 11, Issue 6, Dec. 2004 Page(s): 54-61.
3. Kazem Sohraby, Daniel Minoli, and Taieb Znati *Introduction and Overview Of Wireless Sensor Networks. Wireless Sensor Networks: Technology, Protocols, and Applications.* Copyright # 2007 John Wiley & Sons, Inc
4. J.N. Al-Karak and A.E. Kamal, *Routing Techniques in Wireless Sensor Networks: A Survey, Wireless Communications* IEEE journal, Volume: 11, Dec. 2004. Pages. 6- 28,
5. Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci *A Survey on Wireless Sensor Network.* Georgia Institute of Technology.
6. M. Welsh, D. Malan, B. Duncan, T. Fulford-Jones, S. Moulton, " *Wireless Sensor Networks for Emergency Medical Care*," presented at GE Global Research Conference,Harvard University and Boston University School of Medicine, Boston, MA, Mar. 8, 2004.
7. Yang Xiao,(Eds.) *Security in Distributed, Grid, and Pervasive Computing* pp. 2006 Auerbach Publications, CRC Press Chapter 17 Wireless Sensor Network Security:
8. J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary *Wireless Sensor Network Security: A Survey, 2006* Department of Computer Science Wayne State University
9. Mayank Saraogi *Security In Wireless Sensor Networks.* Department of Computer Science University of Tennessee

10. M. Chen, V.C.M. Leung, S. Mao, Y. Yuan. *Directional Geographical Routing for Real-Time Video Communications in Wireless Sensor Networks,* Elsevier Computer Communications Journal, Volume 30, Issue 17, November 2007. Pages: 3368-3383.
11. Jonathan W. Hui Arch Rock David E. *Culler Extending IP to Low-Power, Wireless Personal Area Networks* University of California, Berkeley
12. Adrian Perrig, John Stankovic, and David Wagner. *Security In Wireless Sensor Networks.*
13. C. Karlof and D. Wagner. *Secure routing in wireless sensor networks: attacks and countermeasures,* Elsevier's Ad Hoc Network Journal, Special Issue on Sensor Network Applications and Protocols, September 2003. Pages 293-315
14. A.K. Pathan, H.W Lee, C.S Hong. *Security in Wireless Sensor Networks: Issues and Challenges,* Proceedings of 8th IEEE ICACT 2006, Volume II, February 20-22, Phoenix Park, Korea, 2006. Pages: 1043-1048.
15. Baneijea, *"A Taxonomy Of Dispersity Routing Schemes For Fault Tolerant Real-Time Channels*," in Proceedings of ECMAST, vol. 26, May 1996, pp. 129-148.
16. Y.-C. Hu, A. Perrig, D.B. Johnson, *Packet Leashes: A Defense Against Wormhole Attacks In Wireless Networks,* in IEEE Infocom, 2003.
17. Adel Ali Ahmed, *Secure Real Time Routing Protocol for Wireless Sensor Networks,* Universiti Teknologi Malaysia, 2008.
18. C. Karlof, N. Sastry, D. Wagner,. *TinySec: a link layer security architecture for wireless sensor networks.* Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, Maryland. November 2004.
19. D. J. Wheeler and R. M. Needham. *TEA: a Tiny Encryption Algorithm. In Fast Software Encryption,* Second International Workshop Proceedings, SpringerVerlag, pages 97-110, 1995.
20. Y. Kanamori, E. Jovanov, and S.-M. Yoo. *Performance comparison between tea and rijndale encryption algorithm for wireless sensor networks.* In ISCA 15th International Conference on Computer Applications in Industry and Engineering (CAINE), San Diego, pages 209-212, Nov. 2000.

21. Deng, R. Han and S. Mishra, *Enhancing Base Station Security in Wireless Sensor Networks,* Technical Report CU-CS-951-03, Department of Computer Science, University of Colorado, April 2003.
22. HangRok Lee, YongJe Choi, Ho Won Kim. Implementation Of Tinyhash Based On Hash Algorithm For Sensor Network. Transaction on Engineering, Computing and Technology, Krakow, Poland, Volume 10, December, 2005.
23. D. Kahn, *The Codebreakers: The Story of Secret Writing,* New York: Macmillan Publishing Co., 1967.
24. Mat'iTs Harvan, "*Connecting wireless Sensor Networks to the internet - a 6lowpan Implementation for TinyOS* 2.0,May 2007.
25. E. Felemban, C. G. Lee, E. Ekici, R. Boder and S. Vural, *Probabilistic Qos Guarantee In Reliability And Timeliness Domains In Wireless Sensor Networks,* 24th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Proceedings, 2005. Pages: 2646 - 2657.
26. D. W. Carman, P. S. Krus, and B. J. Matt. *Constraints And Approaches For Distributed Sensor Network Security.* Technical Report 00-010, NAI Labs, Network Associates, Inc., Glenwood, MD, 2000.
27. Nandakishore Kushalnagar, Gabriel Montenegro, and Christian Peter Pii Schumacher. *6LoWPAN: Overview, Assumptions, Problem Statement and Goals.* Internet-Draft Version 08, IETF, February 2007.
28. Gabriel Montenegro, Nandakishore Kushalnagar, David E. Culler, and Jonathan W. Hui. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks.* Internet-Draft Version 13, IETF, April 2007.
29. Stallings, W. (2004). **Data** *and Computer Communications.* Pearson Prentice Hall, Upper Saddle River, NJ, pp. 670-677.
30. Forouzan,B.A, Fegan.S.C .*TCP/IP Protocol Suite,* McGraw Hill Higher Education, Second Edition Avenue of the America, NY, pp. 728-745
31. Sklar.B *Digital Communicatios. Fundamental and Applications, Communication* Enginering Services, Tarzana, California and University of California, Los Angeles. Pearson Prentice Hall, Upper Saddle River, NJ,pp 728-732
32. *Pseudo Random Number Generator (PRNG) Definition, Theory and Information,* http://en.wikipedia.org/wiki/Pseudorandom_number_generator. Accessed on 2nd March 2009.

33. *Various Techniques Used In Connection With Random Digits,* Applied Mathematics Series, no. 12, 36-38 (1951).
34. *TinyOS tutorial. Introduction into interfaces and Components,* http://docs.tinyos.net/index.php/Getting Started with TinyOS#Components and Interfaces. Accessed on 13th November 2008.
35. *TinyOS tutorial Modules and TinyOS Execution Model* http://docs.tinyos.net/index.php/Modules and the TinyOS Execution Mod el. Accessed on 14th November 2008
36. David Gay, Philip Levis, David Culler, Eric Brewer *nesC 1.1 Language Reference Manual* May 2003
37. *Sensor Node in Wireless Sensor Network TelosB Information* http://www.xbow.com/Products/Product pdf files/Wireless pdf/TelosB Pat asheet.pdf. Accessed on 20th August 2008
38. *Ubuntu Operating system Information, Description and How to download.* http://en.wikipedia.org/wiki/Ubuntu_(Linux_distribution). Accessed on 27th December 2008
39. *TinyOS Information,* http://en.wikipedia.org/wiki/TinyOS ,Accessed on 31/08/2008.
40. B. W. Kernighan and D. M. Ritchie. *The C Programming Language,* Second Edition. Prentice Hall, 1988.
41. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. *System Architecture Directions for Networked Sensors. In Architectural Support for Programming Languages and Operating Systems,* pages 93-104, 2000.
42. D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, and D. Culler. *The nesC Language: A Holistic Approach to Networked Embedded Systems. In PLDI03.* ACM, June 2003.
43. Sharpe.R, Lamping.U *Wireshark User's Guide 28195 for Wireshark 1.0.0,* NS Computer Software and Services P/L Ed Warnicke.
44. Dev-C++ compiler Information http://en.wikipedia.org/wiki/Dev-C++#cite note-0. Accessed on 19th February 2009.
45. *Chipcon. CC2420 low power radio transceiver,* http://www.chipcon.com. Accessed on 23rd January 2008.