3-1-2009

# The Evaluation of Rekeying Protocols within the Hubenko Architecture as Applied to Wireless Sensor Networks

Cory J. Antosh


# THE EVALUATION OF REKEYING PROTOCOLS WITHIN THE HUBENKO ARCHITECTURE AS APPLIED TO WIRELESS SENSOR NETWORKS

THESIS

Cory J. Antosh, Captain, USAF

AFIT/GE/ENG/09-04

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/09-04

# THE EVALUATION OF REKEYING PROTOCOLS WITHIN THE HUBENKO ARCHITECTURE AS APPLIED TO WIRELESS SENSOR NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Cory J. Antosh, BS EE

Captain, USAF

March 2009

AFIT/GE/ENG/09-04

# THE EVALUATION OF REKEYING PROTOCOLS WITHIN THE HUBENKO ARCHITECTURE AS APPLIED TO WIRELESS SENSOR NETWORKS

Cory J. Antosh, BS EE

Captain, USAF

Approved:

Barry Mullins
Dr. Barry E. Mullins (Chairman) — 17 Feb 09 — Date

Rusty O. Baldwin
Dr. Rusty O. Baldwin (Member) — 17 Feb 09 — Date

Richard A Raines
Dr. Richard A. Raines (Member) — 17 Feb 09 — Date

AFIT/GE/ENG/09-04


## Abstract

This thesis investigates the impact of using three different rekeying protocols–pair-wise, hierarchical, and Secure Lock within a wireless sensor network (WSN) under the Hubenko architecture. Using a Matlab computer simulation, the impact of the three rekeying protocols on the number of bits transmitted across the network and the amount of battery power consumed in WSN nodes during rekey operations is investigated.

Baseline pair-wise rekeying performance can be improved by using either Secure Lock or hierarchical rekeying. The best choice depends on the size of the WSN and the size of the key used. Hierarchical rekeying is the best choice for networks with 500 or more nodes using a key size of 512 bits. It is also the best choice for a network of 1,000 nodes using a 256-bit key. For smaller networks with shorter key sizes, Secure Lock is the best choice.

Overall, the number of bits transmitted for rekey operations can be reduced 3.32% to 75.80% and the battery power savings range from 0.03% to 39.94% compared to pair-wise keying. Based on the number of bits transmitted, the savings in battery power and the amount of memory required, hierarchical keying is clearly the best approach for network sizes of 1,000 nodes or more utilizing a key with 128 bits or more in length. For smaller network sizes, Secure Lock can be beneficial, but any savings over hierarchical keying are offset by the weaker security scheme and increased complexity of Secure Lock.

## Acknowledgments

I appreciate the sacrifices my family has made over the past eighteen months. I am grateful for the continued love and support from my wife and our three children. I am blessed to have them in my life.

I thank my thesis advisor, Dr. Mullins for his expert guidance and support. He always knew how to provide "just enough" direction to get me back on track. I truly appreciate the freedom he gave me to conduct this research.

I am also thankful to my thesis committee members, Dr. Baldwin and Dr. Raines for their support.

## Table of Contents

Page

Abstract ..... 1

Acknowledgments ..... 2

Table of Contents ..... 3

List of Figures ..... 5

List of Tables ..... 7

I. Introduction ..... 8

1.1 Motivation ..... 8
1.2 Overview and Goals ..... 9
1.3 Experimental Approach ..... 10
1.4 Thesis Layout ..... 10

II. Background and Literature Review ..... 11

2.1 Mobile Ad Hoc Networks (MANETs) ..... 11
2.2 Wireless Sensor Networks ..... 12
2.3 Unmanned Aerial Vehicles ..... 16
2.4 Multicast Communications ..... 18
2.5 Internet Group Management Protocol (IGMP) ..... 19
2.6 Gothic Group Membership Authentication ..... 20
2.7 Encryption ..... 21
2.8 Key Management in Multicast Groups ..... 27
2.9 The Hubenko Architecture ..... 30
2.10 Hubenko Architecture Applied to Swarms of Autonomous UAVs ..... 33
2.11 Summary ..... 36

III. Methodology ..... 37

3.1 Problem Definition ..... 37
3.2 System boundaries ..... 39
3.3 System Services ..... 40
3.4 Workload ..... 40
3.5 Performance Metrics ..... 41
3.6 System Parameters ..... 42
3.7 Factors ..... 43
3.8 Evaluation Technique ..... 44

3.9 Experimental Design ..... 45
3.10 Summary ..... 45

IV. Results and Analysis ..... 46

4.1 Model Verification and Validation ..... 46
4.2 Results and Analysis of Performance Metrics ..... 54
4.3 Overall Analysis ..... 78
4.4 Summary ..... 79

V. Conclusions and Recommendations ..... 80

5.1 Conclusions of Research ..... 80
5.2 Significance of Research ..... 82
5.3 Recommendations for Future Research ..... 82
5.4 Summary ..... 83

Appendix A. Creation of Secure Locks ..... 84

Appendix B. Communications Link Budget ..... 86

Appendix C. Plots of Log Number of Bits Transmitted ..... 90

Appendix D. Plots of Mean Number of Bits Transmitted ..... 92

Bibliography ..... 95

Vita ..... 99

## List of Figures

Figure Page

1. Crossbow IRIS Wireless Sensor Node [Cro08] ..... 13

2. Wireless Sensor Network Architecture ..... 15

3. Comparison of Unicast to Multicast Communication ..... 19

4. Hierarchical Tree ..... 25

5. Secure Lock Message Size [AnM08] ..... 27

6. The Hubenko Architecture [Hub08] ..... 32

7. Average User Re-Key Count Comparison [Hub08] ..... 33

8. Hubenko Architecture Applied to UAV Swarm [Phi08] ..... 34

9. Conceptual View of Network ..... 38

10. System Under Test: the Rekey Improved Hubenko Architecture ..... 39

11. Verification and Validation of Secure Lock Function ..... 48

12. Verification and Validation of Hierarchical Function ..... 49

13. Model Validation with a 75% Departure Rate ..... 51

14. Model Validation with a 25% Departure Rate ..... 52

15. Visual Plots to Verify ANOVA Assumptions for All Data ..... 55

16. Main Effects Plots for All Data ..... 56

17. Visual Plots to Verify ANOVA Assumptions for WSN Size = 40 Nodes ..... 58

18. Main Effects Plots for WSN Size = 40 ..... 59

19. Factor Interaction Plot for WSN Size = 40 Nodes ..... 60

20. Visual Plots to Verify ANOVA Assumptions for WSN Size = 100 Nodes ..... 61

21. Main Effects Plots for WSN Size = 100 ..... 62

22. Factor Interaction Plot for WSN Size = 100 Nodes ..... 63

23. Visual Plots to Verify ANOVA Assumptions for WSN Size = 500 Nodes .....64

24. Main Effects Plots for WSN Size = 500 .....65

25. Factor Interaction Plot for WSN Size = 500 Nodes .....66

26. Visual Plots to Verify ANOVA Assumptions for WSN Size = 1000 Nodes .....67

27. Main Effects Plots for WSN Size = 1,000 .....68

28. Factor Interaction Plot for WSN Size = 1,000 Nodes .....69

29. Plot of Log Number of Bits Transmitted for WSN Size = 40 .....90

30. Plot of Log Number of Bits Transmitted for WSN Size = 100 .....90

31. Plot of Log Number of Bits Transmitted for WSN Size = 500 .....91

32. Plot of Log Number of Bits Transmitted for WSN Size = 1000 .....91

33. Plot of Number of Bits Transmitted for WSN Size = 40.....92

34. Plot of Number of Bits Transmitted for WSN Size = 100.....92

35. Plot of Number of Bits Transmitted for WSN Size = 500.....93

36. Plot of Number of Bits Transmitted for WSN Size = 1000.....93

37. Plot of Data Presented in Table 15. .....94

## List of Tables

Table Page

1. Power Consumption of Crossbow IRIS [Cro08] ..... 14

2. CPU and Memory Specifications of Crossbow Devices [Cro08] ..... 14

3. Operational Characteristics of UAVs [Gen 08, NoG07, USA08] ..... 17

4. Rates of Mobility Sets [HRB08] ..... 32

5. Factor Levels Scenario 1 [Phi08] ..... 35

6. Factor Levels Scenario 2 [Phi08] ..... 35

7. Reductions Observed [Phi08] ..... 36

8. Factor Levels ..... 43

9. Factor Levels Used for Validation ..... 50

10. ANOVA Results for Log Number of Bits Transmitted for All Data ..... 56

11. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 40 Nodes ..... 58

12. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 100 Nodes ..... 61

13. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 500 Nodes ..... 64

14. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 1000 Nodes ..... 68

15. Mean Number of Bits Transmitted ..... 71

16. Mean Number of Bits Transmitted: Percentage of Baseline (pair-wise) ..... 72

17. Time Required (seconds) to Transmit the Mean Number of Bits From the Relay ..... 74

18. Percentage of Total Battery Life Saved in Comparison to Pair-wise Keying (RX) ..... 75

19. Percentage of Total Battery Life Saved in Comparison to Pair-wise Keying (TX) ..... 76

20. Memory (bytes) Occupied by Rekey Message ..... 77

21. Percentage of Increase in Memory Space Used Over Pair-wise Keying ..... 78

22. Best Rekey Protocol for given WSN size and Key Size ..... 81

# THE EVALUATION OF REKEYING PROTOCOLS WITHIN THE HUBENKO ARCHITECTURE AS APPLIED TO WIRELESS SENSOR NETWORKS

## I. Introduction

### 1.1 Motivation

Wireless Sensor Networks (WSNs) began as a joint initiative between the Defense Advanced Research Projects Agency (DARPA), Intel, and the University of California, Berkeley to create an operating system specifically designed to enable these networks [Ten01]. Believing WSNs could one day revolutionize warfare, DARPA provided initial funding to create a specialized operating system for WSN devices called TinyOS. Since then, the potential military applications of these tiny, wireless, networked sensor platforms have grown immensely. The need to sense the environment, coupled with the ever diminishing size and cost of electronic devices has yielded the emergent technology of Wireless Sensor Networks. As the cost of WSN devices decreases and their capabilities increase, the possible uses for WSNs are almost boundless. However, since the battery life of WSN devices is limited, every WSN operation must be as efficient as possible.

To be of military value, WSNs must have secure communications; yet every processor instruction executed and every bit transmitted consumes battery power. Preserving battery life is important since WSN nodes cease to operate once their batteries are consumed. One promising power conservation approach incorporates the Hubenko architecture into the WSN. Different rekeying protocols are evaluated based on the power consumption and efficiency in rekeying WSN nodes while maintaining the security of the entire WSN.

One use of WSNs is as a replacement for landmines. In this application, the WSN nodes are outfitted with acoustic, pressure, or magnetic sensors. Microphones detect the acoustic signature of approaching soldiers or equipment, while magnetic sensors could detect large metallic objects moving nearby, such as tanks, and pressure switches could detect if they are stepped on or run over. If movement is detected, the WSN can alert friendly forces who can then choose how to respond – unlike a conventional landmine which would indiscriminately detonate. The response could vary from investigation by friendly forces, to artillery, missile, or aircraft fire on the area. Once the area no longer needs to be monitored, the WSN nodes can be left behind since no lingering threat of buried explosives remain. This eliminates the hazard of abandoned landmines in an area; a hazard that maims or kills hundreds of civilians every day around the world [ICB06].

### 1.2 Overview and Goals

This research evaluates the performance of the Hubenko architecture using three different rekeying protocols in the context of a WSN under a variety of configurations.

The goals of this research include:

- Improve the efficiency of the rekeying operation within a WSN.
- Investigate whether using hierarchical keying or Secure Lock [ChC89] provides measurable benefits.
- Evaluate the impact of these rekeying protocols on the resource constrained nodes of a WSN.
- Determine which rekeying protocol should be applied given particular network parameters, such as WSN size.

## 1.3 Experimental Approach

A computer simulation in Matlab is used to determine the impact of the three rekeying protocols using 144 combinations of factors with ten experimental runs each. The simulation measures the number of bits transmitted at a central communications relay in the network for three different rekeying protocols-pair-wise, hierarchical, and Secure Lock for a simulated duration of 30 days. In addition to the rekeying protocol, the size the network, the size of the key, the departure rate, and the rate of node mobility are varied across all combinations, resulting in 144 unique combinations. Each factor combination of factors is replicated ten times.

## 1.4 Thesis Layout

This chapter introduces the research topic and the motivation for the effort. In Chapter 2, background information and fundamental concepts are presented as well as recent work in the area. Chapter 3 outlines the methodology used to carry out the experiments. Chapter 4 provides discussion and analysis of the experimental results. Chapter 5 draws conclusions about the results and suggests areas for future research.

# II. Background and Literature Review

This chapter presents fundamental concepts and recent research in the areas of Unmanned Aerial Vehicles (UAVs), Wireless Networks, Distributed Sensor Networks, securing wireless networks, multicasting technology, and secure, scalable multicast architectures. Section 2.1 introduces mobile ad hoc networks, followed by a discussion of wireless sensor networks in Section 2.2. Section 2.3 defines UAVs, presents some typical UAV operational characteristics, the expanding need for UAVs in the battlespace, and some applications. Section 2.4 presents multicasting. Section 2.5 covers the Internet Group Management Protocol (IGMP), and Section 2.6 discusses the group membership functions of Gothic. Encryption techniques and key distribution techniques are discussed in Section 2.7. Section 2.8 presents key management in multicast groups. Section 2.9 presents the Hubenko architecture and experimental results from previous research. Section 2.10 presents experimental research from adapting the Hubenko architecture to autonomous swarms of UAVs, along with experimental results. Section 2.11 summarizes the chapter.

## 2.1 Mobile Ad Hoc Networks (MANETs)

MANETs are:

> Self-configuring networks comprised of mobile nodes that adopt a completely arbitrary topology that can change rapidly and unpredictably. In a fully distributed set up, the infrastructure lacks any centralized authenticator and uses distributed algorithms to support access control. … Typical applications are rescue operations or tactical networks. [FMA08]

MANETs differentiate themselves from traditional networks by relying on no fixed infrastructure, such as base stations, access points, or remote servers [MDM07]. This allows ad hoc networks to be set up quickly and economically, which is advantageous in times of disaster

[MDM07]. MANET network devices self-organize into a multi-hop topology so packets can be relayed from one device to another across multiple nodes until they reach their destination [PPS08]. The challenge in creating the topology is to discover neighboring nodes close enough in proximity to maintain connectivity [PPS08]. One type of MANET of growing importance to the military is the wireless sensor network (WSN).

## 2.2 Wireless Sensor Networks

### 2.2.1 Overview

Wireless sensor networks are typically ad hoc networks consisting of sensor devices limited in their transmission power and life span due to their fixed battery capacity [KMB07]. Military interest in wireless sensor networks includes disposable networks of sensors that can detect and report an event of interest, such as troop or equipment movements [PST02, AMC07].

The number of nodes in a sensor network can be orders of magnitude larger than in an ad hoc network, sensor networks are typically densely populated, sensor nodes are limited in capability, sensor nodes are prone to failure due to harsh operating environments and limited battery life, and sensor network topology changes frequently due to failures or the movement of sensor nodes [WAR06].

Most deployed wireless sensor networks measure scalar physical data such as temperature, humidity, pressure or the location of objects. These functions require relatively low bandwidth and are tolerant of network delays or congestion [AMC07]. However, as research extends the capabilities of wireless sensor networks, the potential uses of these networks have expanded to include capturing video, still photos, or audio clips [AMC07]. The data throughput requirement is much higher for these applications.

### 2.2.2 Wireless Sensor Network Nodes

A typical wireless sensor node is very limited in resources such as battery power, processing power, and memory [CAA08]. Figure 1 shows a wireless sensor node, the Crossbow IRIS (slightly larger than its actual size). To give a sense of the scale of the device, the bottom of the device houses two standard AA batteries. This device has connectors for an interface expansion board and an external antenna connector. Due to the limited capacity of the batteries, the transmission range and data rate capacity of the node is also limited.

Figure 1. Crossbow IRIS Wireless Sensor Node [Cro08]

Table 1 shows the power consumption of the Crossbow IRIS in four different states. The highest power consumption occurs when the radio transmitter is transmitting at 3 dBm. When the node is not active and can be put to sleep, the power consumption drops by over three orders of magnitude to 8 microamps. With the dual AA battery pack supplying 2.85 Amp-hours of power [Ene08], a node that continuously transmits can operate for less than 7 days, whereas a node in its sleep state can last up to a year [Cro08]. This shows the importance of power conservation in these devices and the impact putting a node to sleep can have on the lifespan of

the node. Because the wireless transmitter is the largest consumer of energy in the node, great effort is taken to minimize the communications overhead and make all communications as efficient as possible [PST02].

Table 1. Power Consumption of Crossbow IRIS [Cro08]

| State | Current Draw |
|---|---|
| TX @ 3 dBm | 17 mA |
| RX | 16 mA |
| CPU active | 8 mA |
| Sleep | 8 μA |

Wireless Sensor Node devices also have limited processing power and memory. Table 2 shows the capabilities of two typical wireless sensor nodes. While a desktop computer may have a processor running at 3.4 GHz with 3 GB of memory, the Crossbow IRIS runs at 16 MHz and has 128 KB of non-volatile flash program memory.

Table 2. CPU and Memory Specifications of Crossbow Devices [Cro08]

| Device | CPU | Memory |
|---|---|---|
| Mica | 8 MHz | 128 KB |
| IRIS | 16 MHz | 128 KB |

### 2.2.3 Wireless Sensor Network Architecture

Figure 2 shows a typical wireless sensor network architecture. To enable WSN communications beyond the wireless sensor network, a gateway node is required. The gateway node sits at the boundary and differs from the other nodes in that it typically has a larger battery, increased communications range, and the capability to communicate with other networks [PST02]. The gateway is the only device in the wireless sensor network with an IP address [Cro08]. Inside the wireless sensor network field, layer two MAC addressing is used to forward data frames [AMC07].

Figure 2 shows a small WSN field organized into a cluster. Spatial clustering, which is described in more detail in Section 2.8.2, breaks up large networks into smaller, more manageable groups. Each cluster has a cluster leader, which is like any other node in the network, but with the added responsibility of tracking cluster membership.

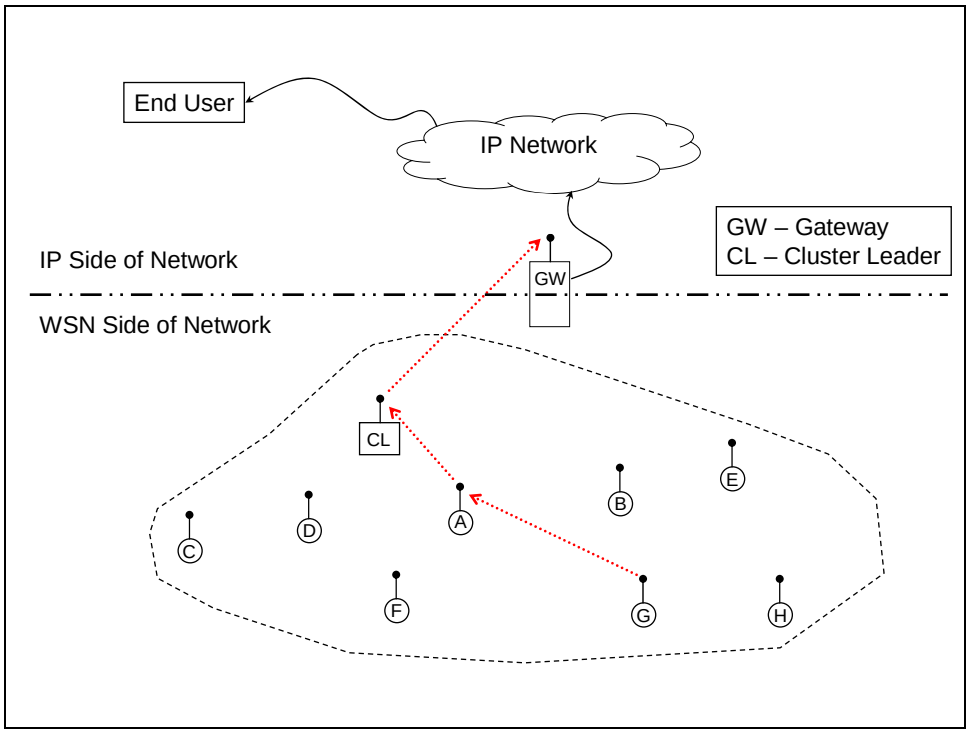


Figure 2. Wireless Sensor Network Architecture

Figure 2 illustrates how a message flows from a sensor to an end user. Sensor G has something to report to the end user, such as temperature or perhaps the movement of a large metallic object nearby. Sensor G determines the best path to the gateway and in this case, sends its message via node A to the cluster leader. The cluster leader forwards the message to the network gateway, which forwards the message across the IP-based network to the end user.

Since the nodes in a wireless sensor network are power limited, one way to extend the life of a WSN is to use a nearby communications asset to relay messages across the network to the end user. For example, a gateway node required to communicate directly with a low earth

orbit satellite would quickly consume its battery in radio transmissions to the satellite due to the distance involved. If the gateway node can use an intermediary, such as an Unmanned Aerial Vehicle (UAV), the gateway would use power and still achieve the required communications path since the distance to the UAV is much shorter.

## 2.3 Unmanned Aerial Vehicles

The DoD defines a UAV as:

> A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload. Ballistic or semiballistic vehicles, cruise missiles, and artillery projectiles are not considered unmanned aerial vehicles. [DoD08]

UAVs were first used in military operations by the United States military during the Vietnam War [USA05]. More recently, UAVs were used in Operation ALLIED FORCE for bomb damage assessment while the aircraft performing the attack were still in the immediate area and available to strike again as needed [HaH03].

### 2.3.1 Operational Characteristics

Table 3 shows the operational limitations and characteristics of UAVs used by the United States Air Force (USAF) that could bridge the gap between a WSN and satellites. These systems are designed with an overarching mission in mind and their characteristics are tailored to meet those primary mission requirements. However, all of these UAVs have useful communications that could create a network that spans an area much larger than a single WSN field.

Table 3. Operational Characteristics of UAVs [Gen 08, NoG07, USA08]

| Type Model Series | Name | Primary Function | Weight (lbs) | Max Speed (mph) | Range (mi) | Ceiling (1,000 ft) | Endurance (hrs) |
|---|---|---|---|---|---|---|---|
| MQ-1 | Predator | Armed Recon | 1,130 | 135 | 454 | 25 | 24 |
| MQ-9 | Reaper | Armed Recon | 4,900 | 230 | 3,682 | 50 | 30+ |
| RQ-4 | Global Hawk | Reconnaissance | 32,250 | 357 | 14,157 | 60+ | 36 |

### 2.3.2 Expansion of UAVs

Since Operation ENDURING FREEDOM and Operation IRAQI FREEDOM, research and acquisition dollars spent on UAVs has experienced huge growth. The Teal Group projects that the United States will spend over $2.4 billion on UAV research and acquisition in FY 2008 [TGC08]. UAV mission areas have expanded from short-term, short-range reconnaissance missions to: border patrol; detection of chemical, biological, or radiological materials; civilian search and rescue; environmental monitoring; weather data collection and more as their range, payload and sensor capabilities increase [USA05]. The United States Navy is considering a UAV system to provide Broad Area Maritime Surveillance [Jez08]. However, even with the vast expansion of UAV systems in use, Secretary of Defense Robert Gates stated the military is "too slow … to deliver more UAVs and other surveillance systems to Iraq and Afghanistan [Sch08]." Given the demand for these systems and the capabilities they provide, each must be used to its full capability.

### 2.3.3 Emerging Operational Concepts

The US military is looking for new ways to use UAVs to meet mission requirements. Instead of a traditional "stove-pipe" data link architecture, where all UAV sensor data is sent back to the US for analysis and dissemination, both the USAF and the Army are looking at ways

for UAVs to become a part of the regional data network so they can provide sensor data direct to the local users [Sch08]. This approach also has the advantage of providing other network services to users in the battlespace who cannot access the network through other means, such as isolated Provincial Reconstruction Teams, which are small teams typically located far from any US military infrastructure [USA05]. The US Army is working on a handheld device that would allow soldiers to access classified and unclassified network data, which could eventually integrate into an expanded network provided by UAVs [Ian08]. This use of UAVs to expand network connectivity provides more opportunities for WSNs to use UAVs to provide sensor data to end users.

### 2.4 Multicast Communications

Efficiently routing messages through a network is another way to increase the endurance of WSN nodes, by reducing or eliminating redundant messages in the network. Multicasting forwards a single message to select multiple recipients in different locations [SaM00]. Multicast differs from broadcasting in that it does not automatically forward a message to all connected nodes, but only to addressed recipients. The advantage of multicasting is that a message intended for multiple users does not require a unique transmission from the source for each destination. Without multicasting, the source has to send a copy of the message to each recipient individually. With multicasting, the source sends one copy to the closest network node who, in turn, sends a copy to the addressed nodes they are connected to. This decreases the bandwidth required to send the message and the time to send the message as shown in Figure 3. On the left of the figure, the source must send out seven copies of the message; one for each destination in the network. On the right, the network with multicasting sends only one copy of the message,

but the message propagates through the network, and is transmitted a total seven times; the intermediate routers take the copy they receive and transmit it to their subordinate nodes. The key differences are that the source sends only one copy instead of seven and node A receives only one message and transmits two, whereas node 1 receives seven messages and transmits six. Nodes B and C realize similar savings over nodes 2 and 3. Figure 3 is simplified slightly; it does not show any nodes not addressed in the message.

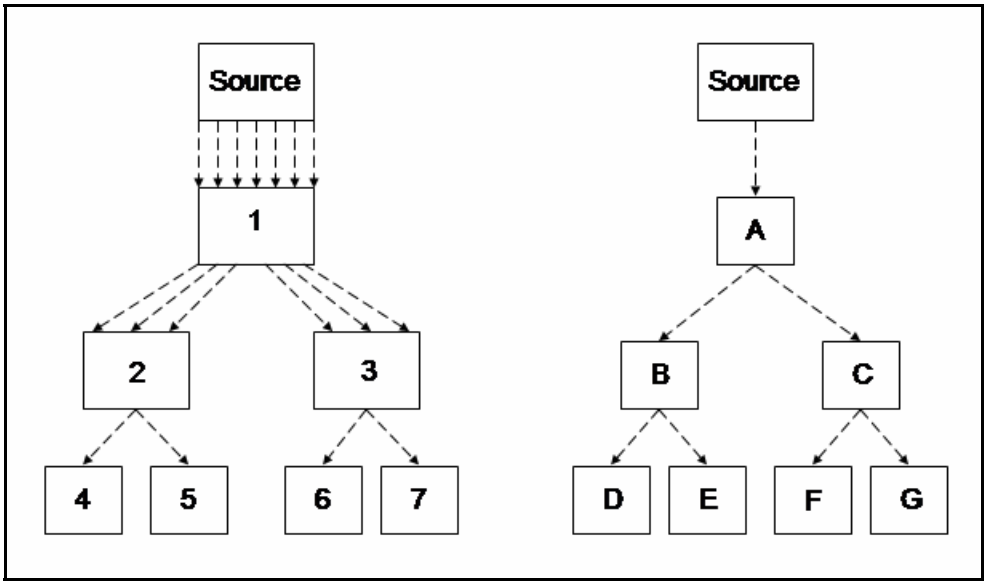


Figure 3. Comparison of Unicast to Multicast Communication

### 2.5 Internet Group Management Protocol (IGMP)

The basic building block of multicasting is IGMP [Ste94]. IGMP-capable routers provide multicast services by keeping a table of each group and its members. When a router receives a packet destined for a group, the router uses the IGMP table to forward the packet across multiple links to all members in the group, except for the link the packet arrived on.

### 2.5.1 IGMP Operation

There are two types of IGMP messages, and both are 8 bytes in length. The Type 1 IGMP message is the IGMP query sent by the router to the host. A Type 2 message is a report sent from the host to the router.

When a host joins an IGMP group, it sends a Type 2 report to the router containing the group address it wishes to join and its own IP address [Ste94]. This report is sent on the physical interface that the host wishes to receive the group messages on. To reduce the risk of the router not receiving the first request, a second request is sent after a random interval of time elapses, between 0 and 10 seconds [Ste94].

At unspecified regular intervals, the router sends Type 1 queries to all hosts in the multicast group [Ste94]. The hosts must respond with a Type 2 message to remain in the group.

### 2.5.2 Limitation of IGMP

While IGMP management of groups can save power by reducing or eliminating redundant transmissions, it is not intended to be a security mechanism. A stealthy node in promiscuous mode can still receive group transmissions not intended for it, while remaining undetected. For this reason, additional security measures must be used to secure the communications.

## 2.6 Gothic Group Membership Authentication

Knowing who is in the multicast network, allowing current users to leave, and providing means for new users to join are fundamental requirements for a secure multicast group. However, in a MANET with users entering and leaving the group, authenticating who is in the group is a challenge. While IGMP can track who is in the group, IGMP permits any user who

wishes to join a group to do so without authentication. A truly secure group must authenticate the members before they are permitted to join. Gothic's Group Access Control (GAC) [JuA02] is one approach to provide authenticated group membership.

### 2.6.1 Gothic Operation

Efficiently allowing users to join and to leave secure multicast groups is the objective of Gothic, a group access control architecture developed at the Georgia Institute of Technology [JuA02]. There are two key parts to Gothic: the GAC and the Group Key Management (GKM) system, which combine to form the Group Access Control Aware – Group Key Management (GACA-GKM) system. The GAC controls group membership and is described here. The GKM portion of Gothic is described in Section 2.8.1.

The GAC interfaces with the network routers to ensure users do not have access before they are validated or after they depart the network. Before a node is allowed to become a member of a group, GAC authenticates the user to ensure the user is authorized to join. If authentication is successful, the node joins the group.

### 2.6.2 Limitation

While Gothic provides authenticated group membership, it is not a complete, standalone security solution. It is easy for an eavesdropper to spoof a MAC address and receive all data intended for the group [Per08]. Gothic is also susceptible to the stealth eavesdropper attack described in Section 2.5.2.

## 2.7 Encryption

Encrypting communications for military operations is important to operational success. There are four desirable properties of secure communications: confidentiality, authentication,

non-repudiation, and availability [KuR05]. Confidentiality ensures only the sender and intended receiver can access the transmitted message. Authentication validates the sender's and receiver's identities. Availability keeps the system accessible to authorized users. Non-repudiation ensures the sender of a message cannot deny sending it and assures the receiver that the message has not been modified since it was sent [KuR05].

### 2.7.1 Encryption Techniques

Encryption provides message confidentiality and is the basis for non-repudiation [KuR05]. There are two types of cryptographic systems: symmetric and asymmetric [MOV96]. In a symmetric encryption system, a transmitter and receiver(s) share a common private key that is used to encrypt and decrypt the message. While the key is symmetric, the network does not have to be. That is, one message can be received and decrypted by multiple receivers as all participants use the same key.

In an asymmetric encryption system, each user has a pair of keys. One is used to encrypt the message, known as the public key. The other is a private key that is used to decrypt the message. The transmitter must have the recipient's public key for the message to be successfully decrypted by the recipient's private key. For example, if Bob wants send a secure message to Alice, Bob uses Alice's public key to encrypt his message to her. Once Alice receives the message, she uses her private key to decrypt and read the message. Once Bob encrypted the message with Alice's key, only recipients who have access to Alice's private key can decrypt the message. Even Bob cannot decrypt the message that he encrypted with Alice's public key. If a third person, Trudy, intercepts the message, she cannot decrypt it without Alice's private key. As long as Alice keeps her private key from anyone else, messages intended for her are secure. In asymmetric key systems, multiple transmitters can use the one public key to send messages to

a single recipient, but since the key pair is tied to a specific user, the keys cannot be used to communicate with anyone else in the network. Because the keys in an asymmetric key system are tied to a specific user, the asymmetric key system also provides authentication control [KuR05, MOV96]. However, since it is computationally intense and because multiple keys are required, asymmetric encryption is not used in WSNs due to the limited CPU and memory capacity of WSN nodes.

### 2.7.2 Key Distribution Techniques

The distribution of keys within a network is an important aspect of the network's overall security. In a symmetric system, the key is secret and must be sent to the transmitter and receiver by secure means. This could be by encryption with another key known as a Key Encryption Key (KEK) or by another means entirely, such as physical delivery by a courier [MOV96].

Embedding keys into the memory of the WSN nodes is not practical since every key stored in memory occupies space that could be used for program code or sensor related data. Instead of embedding keys into WSN nodes, there are rekeying protocols which send new keys to WSN nodes. Four methods are pair-wise keying, group keying, hierarchical trees, and Secure Lock [BaB02, Kru98]. It should be noted in this document, the terms rekeying protocol and keying protocol refer to the same protocol.

#### 2.7.2.1 Pair-wise Keying

In pair-wise keying, a communications session is created between the two participants to exchange the necessary keys. Even in a multicast group, each individual must establish a unicast session to be rekeyed. This technique has the advantage of being the most secure since the

compromise of a key only compromises one node. The drawback is the large number of keys required, which consume limited memory and make key management more difficult. For a network of N nodes, each node must store N keys.

#### 2.7.2.2 Group Keying

In a group keying network, all of the members of a group share a common key. While this reduces the complexity of the system, it increases risk. If the key is compromised, all of the nodes in the group are compromised. A group keying network provides a tradeoff between a single key shared by all users and pair-wise keying.

#### 2.7.2.3 Hierarchical Keying

Hierarchical keying establishes a hierarchical tree of keys [SuR07]. When multiple nodes are addressed in a message, the lowest common key held by all nodes is used to send the message. Figure 4 shows a hierarchical tree for eight nodes, which are assigned positions at the bottom of the tree. The group key manager, not shown in Figure 4, sends each node keys using their assigned KEK. The tree is formed by sending each node the next higher level key using that node's KEK. Then, the next higher level key is sent using the previous level's key as the KEK. In Figure 4, nodes 1 and 2 receive their common Level C key through individually encrypted messages using their unique KEKs. Nodes 3 and 4 receive similar transmissions, but they receive their common Level C key. Next, nodes 1 through 4 receive the common Level B key through encrypted transmissions using the Level C keys as the KEKs. This process continues up the tree until all KEKs are established through Level A. Finally, to establish a common group key, a Session Encryption Key (SEK) is sent to the entire tree using the Level A key as the KEK.

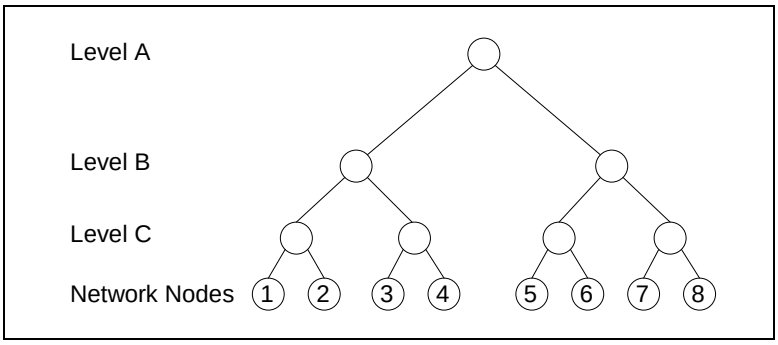


Figure 4. Hierarchical Tree

The advantage of hierarchical keying is the reduction in the total number of keys held by each node. For a network size of N that uses the structure shown in Figure 3, the number of keys for each node is

$$\text{Number of keys} = \log_2(N) + 2 \quad (1)$$

where the two additional keys are the node's unique KEK and the entire group's SEK.

A disadvantage of hierarchical keying is accommodating nodes that join the network. If a node attempts to join a hierarchical tree that is full, the new node is either denied entry or an additional level must be added to the tree. Adding an additional level to the tree doubles the total capacity of the tree at a cost of additional key transmissions to existing nodes, with one additional key added to each level.

#### 2.7.2.4 Secure Lock

In Secure Lock [ChC89], each node is assigned a unique, pair-wise prime identification (ID) number. When a Secure Lock is generated, the ID numbers of the nodes to be included in the message are used to generate a unique numerical solution based on the Chinese Remainder Theorem. When each node divides the unique numerical solution by their assigned ID number, they all obtain the same remainder. This remainder is the message intended for all of the

included nodes. A user that is not intended to receive the message will still receive the Secure Locked message; however, when this user divides the received message by its assigned ID number, the user obtains the wrong result.

To illustrate this, consider an example using 8-bit numbers. An 8-bit message of $A5_{16}$ is to be sent to nodes $131_{10}$, $137_{10}$, and $139_{10}$. Following the process described in Appendix A, the Secure Lock of this message is $26104C_{16}$ ($2494540_{10}$). Node 131 ($83_{16}$)has the remainder of $38_{10}$ ($26_{16}$). A bit-wise XOR of $26_{16}$ and the node's ID of $83_{16}$ gives the correct message of $A5_{16}$. The process for nodes 137 and 139 is the same and yields the same result. For a node that was not included in the Secure Lock generation, such as node 149 ($95_{16}$), the remainder for 2494540 / 149 is $131_{10}$ ($83_{16}$). Performing a bit-wise XOR between $83_{16}$ and the node ID of $95_{16}$ yields $16_{16}$, which is not the correct solution.

Secure Lock does not scale well as the key size or number of recipients increase. Figure 5 shows the linear relationship between the number of nodes included and the size of the Secure Lock message for six different key sizes. The linear relationship between the message size and the key size can be seen as well. At 5,000 nodes the message size doubles between 32 and 64-bit keys. It doubles again between 64 and 128-bit keys. This linear relationship holds for all of the network sizes shown in Figure 5.

An advantage of Secure Lock is the ability to have one broadcast or multicasted message "unlocked" by selected recipients. However, Secure Lock does not scale well. In the example above, the message grew from one byte to three; a linear relationship. However, the time and memory needed at the key generation system to generate the Secure Lock grows exponentially as more recipients are included [AnM08].

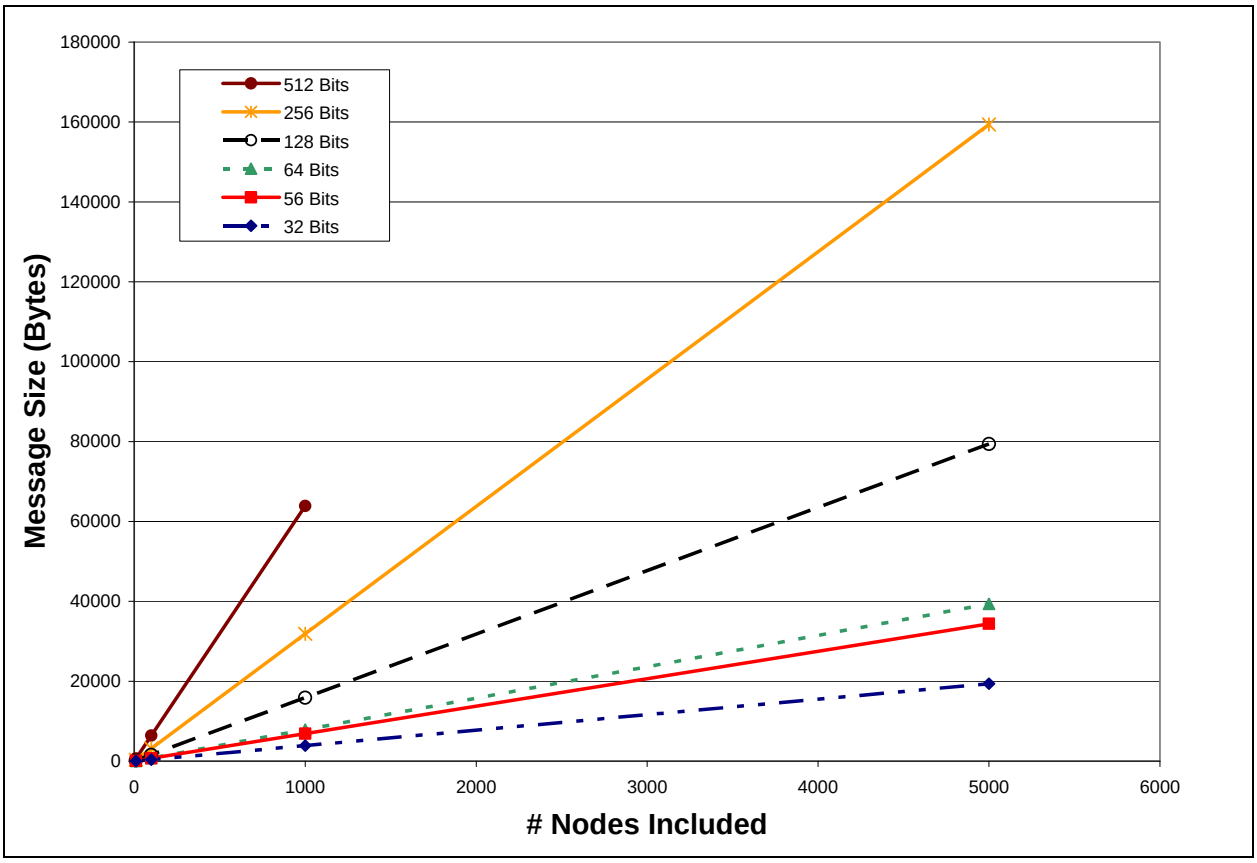


Figure 5. Secure Lock Message Size [AnM08]

### 2.8 Key Management in Multicast Groups

The objectives of a multicast security infrastructure are to preserve authentication and confidentiality for all group communications so only registered senders can send packets to the group and only they can read packets received from the group [MRR99]. To achieve these objectives, a key management system must rekey users in the proper manner and at appropriate times to ensure the security of the system.

Group Key Management focuses on the dynamic group key problem. Since IGMP is open, it is common for group key management protocols to not use IGMP for security. When a member joins or leaves the group, the group key manager usually assumes the member continues to have access before or after being allowed membership [JuA02]. To ensure the security of the

group, it is common to rekey the group to ensure that new members cannot decrypt group messages sent before the new member joined and to ensure a departing member cannot continue to decrypt messages after leaving the group [JuA02].

### 2.8.1 Gothic Group Key Management

Gothic's GKM is a part of the Group Access Control Aware – Group Key Management system (GACA-GKM). By using the authorized membership information in the GAC, the key manager can make efficient decisions about when group rekeying is required. With perfect group access control provided by the routers, only authorized nodes can receive the group's communications, a new member joining the group does not require the group to rekey since the new member could not receive the groups communications before it joined. Likewise, after leaving the group, the departed member can no longer receive group messages and a rekey is not required [JuA02]. This ignores the possibility of an eavesdropping node, but the reduction of rekey operations improves the scalability of the network.

Gothic uses these three rules to determine if a rekey is required [JuA02]:

1. If a host joins a multicast session and has previously been a member of the same multicast session, a rekey must occur.
2. If a host leaves a multicast session and will remain in the same multicast tree session, a rekey must occur.
3. Otherwise, there is no need to rekey.

In a wireless setting, suppose two nodes are within range of a wireless access point. One is an authorized group member and second is not. However, they both are able receive wireless network traffic. If the second user eventually becomes a group member, the group must rekey due to rule 1, above. Since the second node could have stored past traffic and now has access to

the key used to secure those messages, the second node could decrypt the old messages that were sent before it became a member. A similar situation occurs when a member leaves, but still is able to receive group communications (rule 2, above).

In a wireless environment, there is no way to prevent unauthorized nodes from receiving network traffic. Therefore, the GACA-GKM cannot ensure the security of the group through limiting access to the encrypted communications as can be done with IGMP-capable wired routers. However, GACA-GKM still provides a benefit to the group since it tracks the cluster membership of each group member. Based on this information, the GACA-GKM reduces rekey operations when an authorized member from one cluster moves to another cluster. In this case, a rekey is not required for either cluster.

### 2.8.2 Spatial Clustering

Spatial clustering reduces the number of rekey operations by dividing a large network into smaller clusters along geographic boundaries [BaB02]. These clusters are bounded by size, do not overlap [BaB02], and nodes are not allowed to be a member of more than one cluster. Each cluster is independent and has its own group key. If a member of the group leaves or if a new member joins, only that cluster requires a rekey.

Spatial clustering does not require any prior deployment knowledge of the network. The spatial clustering protocol determines clusters by fixing an integer value for $k$. Stable clusters have between $k$ and $2k$-1 members [BaB02]. To form clusters, the protocol traverses the multicast tree from the leaves to the root, assigning members to clusters as it traverses the multicast tree [HRB08]. If new nodes join the network, a cluster may exceed $2k$-1 members. In that case, the cluster splits into two stable clusters of size $k$ or larger. After the protocol

stabilizes across the entire network, it is still possible that one undersized, unstable cluster will remain near the root.

Each group is assigned a cluster leader. The cluster leader maintains information about cluster membership, but is not involved in any of the security aspects of allowing or authenticating requests for membership into the multicast group [BaB02]. Spatial clustering relies on an authentication service outside the protocol to authorize and authenticate the multicast group members.

Since spatial clustering does not require any pre-deployment knowledge of the network, is based on spatial relationships, and reduces the impact of a rekey operation, it is a promising protocol for a wireless sensor networks deployed in a random fashion. Spatial clustering should conserve battery power in such a WSN.

### 2.8.3 Iolus

Iolus divides large networks into smaller clusters, like spatial clustering [BaB02], but uses administrative boundaries and does not limit the number of members in a cluster. Iolus also uses predetermined nodes as Group Security Agents (GSAs), instead of the dynamically assigned cluster leaders of spatial clustering [BaB02]. Iolus is well suited for networks or portions of networks that are statically assigned and do not change very often.

## 2.9 The Hubenko Architecture

The Hubenko architecture is designed to provide secure communications to large networks of mobile users. It combines elements of Iolus, Spatial Clustering, and Gothic and applies them to a network of low earth orbit satellites providing network connectivity.

At the top level of the Hubenko architecture, the low earth orbit satellites, Iolus divides the network into smaller pieces. Since the satellites follow pre-determined orbits, it is feasible to pre-assign clusters and the assign the role of GSA to the satellites [HRB08].

Each spot beam of each satellite constitutes a spatial cluster. Given the dynamic nature of the membership of each cluster, spatial clustering is preferable, since it can adapt to changes in the network as nodes join or leave.

However, high mobility users that move from one cluster to another can trigger a large number of rekey operations. To eliminate unnecessary rekey operations, Gothic's GACA-GKM controls membership and determines if a rekey is needed [HRB08].

An example of the Hubenko architecture is shown in Figure 6. Satellite spot beams are labeled A thru H. These constitute clusters, each with their own group key. The numbers 1 through 16 identify unique users in the network. The satellites form their own cluster with links labeled by the common group key, "V". Gothic's GACA-GKM at each satellite controls multicast group membership and determines when group rekeys are required, such as when a user in cluster B moves to cluster C. In this example, since the user is trusted before, during, and after its move from cluster B to cluster C, a rekey operation is not required.

### 2.9.1 Experimental Results

The Hubenko architecture is compared to a flat baseline architecture and to the spatial clustering architecture in five different scenarios of varying user mobility with one hundred iterations for each scenario [HRB08]. Table 4 shows the user types for each iteration in the five scenarios, with users that increase in mobility from lowest to highest: stationary, ground, sea, and air. The scenarios start with stationary nodes and increase in mobility. The last scenario has the most high-mobility users.

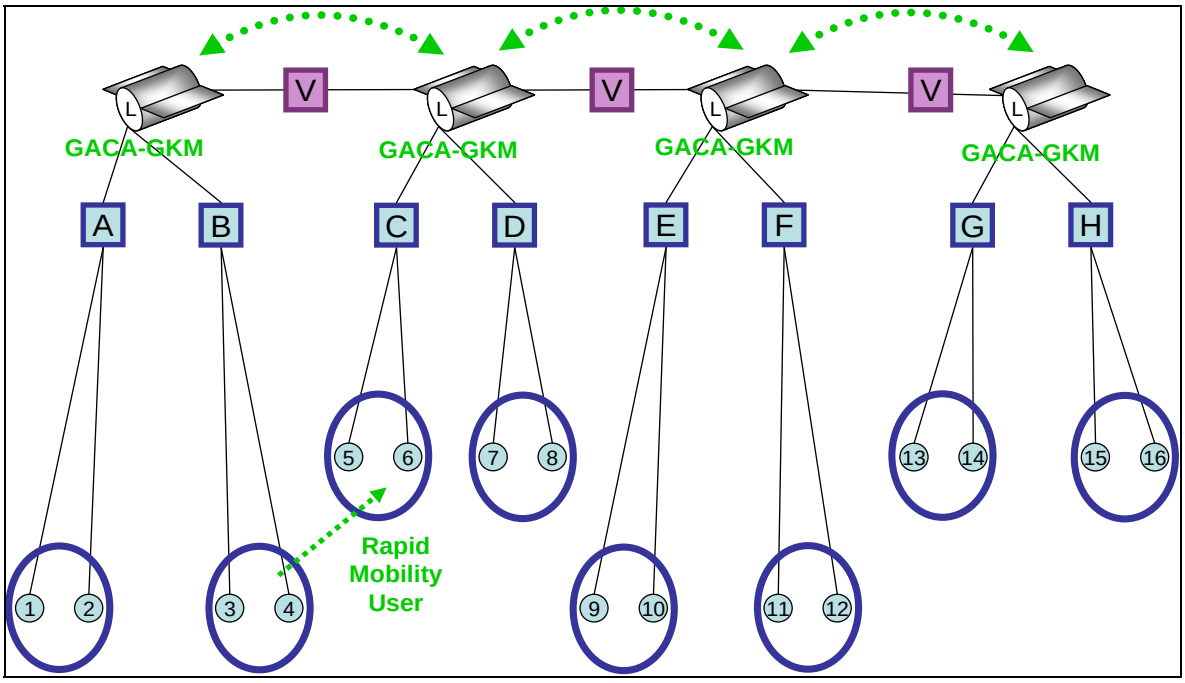


Figure 6. The Hubenko Architecture [Hub08]

Table 4. Rates of Mobility Sets [HRB08]

| Iteration | User Types |
|---|---|
| 1 through 100 | All Stationary |
| 101 through 200 | ~1/2 Stationary, 1/2 Ground |
| 201 through 300 | ~1/3 Stationary, Ground, Sea |
| 301 through 400 | ~1/4 Stationary, Ground, Sea, Air |
| 401 through 500 | ~1/5 Stationary, Ground, Sea; 2/5 Air |

Figure 7 has the results of the five scenarios shown in Table 4. For the first scenario, all users are stationary and the Hubenko architecture does not provide any benefit over spatial clustering alone. The flat architecture shows the highest number of rekeys since all users belong to one group, sharing the same key. As the mobility of the users increases the number of rekey operations increases rapidly in both the flat baseline and spatial clustering architectures, while the Hubenko architecture increases linearly. While the Hubenko Architecture has similar performance for each scenario, other rekeying techniques show dramatic increases in the amount rekeying operations they perform [HRB08].

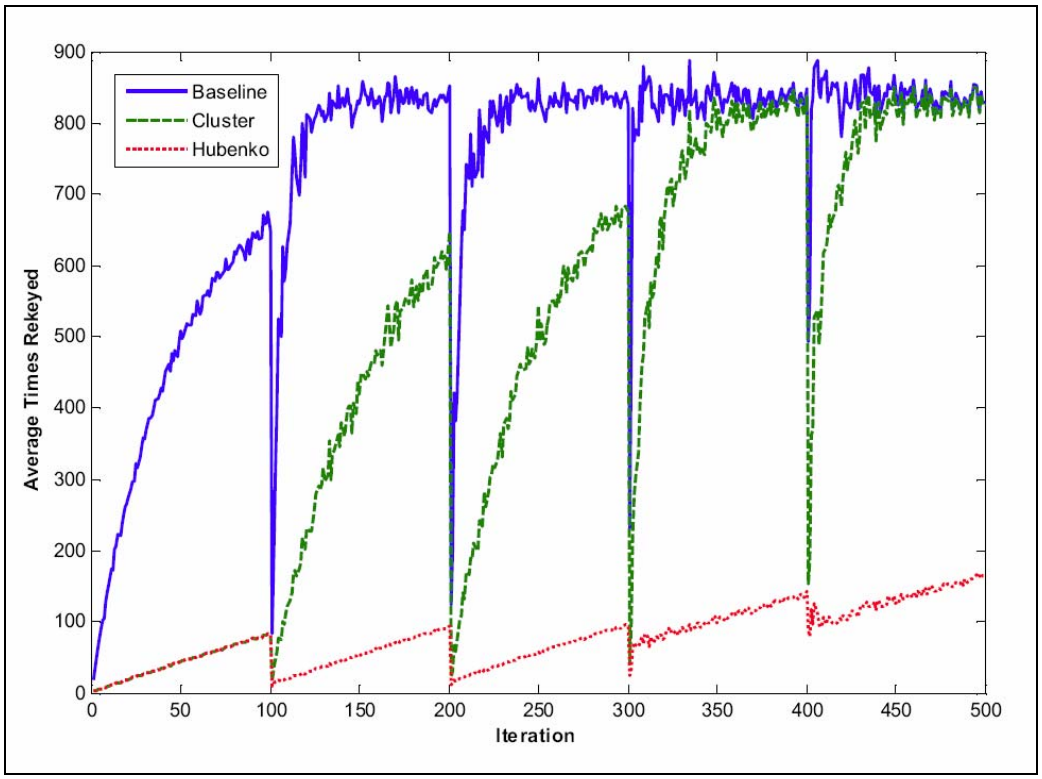


Figure 7. Average User Re-Key Count Comparison [Hub08]

A WSN that does not experience high mobility will still experience changes in network topology as nodes become inoperative or as new nodes join. The Hubenko architecture is expected to benefit WSNs similar to the second scenario, iterations 101 through 200.

## 2.10 Hubenko Architecture Applied to Swarms of Autonomous UAVs

The Hubenko architecture has been studied in the context of swarms of autonomous UAVs [Phi08]. Improved scalability of the Hubenko architecture over spatial clustering and a flat architecture was demonstrated [Phi08].

### 2.10.1 Research Approach

The Hubenko architecture is adapted to a network of UAVs, shown in Figure 8. At the top of the network is a Global Hawk UAV, which acts as the overall security controller for the entire network. In the middle tier are mid-sized UAVs which normally operate at 10,000 feet.

The lowest tier is comprised of micro UAVs which operate at an altitude of hundreds of feet. Instead of being limited by the satellite beam size, the size of the clusters is limited by the radio footprint of the middle tier UAVs. Each cluster has its own key, identified by "CK*n*", and the dashed lines between the Global Hawk and the cluster leaders show the GACA-GKM relationships [Phi08].

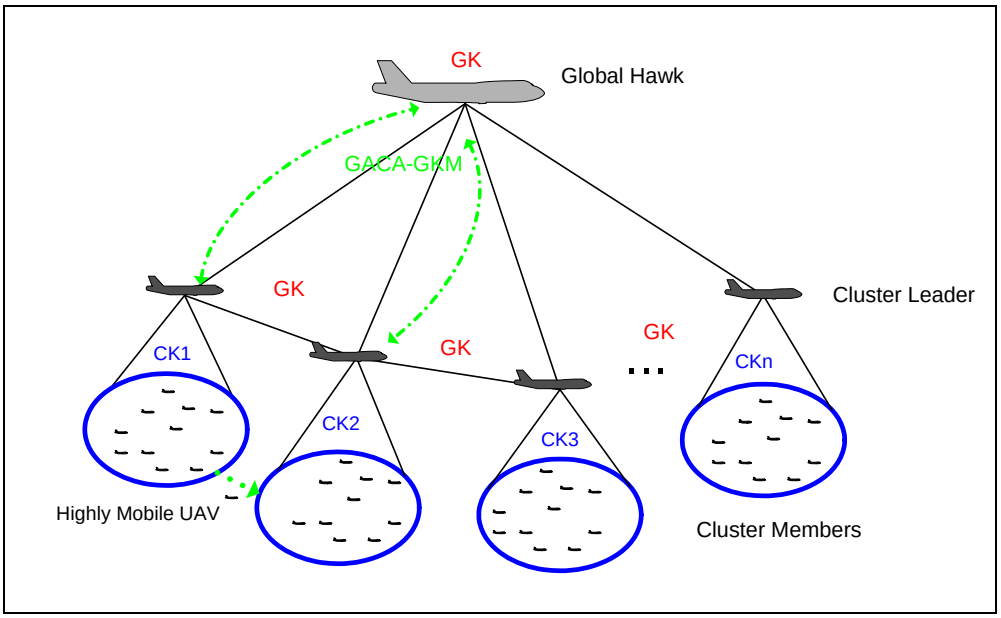


Figure 8. Hubenko Architecture Applied to UAV Swarm [Phi08]

When the multicast group first forms, the Global Hawk assigns cluster leaders. The ideal cluster leaders are the mid sized UAVs, which are closer to the swarms of smaller UAVs, have longer endurance than the micro UAVs and have more powerful radios [Phi08].

Tables 5 and 6 show the factors used for two experimental scenarios. Besides the differences shown in the factors below, scenario 1 has a duration of 2 hours, while scenario 2 has a duration of 12 hours. The experimental design is full factorial with 1,968 simulation runs total [Phi08].

Table 5. Factor Levels Scenario 1 [Phi08]

| Factor | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Swarm Size | 40 | 100 | 200 | 500 |
| Swarm Mobility | 25% | 75% | | |
| Group Join Rate | 15% | 30% | | |
| Group Departure Rate | 25% | 75% | | |
| Security Architecture | Baseline | Clustered | Hubenko | |

Table 6. Factor Levels Scenario 2 [Phi08]

| Factor | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 |
|---|---|---|---|---|---|
| Swarm Size | 40 | 100 | 200 | 500 | 1000 |
| Swarm Mobility | 25% | 50% | 75% | 90% | |
| Security Architecture | Baseline | Clustered | Hubenko | | |

The swarm size indicates how many UAVs are in the network. The swarm mobility is the percentage of the swarm that is highly mobile, where highly mobile is defined as a UAV that moves beyond a 5 km radius. The group join rate is the percentage of the total simulation time for the entire swarm to join the multicast group. The group departure rate is the percentage of the swarm that departs before the end of the simulation. The security architecture varies between flat, where all nodes use the same key, spatial clustering, and the Hubenko architecture [Phi08].

### 2.10.2 Results

Philips found that the Hubenko architecture consistently improves performance as shown in Table 7. In the four areas considered, the Hubenko architecture reduced the number of rekey operations, the bandwidth used, and battery power compared to the baseline and cluster architectures for the mobile swarms of UAVs [Phi08]. The range of values includes the results for all the factors shown above.

Table 7. Reductions Observed [Phi08]

| | Hubenko compared to | |
|---|---|---|
| | **Baseline** | **Clustering** |
| Total Keys Distributed | 58 - 88% | 55 - 95% |
| Rekeys per UAV | 60 - 88% | 59 - 95% |
| Bandwidth Used to Rekey | 73 - 88% | 55 - 85% |
| Battery Power Used to Rekey | 17 - 59% | 54 - 85% |

## 2.11 Summary

This chapter presents the fundamental concepts and recent research in the areas of UAVs, MANETs, wireless sensor networks, multicast communications, security of wireless communications, encryption techniques, key distribution techniques, and access control techniques. Current Hubenko architecture research results are also presented.

# III. Methodology

This chapter presents the experimental research methodology. Section 3.1 presents the problem definition and approach. Section 3.2 covers the system boundaries, while Section 3.3 lists the services provided by the system. Section 3.4 explains the system workload. Section 3.5 presents the performance metrics, followed by the system parameters in Section 3.6. Section 3.7 explains the factors used. Section 3.8 presents the evaluation technique. Section 3.9 covers the experimental design. Section 3.10 summarizes the chapter.

## 3.1 Problem Definition

### 3.1.1 Goals and Hypothesis

For WSNs to be a useful and reliable military resource, they must have secure communications. However, the cost of the security must be minimized, especially for the capability and resource-constrained nodes of a WSN. Previous research on the Hubenko architecture indicates it is reasonable to expect the realized benefits will also apply to WSNs, even though WSNs experience fewer joins, leaves, and re-joins than autonomous UAV networks do [Phi08]. Prior research also concluded the Hubenko architecture could benefit from a different rekeying protocol [Hub08, Phi08].

This research evaluates the performance of the Hubenko architecture using three different rekeying protocols within the Hubenko architecture as applied to a WSN.

The goals of this research are to:

- Improve the efficiency of rekeying operations within a WSN.
- Investigate whether using hierarchical rekeying or Secure Lock provides measurable benefits.

- Evaluate the impact of these rekeying protocols on the resource-constrained nodes of a WSN.
- Determine which rekeying protocol should be applied given particular network parameters, such as WSN size.

It is expected that the pair-wise keying approach used in the Hubenko architecture is a large source of inefficiency and that an alternative method will greatly reduce rekey overhead.

### 3.1.2 Approach

The Hubenko architecture reduces rekeying operations by reducing the number of users who have to be rekeyed and how often they are required to rekey through pair-wise keying. The effect of two alternative rekeying methods: hierarchical and Secure Lock is determined. Figure 9 is a conceptual view of the network. At the bottom left is the WSN, split into n clusters. Each cluster has its own SEK, provided by the GACA-GKM via a communications relay.

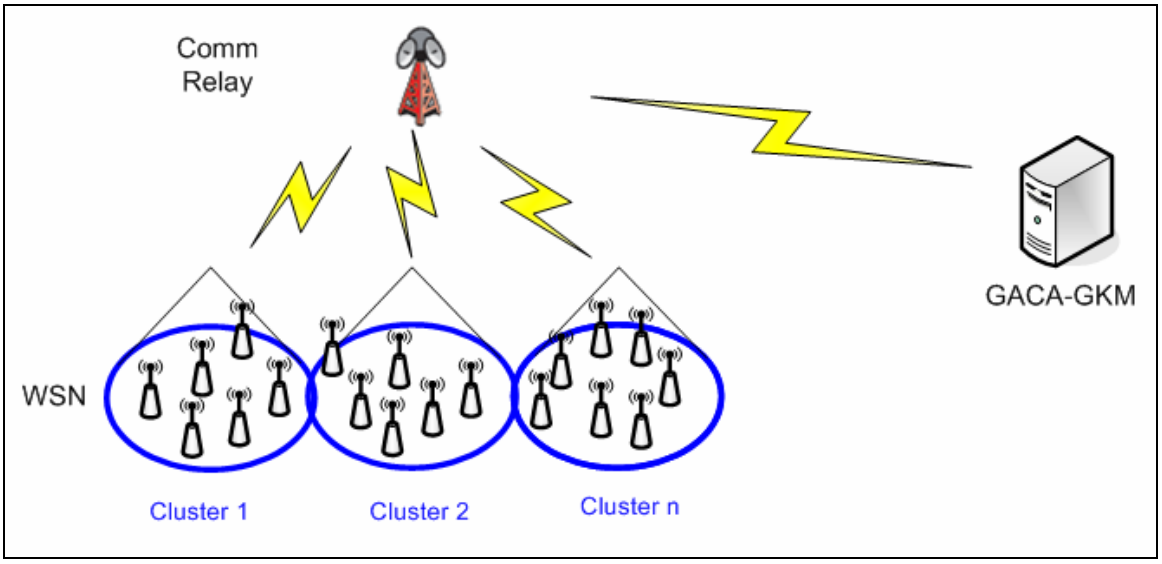


Figure 9. Conceptual View of Network

The communications relay is positioned within 300 meters of the WSN [Cro08] and extends the transmission range of the WSN to reach the GACA-GKM. This allows a

communication link via UAV to a satellite, enabling world-wide placement of the GACA-GKM and any external users to the network. More details about the communications relay and its link budget are in Appendix B.

### 3.2 System boundaries

The System Under Test (SUT), the Rekey Improved Hubenko Architecture, is a communication network that conforms to the Hubenko architecture and is comprised of the GKM and WSN as shown in Figure 10. Additional components within the SUT include the communications channels and the rekey protocol. The Component Under Test (CUT) is the rekey protocol: pair-wise, hierarchical or Secure Lock.

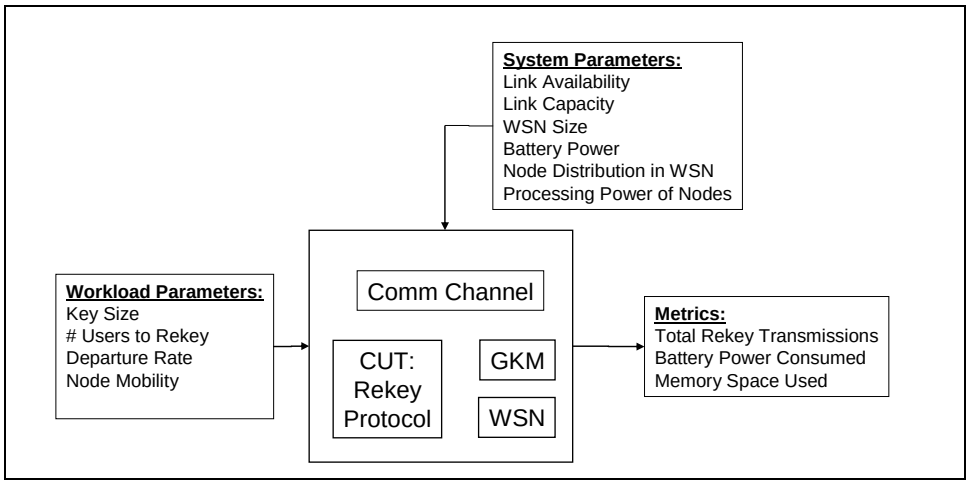


Figure 10. System Under Test: the Rekey Improved Hubenko Architecture

The system parameters include the battery power of the WSN nodes, the size of the WSN, the distribution of nodes within the WSN, the processing power of the nodes in the network, the link availability, and link capacity.

The workload parameters include the size of the key, the number of nodes in the network to rekey, the mobility rate, and the departure rate of nodes from the network.

The metrics include the total number of bits transmitted to perform a rekey within the network, the amount of battery power consumed within the network, and the amount of memory within the WSN nodes required to store rekey messages.

Since this research focuses on different rekeying protocols, a reliable communication channel is assumed and all messages sent in the network are received without error. The modeling of a network with retransmissions is left for future work.

### 3.3 System Services

The system provides secure network communications as well as a rekeying service for the entire WSN. One of the requirements of secure communications is the ability to securely provide new keys to authorized recipients to ensure continued network access of authorized users, while denying unauthorized users access to past, present or future network traffic.

### 3.4 Workload

The workload of the SUT consists of the rekeying messages passed between nodes. For this research effort, the SUT workload is the network traffic associated with a rekey operation, which is dependent on the rekeying protocol used, the size of the key used, the number of users to be rekeyed, the mobility rate of nodes within the network, and the rate at which nodes leave the network.

The rekey protocol can affect the SUT workload by changing the number of individual transmissions sent within the network. For example, pair-wise keying requires a separate communication from the key generation node to every other node in the network. On the other hand, a rekey message sent with Secure Lock requires only one rekey message for all members

of the group. The number of keys sent under the hierarchical protocol depends on the number of nodes in the network, which determines the size of the hierarchical tree.

The size of the key has a direct impact on the SUT workload. In addition to having more bits to transmit across the network, generating and using a longer key affects each node in the network by inducing a heavier workload.

Under any of the rekey protocols, a subset of the entire WSN can be rekeyed. As more users are required to be rekeyed, the workload increases.

Nodes may be re-assigned to new clusters within the network to achieve enhanced efficiency or some operational effect. As the nodes move from one cluster to another, they do not trigger rekey operations in their old or new cluster, but they must receive the new keys to communicate within their new cluster. This places additional workload on the system distinct from node departures.

Nodes in a WSN depart the network when their battery is depleted. Nodes can also be damaged, captured, or otherwise unavailable. To maintain the security of the network, a rekey operation ensures the trust of the remaining nodes within the network. It is assumed the system is aware of all departures. To track departures, cluster leaders are responsible for tracking cluster members as well as dormant nodes.

### 3.5 Performance Metrics

Performance metrics provide a means to measure the impact a particular rekeying protocol has. The metrics collected include the total number of bits required for rekey transmissions in the network, the battery power consumed, and the memory space used in the WSN nodes.

When a rekey operation is requested, the number of bits transmitted within the network distinguishes the impact of the different rekeying protocols. The number of bits transmitted is measured at the communications relay outside the WSN, shown in Figure 9.

Conserving energy in WSN nodes is important, since once the battery is consumed, the node is inoperative. Nodes in the middle of a WSN are at the most risk since they route much of the traffic through the network. By calculating the power consumed by each radio transmission during a rekey operation, power consumption is determined.

The memory in WSN nodes is also limited. By measuring the number of bits each protocol requires during a rekey operation, the memory space occupied can be determined. Comparing how each rekeying protocol uses memory is a useful metric in choosing the best rekeying protocol for use in a network.

### 3.6 System Parameters

System parameters affect the performance of the SUT. These include: link capacity, link availability, the size of the WSN, WSN node battery power, the distribution of nodes within the WSN, and the processing power of nodes.

The link capacity used for this research is 250 kb/s [Cro08]. Each link provides reliable communications.

The WSN node battery power matches the Crossbow IRIS, which has a pair of standard AA batteries providing 2800 mAh of power [Ene08].

The Crossbow IRIS processor runs at 16 MHz and has 128 KB of memory [Atm08, Cro08]. The key generation node (GACA-GKM) is a desktop PC with a 3.4 GHz processor and 2 GB of memory.

The WSN is split into ten clusters of equal capacity. The individual nodes are assigned to clusters according to a uniform distribution.

### 3.7 Factors

The factors herein provide insight into the impact of the different rekeying protocols without excessive or redundant effort. The factors include: rekeying protocol, key size, WSN size, mobility rate, and departure rate. Table 8 shows a summary of the factors and their associated levels.

Table 8. Factor Levels

| Factor | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Rekey Protocol | pair-wise | Secure Lock | hierarchical | |
| Key size | 128 bits | 256 bits | 512 bits | |
| WSN Size | 40 | 100 | 500 | 1000 |
| Mobility | 0% | 5% | | |
| Departure Rate | 25% | 75% | | |

The rekeying protocol factor has three levels: pair-wise, Secure Lock, and hierarchical. The pair-wise protocol is the baseline. Secure Lock and hierarchical provide two alternatives.

The key size has three levels: 128, 256, and 512 bits. The 256-bit key is the baseline and matches previous research [Phi08]. These key sizes are typical and give some variation to determine the impact of the key size on the system.

The size of the WSN has four levels: 40, 100, 500, and 1000 nodes. These levels represent a reasonable range of WSN sizes to assess how the rekeying protocols scale. The three smallest values mirror previous research [Phi08]. The largest value represents a large WSN.

The mobility rate varies between 0% and 5%. This factor specifies how many mobile nodes there are within the network. For the Hubenko architecture to have a benefit over

clustering alone, nodes must move within the network. Mobility does not mean the nodes have to physically move. As time goes on, it may be more efficient for a node to be re-assigned from one cluster to another if it is in close proximity to two or more clusters and the cluster it is assigned to experiences a large number of departures. In this simulation, the GACA-GKM tracks membership of all clusters and re-assigns nodes to new clusters. Each mobile node can move up to two times during the simulation, based on a random uniform distribution.

The departure rate has two levels: 25% and 75%. This factor represents how many nodes depart the network before the end of the simulation and matches previous research [Phi08]. The departure times are based on a random normal distribution with a mean of 75% of the simulation length and standard deviation of 10% of the simulation length. The normal distribution approximates the lifespan of a battery powered WSN node.

### 3.8 Evaluation Technique

The network is simulated using Matlab 2007a. The network is shown in Figure 9 and rekeyed based on the factors above.

Simulation is the only practical choice for this research. Even if a WSN of 1,000 nodes was available, configuring the network to perform as required, operating it, and collecting the data generated would be impractical.

The simulation is a discrete-time simulation. The simulation randomly assigns trigger events at the start of the simulation (such as joins and departures). Results are stored in matrices that are exported to Excel spreadsheet files for post-simulation analysis. The simulation has duration of 30 days, which is equivalent to systems currently fielded by the US Army [L3C04, L3C04b].

The simulation is validated by comparing the baseline results with the results of the previous research for the autonomous UAV network performed by Phillips [Phi08].

### 3.9 Experimental Design

The experiment uses a full factorial design. Given the factors listed above, 144 unique scenarios are required (3*3*4*2*2=144) to collect the data for each combination of factors.

A confidence level of 95% is used and each scenario is replicated ten times. This requires 1440 distinct runs of the simulation.

### 3.10 Summary

This chapter presents the methodology to evaluate three different rekeying protocols using the Hubenko architecture in a WSN. The performance of these protocols is evaluated by simulation and collects: total number of bits transmitted for rekeys, battery power consumed, and the memory space used at the WSN nodes. A full factorial experiment defines 144 unique scenarios to collect the metrics using the following factors: rekey protocol, key size, the size of the WSN, the mobility rate, and the departure rate. Each scenario is run ten times, totaling 1440 simulation runs.

# IV. Results and Analysis

This chapter presents and analyzes the experimental results. First, the methods used to verify and validate the simulation models are discussed in Section 4.1. Next, the results of each individual performance metric are presented in Section 4.2. Finally, an overall analysis of the results is provided in Section 4.3. Section 4.4 summarizes the chapter.

## 4.1 Model Verification and Validation

This section presents the verification and validation of the Matlab computer models in the experiment. Verification of a computer model ensures the model does what it is intended to do, that it has been debugged, and has been implemented properly [Jai91]. Since the model used is modified from previous research [Phi08], an incremental approach used during modification ensures the model continues to work properly.

### 4.1.1 Verification of the Matlab Simulation

The modified Matlab model has some key differences from the baseline Matlab model, which investigated the advantage of the Hubenko architecture in swarms of autonomous UAVs [Phi08]. The baseline model is modified to represent a field of WSN nodes instead of UAVs flying about a cluster leader [Phi08]. The modifications include changes to how nodes depart the network, when rekey operations are triggered and when rekey operation statistics are gathered. Additional functions are added to the modified model to represent the hierarchical and Secure Lock keying functions that are not in the baseline model.

The modified model has normally distributed departures instead of the uniform distribution of the baseline model. Since the modified model is adapted to represent battery powered devices with the same specifications conducting similar functions under similar

conditions, a normal distribution of the entire population more closely represents how nodes will expire when their battery is depleted than a uniform distribution. The model uses a mean departure time of 75% of the total simulation time with a standard deviation of 10% of the simulation time. When the simulation is set to a 25% departure rate, 75% of the nodes will function for the entire simulation time. The remaining 25% of the nodes depart according to the normal distribution described.

The modified model triggers cluster rekeys and collects statistics differently than the baseline model. Since a rekey to a hierarchical keyed cluster depends on the position of the node in the hierarchical tree, it is not feasible to wait until the end of the simulation's discrete time step to initiate the rekey process. The modified model triggers a rekey as soon as a cluster requires it. Metrics are collected at this time instead of waiting until the end of the time step.

To represent currently-available WSN security systems [L3C04, L3C04b], the simulation time in the modified model is extended from the baseline model's two hours to thirty days. To keep the time required to collect data reasonable, the discrete time step in the modified model is increased from one second to twenty seconds.

To draw comparisons between the three rekeying protocols, the pair-wise rekeying protocol in the baseline model is modified to record the number of bits transmitted instead of the number of rekeys performed.

### 4.1.2 Verification of the Secure Lock Measurements

To measure the performance of Secure Lock, a function is added to the Matlab simulation that reports the correct Secure Lock size in bits for a given number of recipients. An independent Matlab function determines the size of the Secure Lock rekey message sizes. However, due to the precision limitations of Matlab, only the 8-bit result is calculated in the

Matlab function. This function returns the same result as shown in Section 2.7.2.4, which is verified by hand calculation. An independent Java simulation [AnM08] with arbitrary precision provides the Secure Lock message sizes for larger networks and key sizes. The Java simulation also provides the same result as the example shown in Section 2.7.2.4.

In the Matlab simulation, the Secure Lock function returns the same results as the Java simulation as shown in Figure 11. Because the function returns the same result for the 8-bit example shown in Section 2.7.2.4 and displays the linear expansion of the message size as key size and network size increase (as described in Section 2.7.2.4), the Secure Lock function used in Matlab is considered verified.

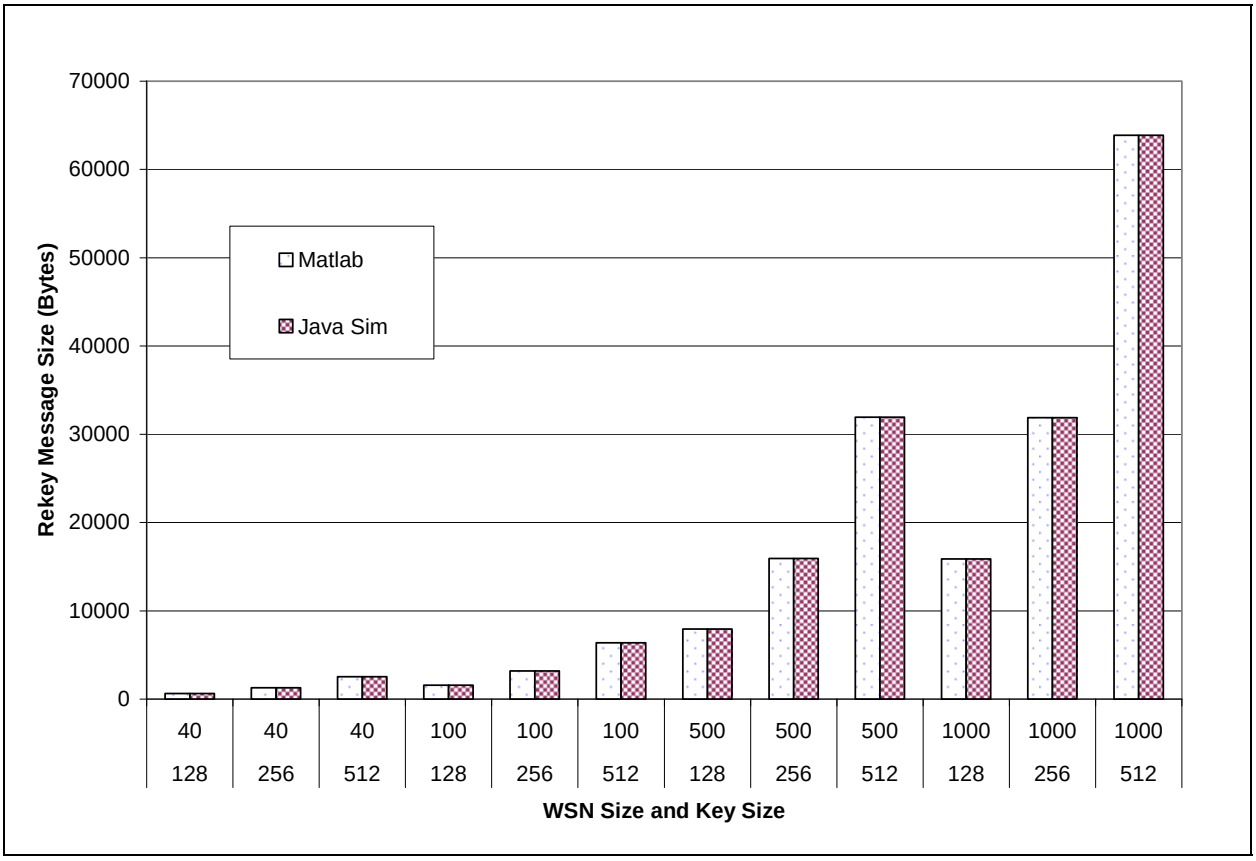


Figure 11. Verification and Validation of Secure Lock Function

### 4.1.3 Verification of the Hierarchical Measurements

To measure the rekey performance of a hierarchical tree, two functions are added to the Matlab simulation to report the correct number of keys transmitted when a node leaves or joins the cluster. These functions are verified against measurements of the hierarchical rekey protocol described in previous research [BaB02]. Figure 12 shows the verification and validation of the Matlab function compared to the expected result from previous research [BaB02]. The response variable in Figure 12 is the number of keys required to build the hierarchical tree.

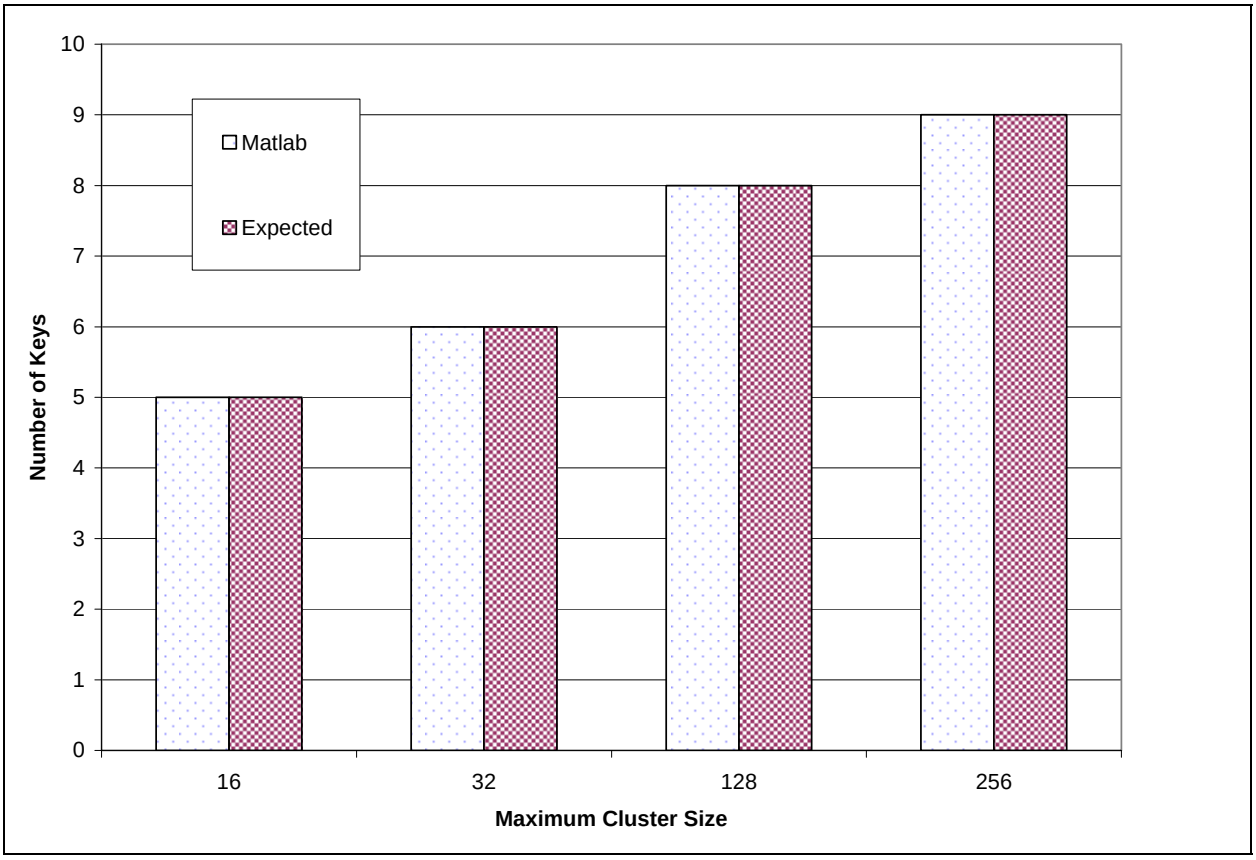


Figure 12. Verification and Validation of Hierarchical Function

### 4.1.4 Validation of the Matlab Simulation

Validation ensures that assumptions used in developing the model are reasonable and that the model produces results close to what is observed in real systems, theoretical results, or expert

intuition [Jai91]. The modified model is validated against the theoretical results of previous research [Phi08]. To validate the modified model against the baseline model, the modified model is compared to the baseline under the same conditions. To accurately compare with previous research [Phi08], the WSN sizes include 40, 100, 200 and 500 nodes, and the departure rates are 25% and 75%. The mobility rates include the previous research values of 25% and 75% [Phi08], as well as an additional mobility rate of 0% to verify the model works in the stationary case. Table 9 summarizes the factor levels used for validation. These levels differ slightly from the set of experimental levels; however the levels match previous research [Phi08] and allow direct comparison of the new model to the previous one.

Table 9. Factor Levels Used for Validation

| Factor | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Mobility Rate | 0% | 25% | 75% | |
| WSN Size | 40 | 100 | 200 | 500 |
| Departure Rate | 25% | 75% | | |

The validation results are shown in Figure 13 and Figure 14. Figure 13 shows the number of 256-bit keys transmitted during the entire simulation for four different WSN sizes when the entire network has a node departure rate of 75%. The responses graphed are for two cases of the baseline model (represented with dashed lines) and three cases of the modified model (plotted with solid lines). The lowest curve is the modified model with zero percent mobility; this curve parallels the baseline model. Since it maintains the lowest number of keys transmitted for all cases, it also maintains the correct relationship compared to the baseline model's 25% and 75% mobility cases.

The modified 25% and 75% mobility cases both report consistently higher numbers of keys transmitted than the baseline model for the same level of mobility. To support a fair comparison between hierarchical keying and the other methods, the model is modified from the baseline to immediately rekey the cluster when required. The baseline model waits until the end of the current discrete time step before it records a rekey operation. This difference results in more keys being transmitted in the modified model, consistent with the plotted responses. Again, the modified 25% and 75% mobility cases are consistent with the baseline responses. Therefore, the model is valid for a 75% departure rate.

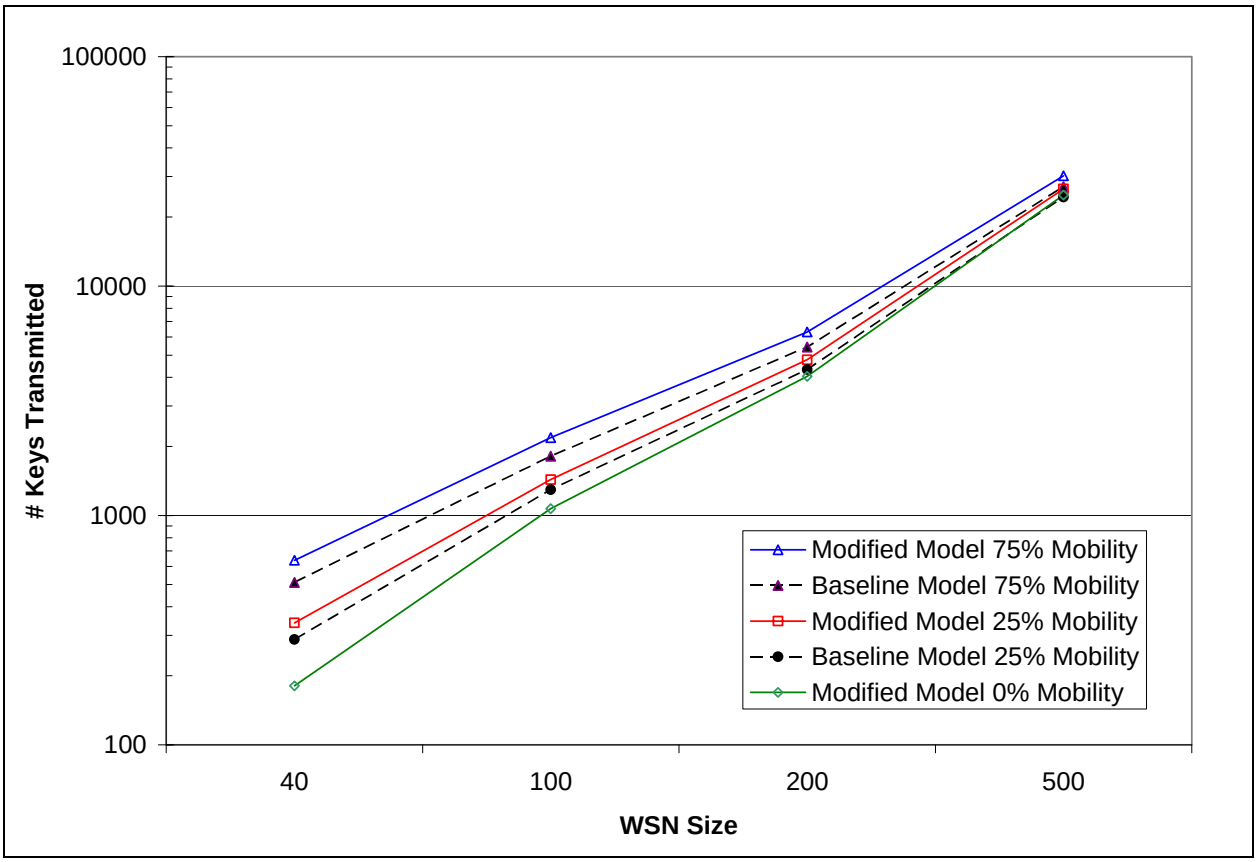


Figure 13. Model Validation with a 75% Departure Rate

Figure 14 is similar to Figure 13, but shows the results for a node departure rate of 25%. As before, the lowest curve is the modified model with zero percent mobility; this curve is

consistent to the baseline model. Since it maintains the lowest number of keys transmitted for all cases, it also has the correct relationship compared to the baseline model's 25% and 75% mobility cases. Again, the modified 25% and 75% mobility cases both have consistently higher numbers of keys transmitted than the baseline model for the same level of mobility. For the same reason given above, this is expected. As before, the modified 25% and 75% mobility cases are consistent with the baseline responses. Therefore, the model is valid for a 25% departure rate.

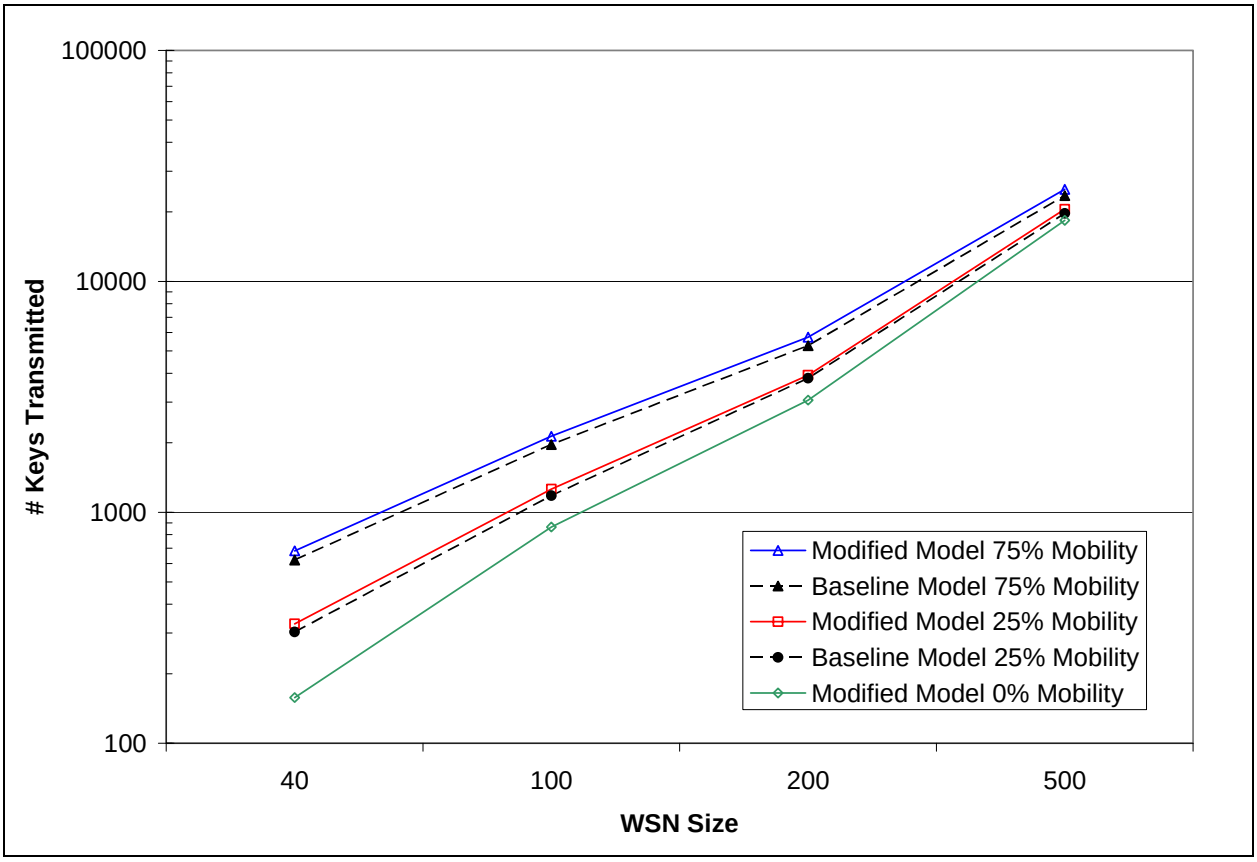


Figure 14. Model Validation with a 25% Departure Rate

The modified model performs as expected and is consistent with results from previous research [Phi08].

### 4.1.5 Validation of Secure Lock Measurements

To evaluate the performance of Secure Lock, a separate Java simulation is created for a variety of network conditions [AnM08]. The results from the Java simulation are coded into a Matlab function that returns the Secure Lock message size in bits based on the number of network nodes to be rekeyed and the size of the key used. As shown in Figure 11, the Matlab function returns Secure Lock message sizes identical to the published Java simulation results. Both functions return identical results to the example shown in Section 2.7.2.4. Therefore, the Secure Lock function is considered valid.

### 4.1.6 Validation of Hierarchical Measurements

To measure the performance of the hierarchical rekeying protocol, two Matlab functions are included in the simulation. The first Matlab hierarchical function reports the number of bits transmitted to a new cluster member if the rest of the hierarchical tree is not required to rekey, such as when an authorized user moving from one cluster to another. The number of bits transmitted in this case is based on the number of levels in the hierarchical tree. The second Matlab function returns the number of bits transmitted to the entire cluster if it is required to rekey. This number is based on the number of levels in the tree, as well as the location of the joining/departing node in the tree. The second function accounts for vacant spaces in the tree and does not transmit keys to users not in the tree. This simulates the GACA-GKM awareness of the membership of the hierarchical tree. Both functions are based on previous work [BaB02] and the values returned by both functions match hand-calculated values, as shown in Figure 12. Therefore, the hierarchical functions are considered valid.

## 4.2 Results and Analysis of Performance Metrics

This section presents and analyzes the results of the measurements made from the Matlab simulation. The number of bits transmitted is closely examined, followed by analysis of battery power and the amount of memory space consumed for rekey operations.

To detect if there is any statistically significant cause of variation within the experiment, an Analysis of Variance (ANOVA) table, like Table 10 is used. By filling in the table from left to right, the P value is calculated. The P value indicates whether a factor (such as rekey protocol) contributes more variation between different groups than is expected. If P is less than 0.05, the factor is considered statistically significant.

The raw bits transmitted data exhibited a residual-value-versus-fits plot with a strong increasing trend from very small to very large, which precludes the use of ANOVA. A log transform of the number of bits transmitted data successfully compensated the trend and allows for an ANOVA to be used.

### 4.2.1 Analysis of Bits Transmitted

The number of bits transmitted is measured from the communications relay on the edge of the WSN, as shown in Figure 9. Table 10 presents the general linear model ANOVA table for the log number of bits transmitted by the communications relay for rekey operations. The first and second order terms used in the ANOVA account for 99.95% of the variance in the number of bits transmitted. The three terms that contribute the most variance are marked in bold. Since all values in the P column are less than 0.05, all terms are considered statistically significant.

For the ANOVA results to be valid, the residuals must be independent and normally distributed with zero mean. Figure 15 presents four charts to examine these assumptions. The ideal, normal probability plot is a straight diagonal line. In the top left corner of Figure 15, the

normal probability plot shown has some curvature in tails, which indicates skewness in the data. Because the mean of the histogram shown in the bottom left corner is slightly offset to the right, the histogram also shows skewness in the data. Overall, the curve does follow a normal distribution and skewness is minor. The mean is zero to sixteen decimal places. The scatter plots on the right side of Figure 15 are visual tests of independence of the residuals. Since there are no clear trends in either plot, the residuals are assumed independent.

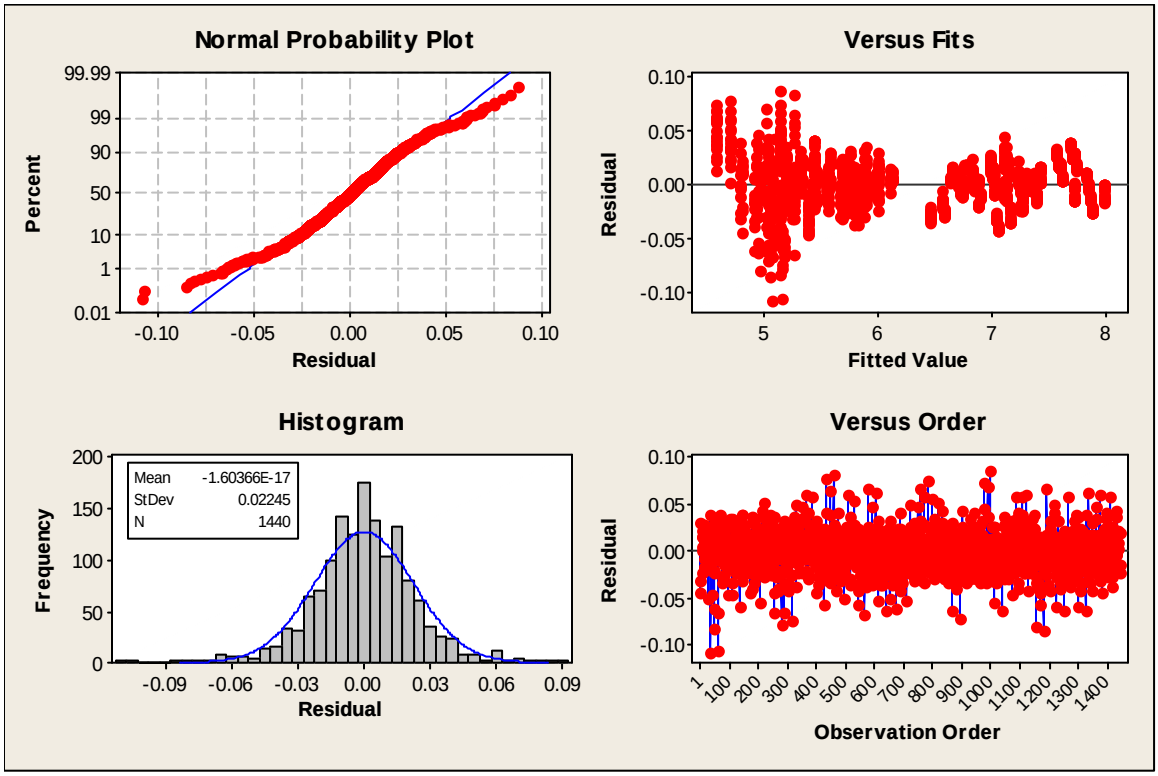


Figure 15. Visual Plots to Verify ANOVA Assumptions for All Data

The ANOVA shown in Table 10 attributes the majority of the variance to WSN Size (92.40%) followed by the second-order effect of rekey protocol * WSN size (2.67%) and then the rekey protocol (2.26%). The first-order effect of key size (1.87%) is next, followed by the departure rate (0.41%) and the second-order effect of (0.31%). The error term accounts for 0.05% and the remaining first and second order terms contribute less to the variance than the error term.

Table 10. ANOVA Results for Log Number of Bits Transmitted for All Data

| Source | DF | SS | % Variance | MS | F | P |
|---|---|---|---|---|---|---|
| **Rekey Protocol** | 2 | 29.879 | 2.26 | 14.940 | 29091.86 | 0.000 |
| **WSN Size** | 3 | 1220.004 | 92.40 | 406.668 | 791900.98 | 0.000 |
| Key Size | 2 | 24.663 | 1.87 | 12.331 | 24012.76 | 0.000 |
| Departure | 1 | 5.392 | 0.41 | 5.392 | 10500.03 | 0.000 |
| Mobility | 1 | 0.045 | 0.00 | 0.045 | 87.21 | 0.000 |
| **Rekey Protocol * WSN Size** | 6 | 35.224 | 2.67 | 5.871 | 11431.80 | 0.000 |
| Rekey Protocol * Key Size | 4 | 4.155 | 0.31 | 1.039 | 2022.92 | 0.000 |
| Rekey Protocol * Departure | 2 | 0.008 | 0.00 | 0.004 | 7.66 | 0.000 |
| WSN Size * Key Size | 6 | 0.203 | 0.02 | 0.034 | 65.81 | 0.000 |
| Error | 1412 | 0.725 | 0.05 | 0.001 | | |
| Total | 1439 | 1320.298 | 100.00 | | | |

Figure 16 shows the main effects plots, which visually presents the first-order effects of the measured data.

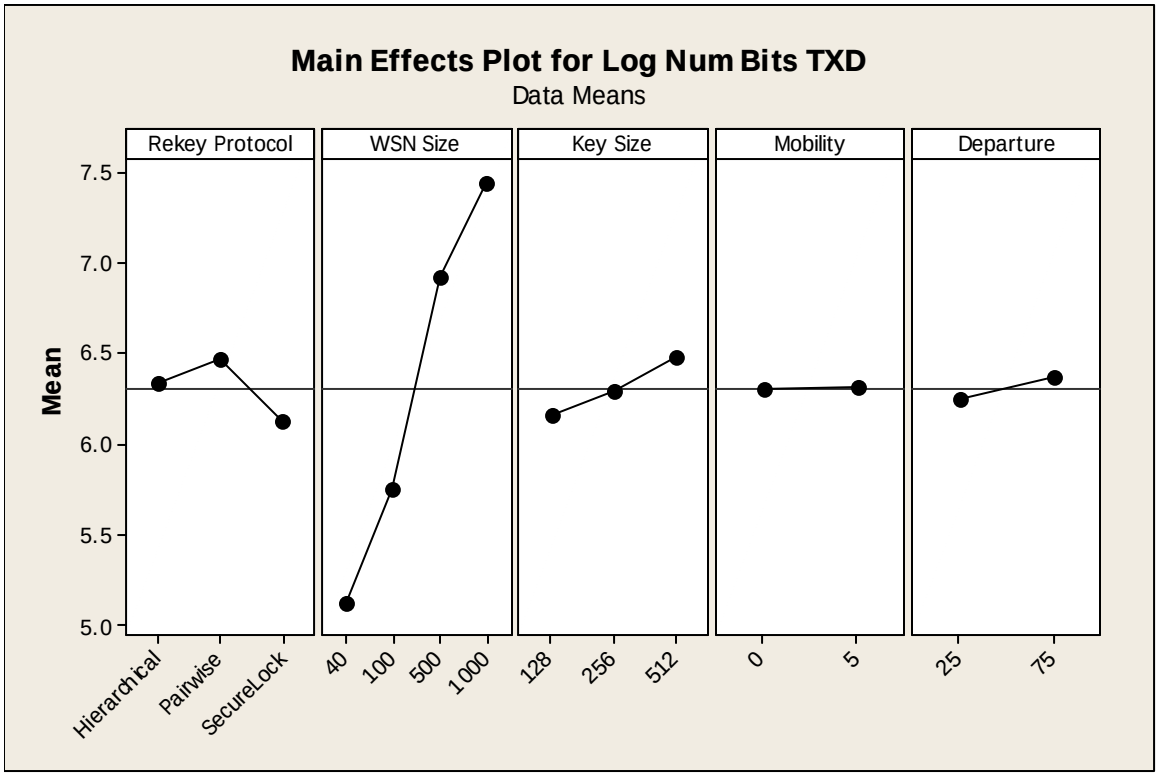


Figure 16. Main Effects Plots for All Data

Overall, the data shows that larger WSNs require more bits to be transmitted during rekey operations, but this is expected. As seen in the ANOVA, the WSN size has the largest effect.

This is expected since the number of bits transmitted increases as the WSN increases in size. However, since the WSN size accounts for so much of the variation (92.40%), additional analysis is performed to determine if more insight is gained by removing WSN size as a factor.

#### 4.2.1.1 Analysis of Bits Transmitted by WSN Size

The data is partitioned into four groups by WSN size and each group analyzed separately. Appendix C presents the plots of the data with confidence intervals. The first group analyzed is for the WSN size of 40 nodes. Figure 17 shows the four plots used to visually verify the underlying assumptions of the ANOVA shown in Table 11. In the top left of Figure 17, the normal probability plot is fairly linear. The bump near the middle indicates more residual measurements in this portion of the data than a normal distribution predicts. This is confirmed in the histogram in the lower left of Figure 17. The curve shown in the histogram is normal with the exception of the spike of residual measurements to the right of the mean. The mean is zero and there is very little skew. The spike in residual measurements is slight, but is accepted as not affecting normality since conclusions are drawn across all four cases of WSN size. On the right-hand side of Figure 17, there is no clear pattern, so the residuals are assumed to be independent. With the exception of mobility, the Appendix C plots indicate statistically significant differences between the data points.

Table 11 presents the ANOVA for a WSN size of 40 nodes. The factors presented account for 99.14% of the total variance. The values in the P column less than 0.05 are considered statistically significant. Since the P values for the second-order factors of rekey protocol * mobility and key size * departure are greater than 0.05, the null hypothesis cannot be ruled out and they are not considered to be statistically significant contributors to variance. The top three contributors to the variance are marked in bold. The largest is the rekey protocol

(73.40%), followed by the key size (19.81%) and then the departure rate (4.39%). The only term that contributed more than the error rate (0.86%) is the second-order effect of rekey protocol * key size (1.34%).

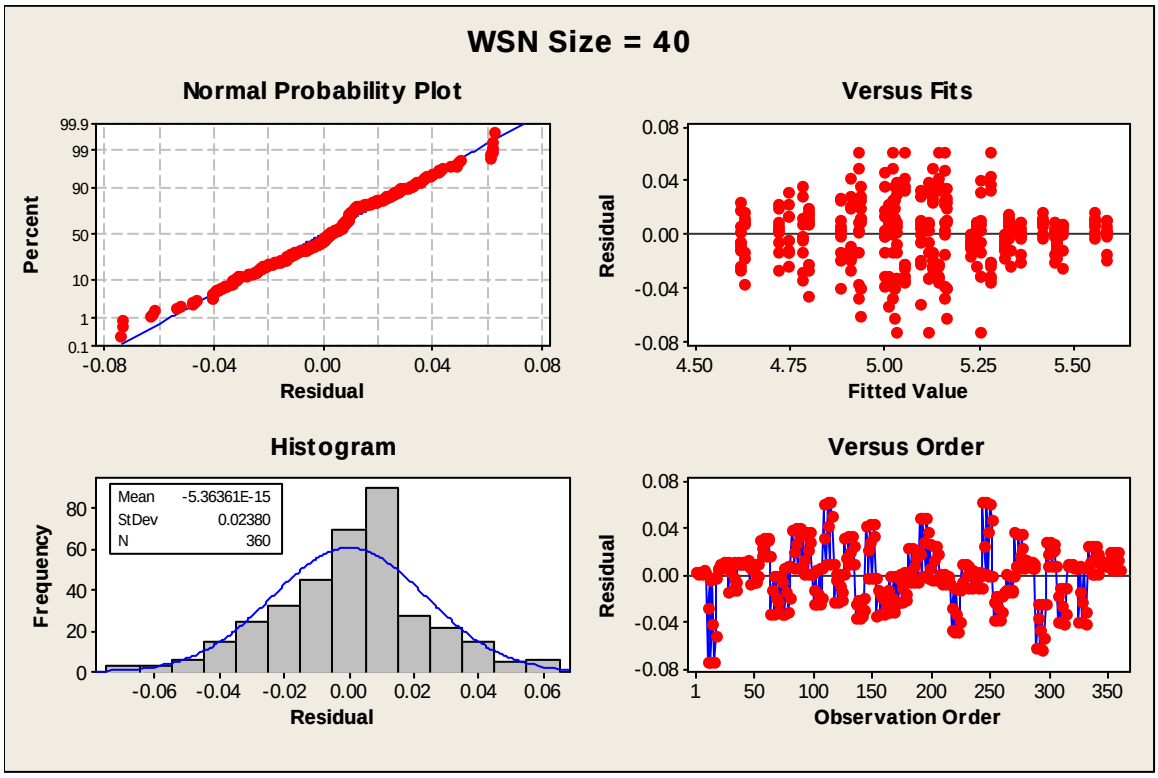


Figure 17. Visual Plots to Verify ANOVA Assumptions for WSN Size = 40 Nodes

Table 11. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 40 Nodes

| Source | DF | SS | % Variance | MS | F | P |
|---|---|---|---|---|---|---|
| **Rekey Protocol** | 2 | 17.3699 | 73.40 | 8.685 | 14686.78 | 0.000 |
| **Key Size** | 2 | 4.6878 | 19.81 | 2.344 | 3963.70 | 0.000 |
| **Departure** | 1 | 1.0395 | 4.39 | 1.040 | 1757.80 | 0.000 |
| Mobility | 1 | 0.0386 | 0.16 | 0.039 | 65.20 | 0.000 |
| Rekey Protocol * Key Size | 4 | 0.3167 | 1.34 | 0.079 | 133.87 | 0.000 |
| Rekey Protocol * Mobility | 2 | 0.0012 | 0.01 | 0.001 | 0.99 | 0.374 |
| Departure * Mobility | 1 | 0.0062 | 0.03 | 0.006 | 10.43 | 0.001 |
| Key Size * Departure | 2 | 0.0001 | 0.00 | 0.000 | 0.05 | 0.953 |
| Error | 344 | 0.2034 | 0.86 | 0.001 | | |
| Total | 359 | 23.6633 | 100.00 | | | |

Figure 18 shows the main effects plots for the 40 node WSN. Because it has the largest range of all four factors, the rekey protocol is the largest contributor to the response.

Hierarchical performs worse than pair-wise and both of them perform worse than Secure Lock. The response to key size is expected. As the key size increases, the number of bits transmitted increases. The departure rate response is also expected. As more nodes depart the network, more rekey operations are required, thus more bits must be transmitted. The mobility factor's slight response is expected since the Hubenko architecture does not require rekey operations for nodes that move between clusters.

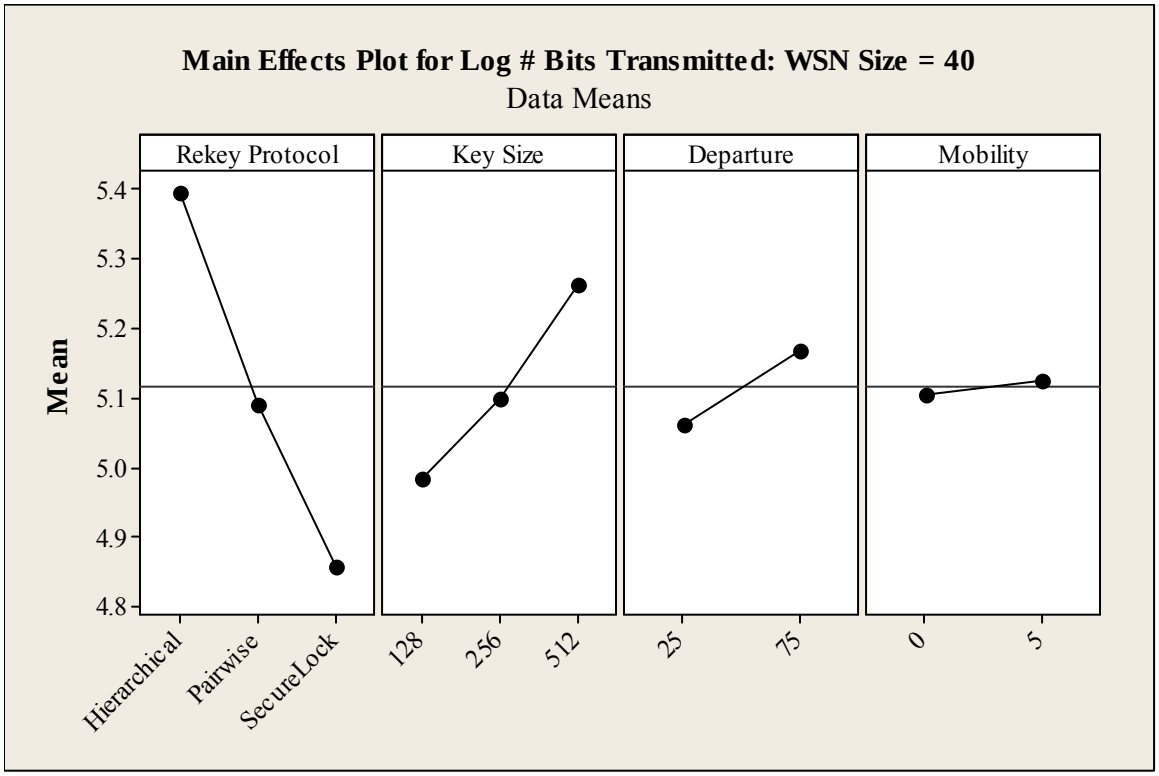


Figure 18. Main Effects Plots for WSN Size = 40

Figure 19 shows the interactions between the factors for a WSN size of 40 nodes. Most of the plots are comprised of parallel lines, indicating no significant interaction between the factors. For example, the plot in the bottom right corner of Figure 19 shows the interaction between the departure rate and the mobility rate. Since the lines are somewhat parallel and they never cross over each other, there is nothing of interest in the interaction between the two factors.

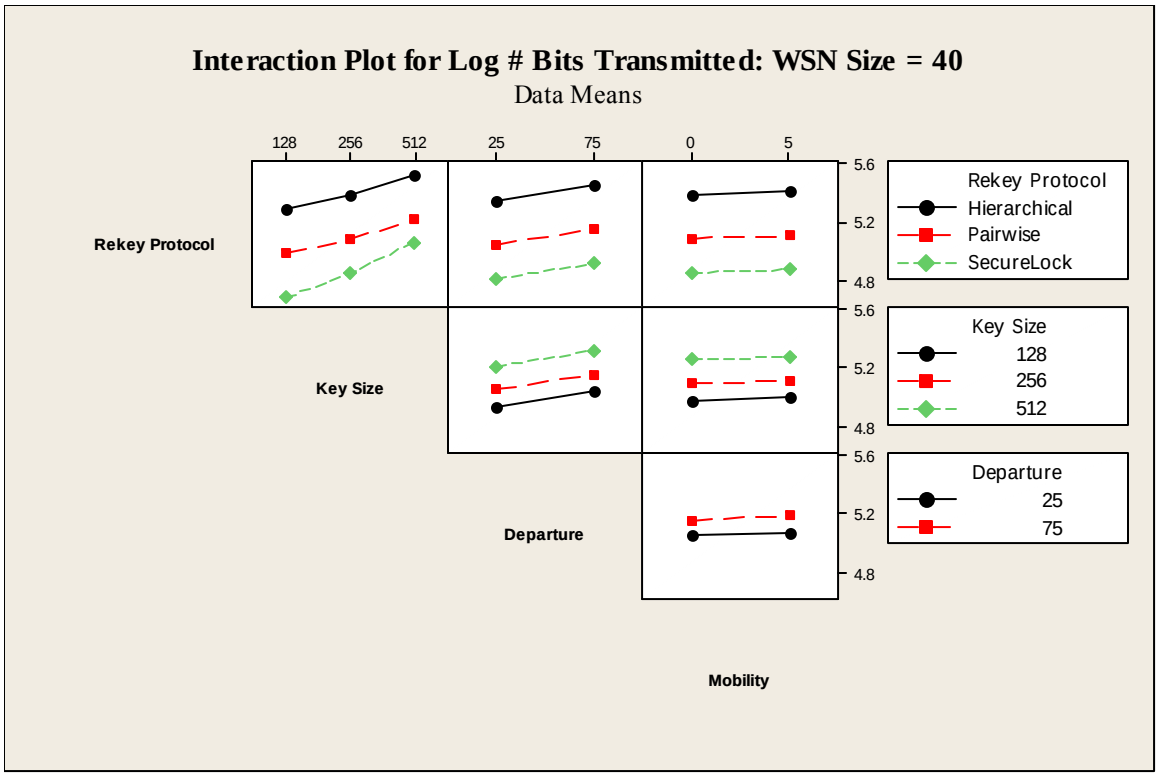


Figure 19. Factor Interaction Plot for WSN Size = 40 Nodes

For the WSN size of 100 nodes, Figure 20 presents visual plots to verify the assumptions of an ANOVA. In the top left of Figure 20, the normal probability plot shows a mostly linear plot along the diagonal axis, with a shallow dip at an approximate residual values of -0.01 and with curvature in the right tail at the residual value of approximately 0.03. The histogram in the bottom left shows a spike in residual measurements to the left of the mean which causes the shallow dip in top left plot. A cluster of residual measurements in the extreme right of the histogram corresponds to the curvature in the right tail of the normal probability plot. The mean is zero and there is slight skewness. However, the plot is largely normal. The right side of Figure 20 does not indicate any clear pattern, so the residuals are considered independent.

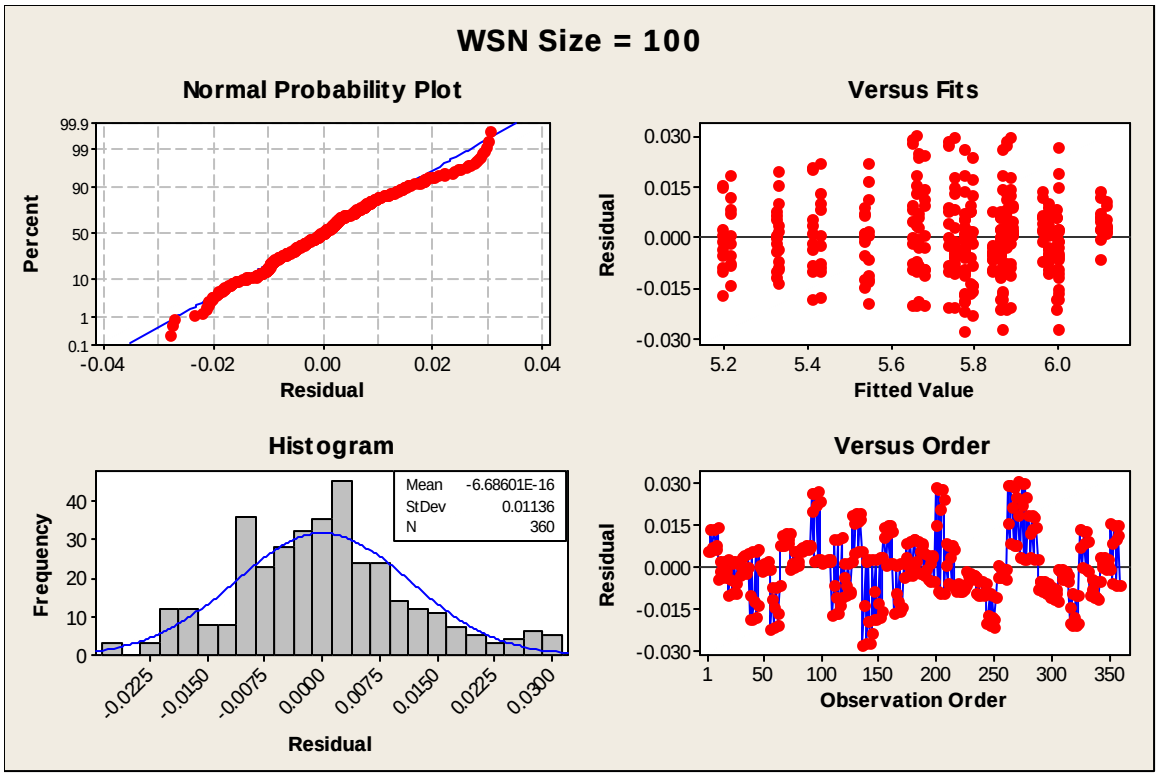


Figure 20. Visual Plots to Verify ANOVA Assumptions for WSN Size = 100 Nodes

Table 12 presents the ANOVA for a WSN size of 100 nodes. The factors in the table account for 99.77% of the total variance. Most of the factors are statistically significant; only the second-order effect of key size * departure rate cannot exclude the null hypothesis with a P value of 0.843.

Table 12. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 100 Nodes

| Source | DF | SS | % Variance | MS | F | P |
|---|---|---|---|---|---|---|
| **Rekey Protocol** | 2 | 12.5009 | 61.40 | 6.251 | 46370.63 | 0.000 |
| **Key Size** | 2 | 5.7150 | 28.07 | 2.858 | 21199.07 | 0.000 |
| **Departure** | 1 | 1.2959 | 6.36 | 1.296 | 9613.87 | 0.000 |
| Mobility | 1 | 0.0143 | 0.07 | 0.014 | 106.38 | 0.000 |
| Rekey Protocol * Key Size | 4 | 0.7844 | 3.85 | 0.196 | 1454.85 | 0.000 |
| Rekey Protocol * Mobility | 2 | 0.0017 | 0.01 | 0.001 | 6.36 | 0.002 |
| Departure * Mobility | 1 | 0.0026 | 0.01 | 0.003 | 19.31 | 0.001 |
| Key Size * Departure | 2 | 0.0000 | 0.00 | 0.000 | 0.17 | 0.843 |
| Error | 344 | 0.0464 | 0.23 | 0.000 | | |
| Total | 359 | 20.3613 | 100.00 | | | |

The three terms that contribute the most to the variance are highlighted in bold. The largest contributor is the rekey protocol (61.40%) followed by the key size (28.07%) and then the departure rate (6.36%). The only second order effect to contribute more to the variance than the error rate (0.23%) is rekey protocol * key size (3.85%).

Figure 21 shows the main effects plots for the 100 node WSN. The rekey protocol has the largest variance. Its performance is similar to the 40 node network. Hierarchical performs the worse than pair-wise, and both of them perform worse than Secure Lock. The response to key size is expected; as the key size increases, the number of bits transmitted increases. The departure rate response is also expected. As more nodes depart the network, more rekey operations are required, thus more bits must be transmitted. The mobility factor's slight response is expected due to the efficiency of the Hubenko architecture. With the exception of mobility, the Appendix C plots indicate statistically significant differences between the data points.

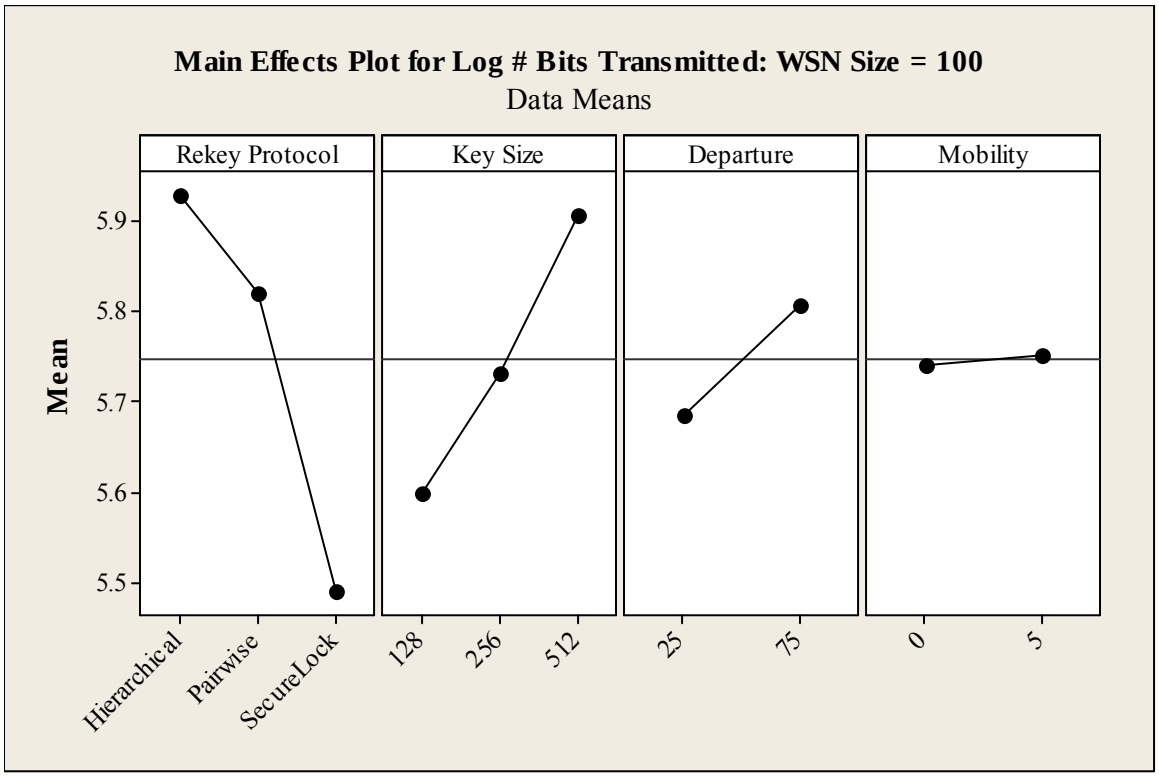


Figure 21. Main Effects Plots for WSN Size = 100

Figure 22 displays the interactions between the factors for a WSN size of 100 nodes. Most of the plots are comprised of parallel lines and none cross, indicating no significant interaction between the factors.

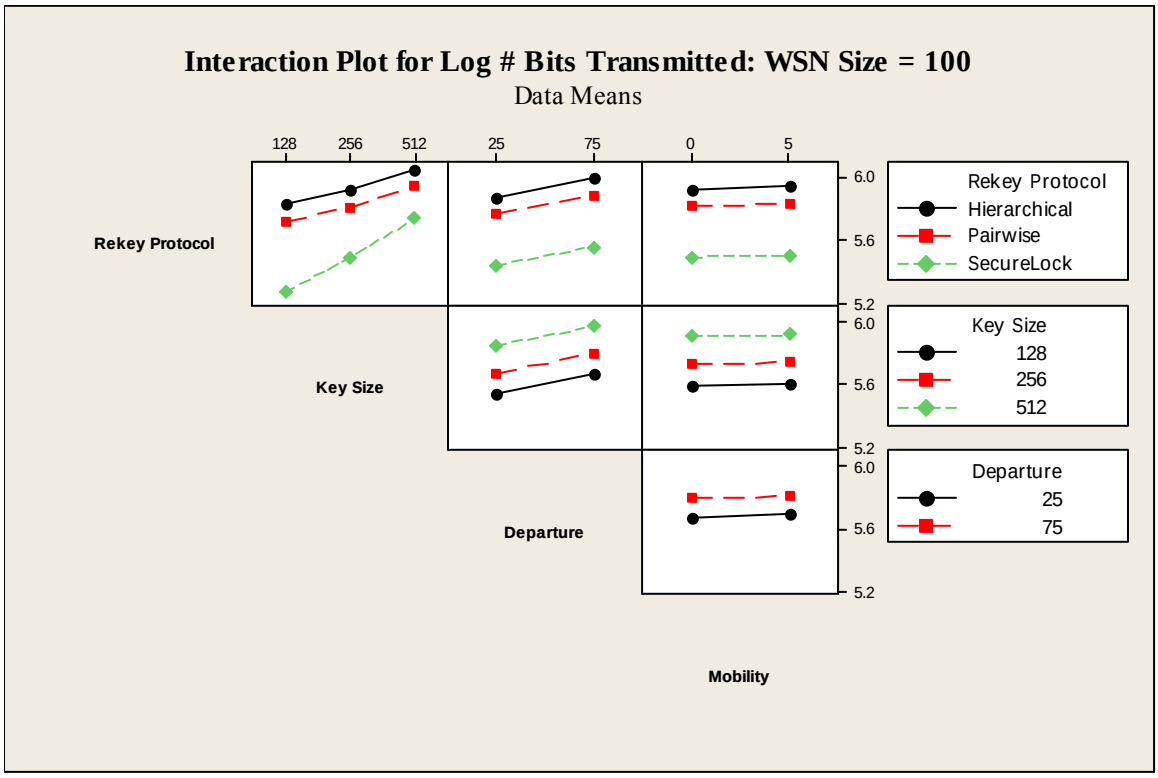


Figure 22. Factor Interaction Plot for WSN Size = 100 Nodes

Figure 23 presents the visual plots to verify the assumptions required for the ANOVA table to be valid for a WSN size of 500 nodes. In the top left of Figure 23, the normal probability plot shows a mostly linear plot along the diagonal axis, with curvature in the tails. The histogram in the bottom left shows a spike in residual measurements in the extreme right of the histogram, which corresponding to the curvature in the right tail of the normal probability plot. The histogram ends on the left side at -0.09. Going from approximately five measurements to none past -0.09 causes the downward curvature in the left end of the normal probability plot. The mean is zero and there is slight skewness. However, the plot is mostly normal. The right

side of Figure 23 does not indicate any clear pattern or trend, so the residuals are considered independent.

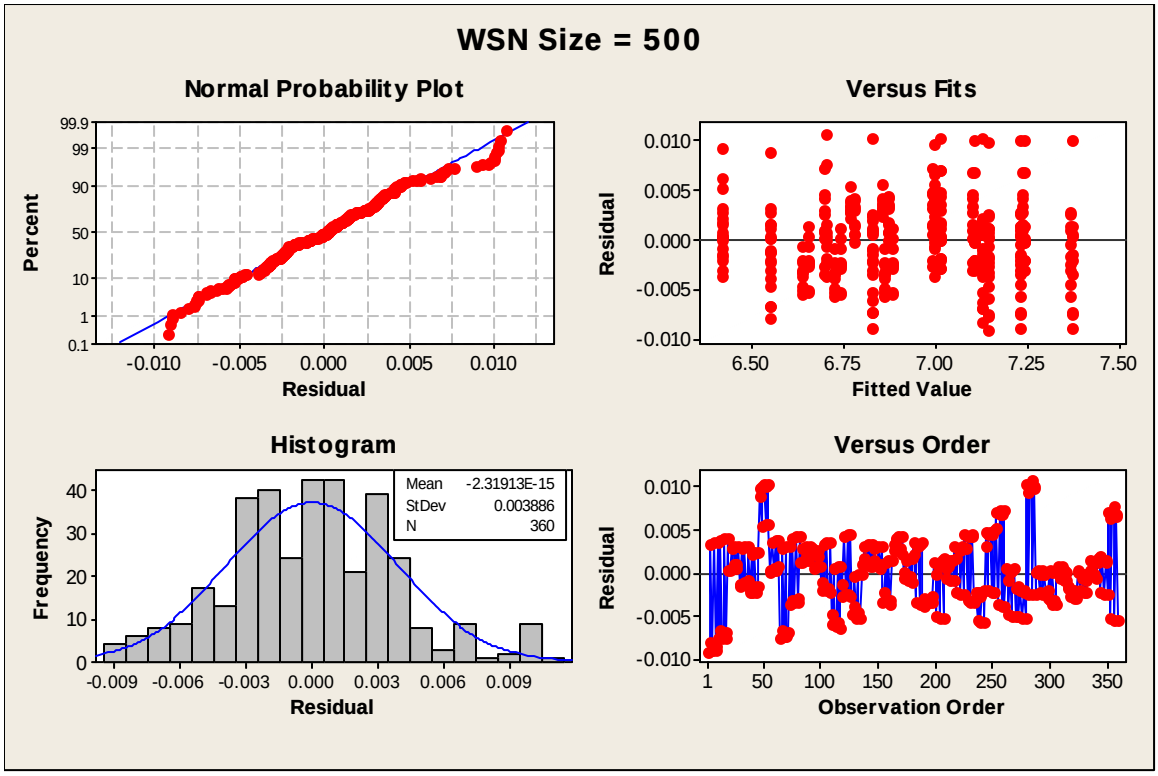


Figure 23. Visual Plots to Verify ANOVA Assumptions for WSN Size = 500 Nodes

Table 13 presents the ANOVA for a 500 node WSN.

Table 13. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 500 Nodes

| Source | DF | SS | % Variance | MS | F | P |
|---|---|---|---|---|---|---|
| **Rekey Protocol** | 2 | 12.4049 | 54.65 | 6.202 | 393578.88 | 0.000 |
| **Key Size** | 2 | 7.1088 | 31.32 | 3.554 | 225544.78 | 0.000 |
| Departure | 1 | 1.5299 | 6.74 | 1.530 | 97081.15 | 0.000 |
| Mobility | 1 | 0.0035 | 0.02 | 0.004 | 221.22 | 0.000 |
| **Rekey Protocol * Key Size** | 4 | 1.6411 | 7.23 | 0.410 | 26034.29 | 0.000 |
| Rekey Protocol * Mobility | 2 | 0.0041 | 0.02 | 0.002 | 130.57 | 0.000 |
| Departure * Mobility | 1 | 0.0007 | 0.00 | 0.001 | 42.45 | 0.000 |
| Key Size * Departure | 2 | 0.0000 | 0.00 | 0.000 | 0.11 | 0.898 |
| Error | 344 | 0.0054 | 0.02 | 0.000 | | |
| Total | 359 | 22.6983 | 100.00 | | | |

The factors account for 99.98% of the variance in the log number of bits transmitted for rekeys to the network. With the exception of the second order factor of key size * departure rate, all factors are statistically significant, with P values less than 0.05. The top three factors are marked in bold. The largest contributor to the variance is rekey protocol (54.65%) while the key size (31.32%) is the second largest contributor. In third place is the second-order factor of rekey protocol * key size (7.23%). Departure (6.74%) is the only other factor to have more contribution to the variance than the error rate (0.02%).

Figure 24 shows the main effects plot when the WSN size is 500 nodes. The main difference between this plot and the ones for the smaller network sizes, is the hierarchical rekey protocol, with a mean of approximately 6.8, is almost on par with Secure Lock, with a mean value of approximately 6.75. Both perform better than pair-wise, which has a mean value of almost 7.2. The other factors exhibit similar results as the smaller networks.

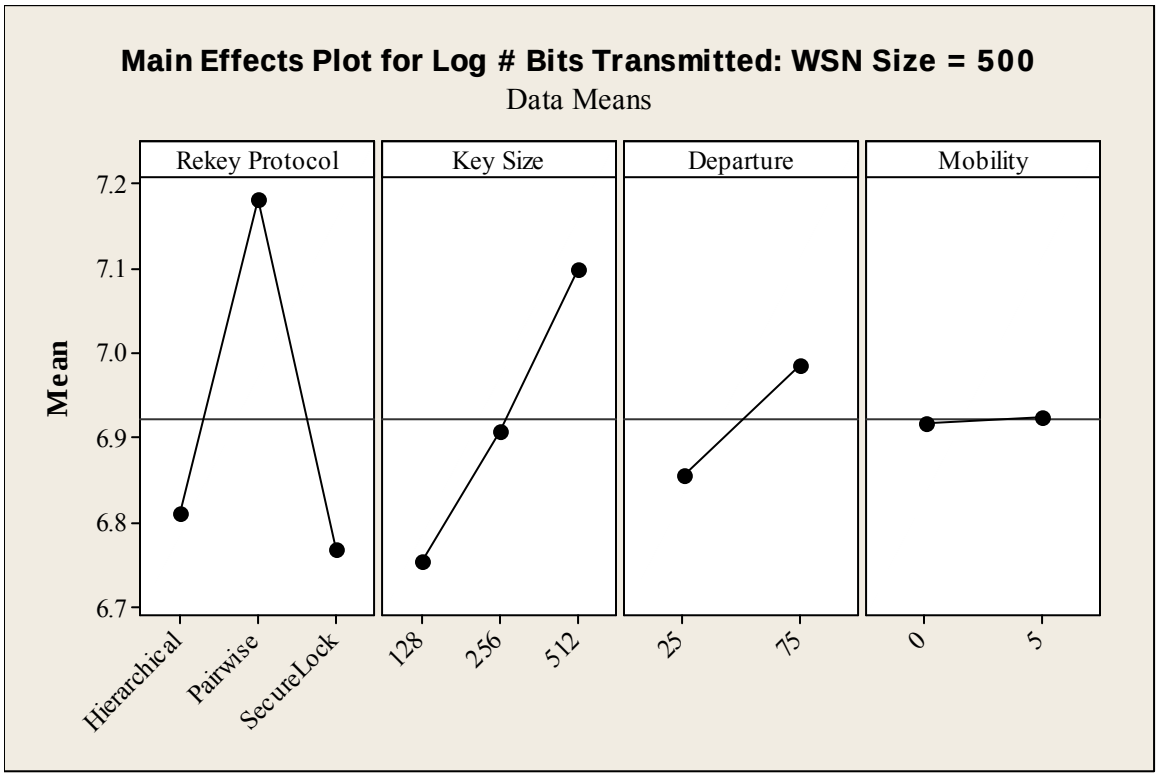


Figure 24. Main Effects Plots for WSN Size = 500

Figure 25 presents the interactions between factors when the WSN size is 500 nodes. This chart shows an interesting interaction between the rekey protocol and key size. Secure Lock performs the best when the key size is 128 bits, but performs only slightly better than hierarchical when the key size becomes 256 bits. When the key size is increases to 512 bits, the hierarchical rekey protocol is the best performer. The other factors have parallel lines that do not cross over each other, indicating no significant interaction between the other factors. With the exception of mobility, the Appendix C plots indicate statistically significant differences between the data points.

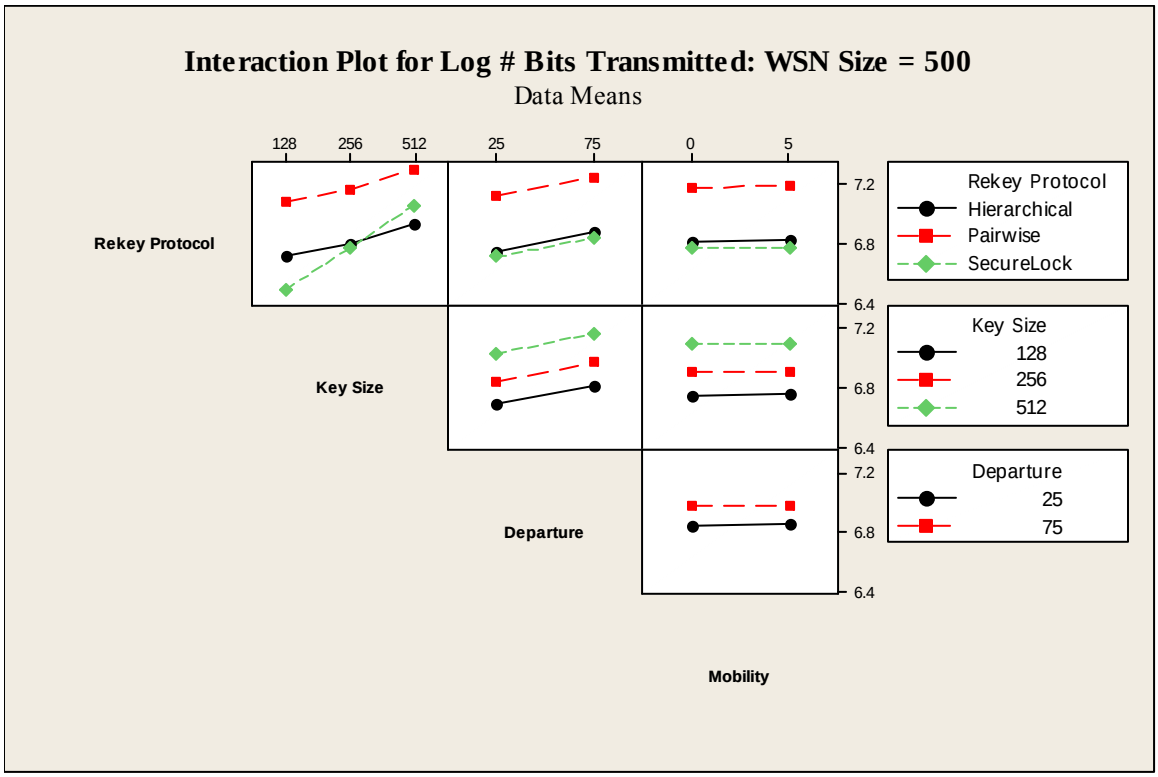


Figure 25. Factor Interaction Plot for WSN Size = 500 Nodes

Figure 26 presents the visual plots to verify the assumptions required for the ANOVA table to be valid for a WSN size of 1,000 nodes. In the top left of Figure 26, the normal probability plot shows a mostly linear plot along the diagonal axis, with curvature in the tails. The histogram in the bottom left shows higher than expected numbers of residual measurements

in the extreme left and right of the histogram, which corresponds to the curvature in the tails of the normal probability plot. The mean is zero and there is little, if any, skewness. Overall, the plot trends along the overlaid normal distribution line. The right side of Figure 26 does not indicate any clear pattern or trend, so the residuals are considered independent.

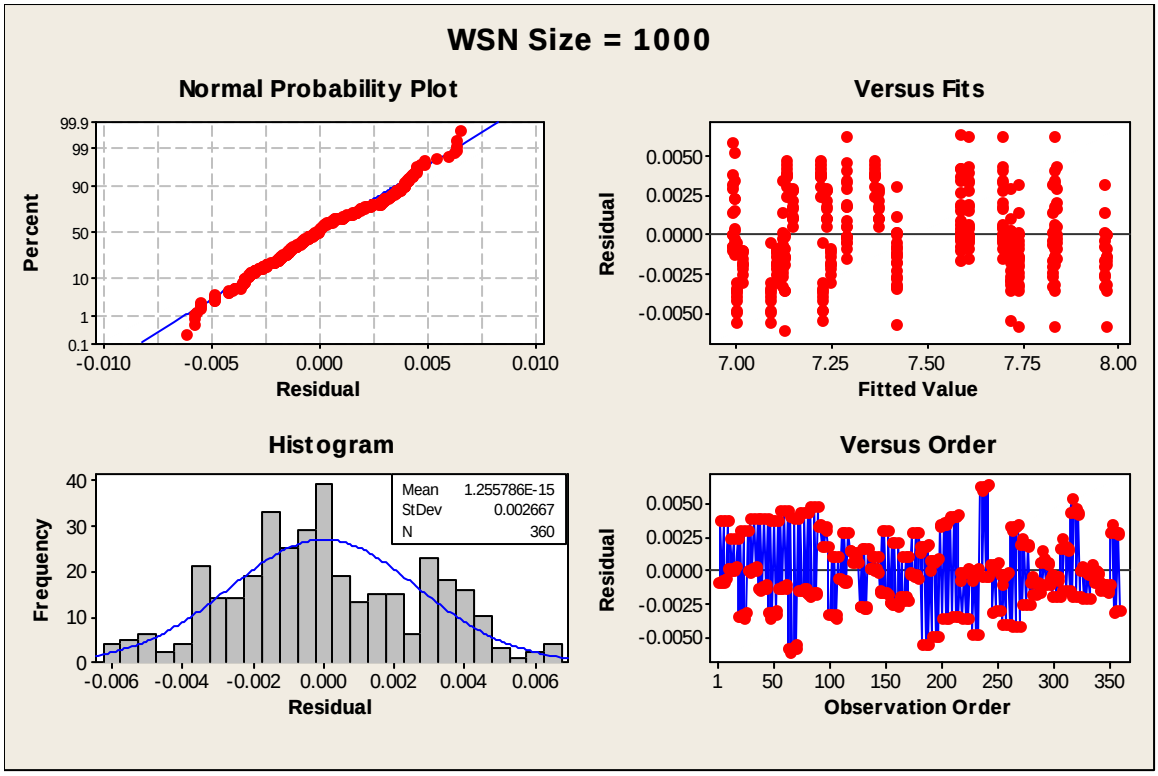


Figure 26. Visual Plots to Verify ANOVA Assumptions for WSN Size = 1000 Nodes

The ANOVA table for a WSN size of 1,000 nodes is presented in Table 14. The factors in the table account for 99.99% of the variance between measurements. With the exception of key size * departure rate, all of the factors are statistically significant with P values less than 0.05. The top three factors to influence the variance are marked in bold. Rekey protocol (68.00%) is followed by key size (21.91%) and the second order effect of rekey protocol * key size (5.42%). Departure rate (4.65%) is the only other factor to cause more variance than the error rate (0.01%).

Table 14. ANOVA Results for Log Number of Bits Transmitted for WSN Size = 1000 Nodes

| Source | DF | SS | % Variance | MS | F | P |
|---|---|---|---|---|---|---|
| **Rekey Protocol** | 2 | 22.8273 | 68.00 | 11.414 | 1537830.01 | 0.000 |
| **Key Size** | 2 | 7.3539 | 21.91 | 3.677 | 495419.42 | 0.000 |
| Departure | 1 | 1.5610 | 4.65 | 1.561 | 210316.36 | 0.000 |
| Mobility | 1 | 0.0023 | 0.01 | 0.002 | 311.69 | 0.000 |
| **Rekey Protocol * Key Size** | 4 | 1.8187 | 5.42 | 0.455 | 61262.18 | 0.000 |
| Rekey Protocol * Mobility | 2 | 0.0048 | 0.01 | 0.002 | 322.76 | 0.000 |
| Departure * Mobility | 1 | 0.0003 | 0.00 | 0.000 | 43.06 | 0.000 |
| Key Size * Departure | 2 | 0.0000 | 0.00 | 0.000 | 0.00 | 0.999 |
| Error | 344 | 0.0026 | 0.01 | 0.000 | | |
| Total | 359 | 33.5709 | 100.00 | | | |

Figure 27 presents the main effects plot when the WSN size is 1,000 nodes. The notable difference between this chart and the previous ones is that the hierarchical rekey protocol is the best performing rekey protocol, with Secure Lock in second place. Pair-wise is the worst performing rekey protocol as it requires the most bits to be transmitted for rekey operations.

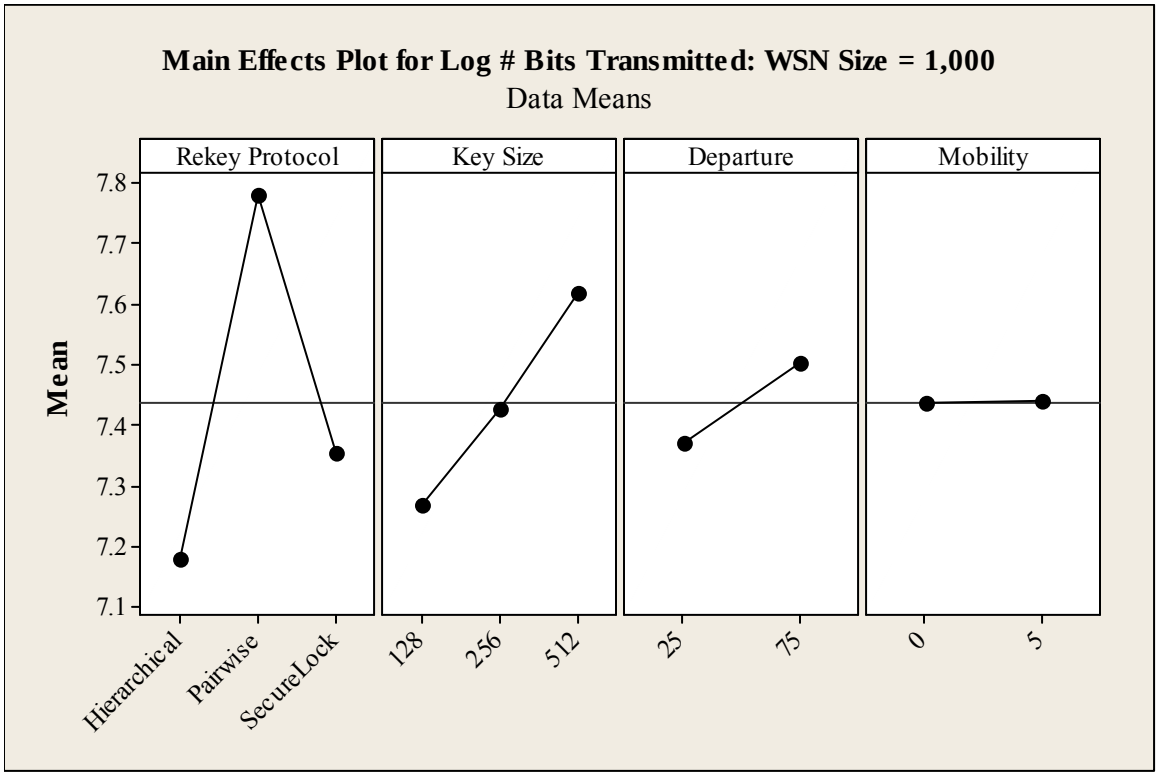


Figure 27. Main Effects Plots for WSN Size = 1,000

Figure 28 presents the interactions between factors when the WSN size is 1,000 nodes. This chart continues the trend seen in the previous interaction chart in Figure 25. Figure 28 shows a similar interaction between the rekey protocol and key size. Secure Lock and hierarchical perform in a similar fashion when the key size is 128 bits, but hierarchical performs better for key sizes of 256 and 512 bits. Pair-wise is the worst performer. The other factors exhibit parallel lines that do not cross over each other, indicating no significant interaction between the other factors. With the exception of mobility, the Appendix C plots indicate statistically significant differences between the data points.

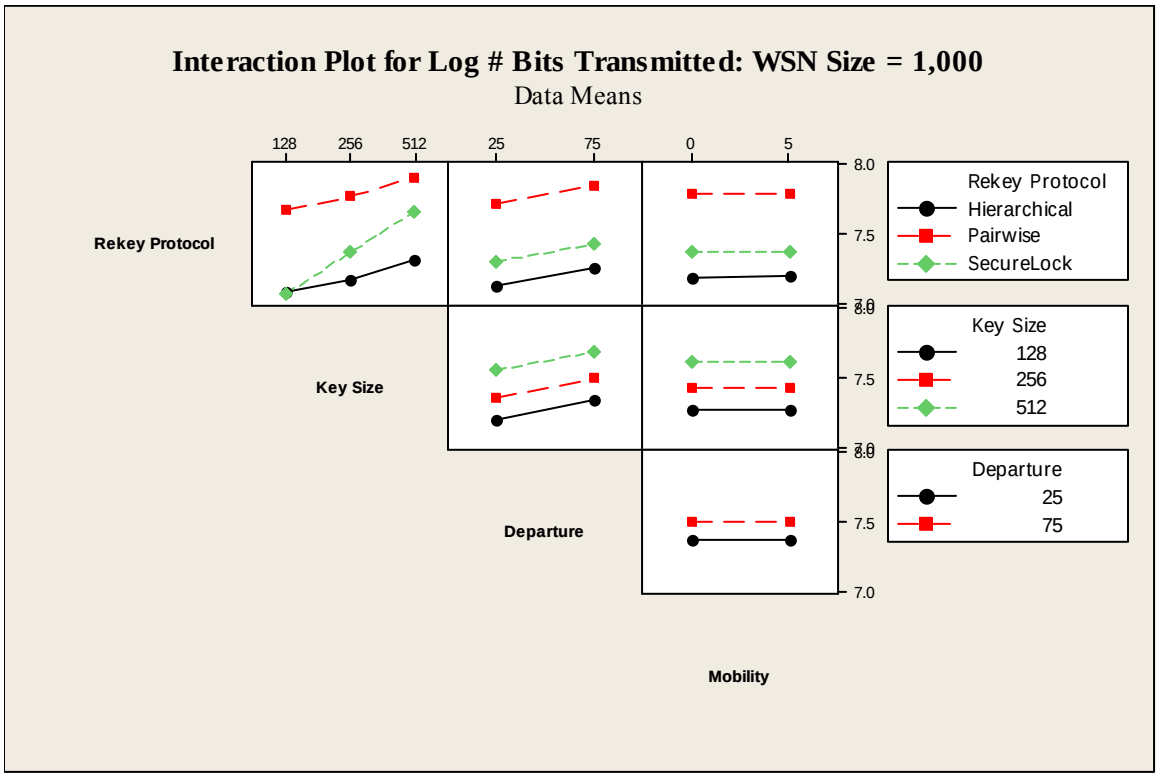


Figure 28. Factor Interaction Plot for WSN Size = 1,000 Nodes

#### 4.2.1.2 Combined Analysis of Bits Transmitted

Appendix C presents the Log Number of Bits Transmitted data with the confidence intervals included. When the individual results for all four WSN sizes are looked at in aggregate, there are several aspects that are consistent with the results of the combined data analysis.

Mobility has no significant effect on the number of bits transmitted. The node mobility rate either matches or is exceeded by the error rate for every ANOVA table presented. This is expected since the Hubenko architecture minimizes the number of rekey transmissions when an authorized user moves from one cluster to another. The Hubenko architecture does not require a rekey of the cluster that a node departed from if it remains in the multicast group and becomes a member of a new cluster. Likewise, the new cluster does not need to rekey since an already authorized node has joined the cluster. In this case, the mobile node only needs the required keys to communicate within the new cluster. Otherwise, a node that is new to the multicast group, or departing the multicast group entirely, requires a rekey of the cluster that the node is joining or leaving to ensure forward and backward security.

Additionally, the departure rate and key size both have direct impacts on the amount of bits transmitted for rekey operations. As expected, when the departure rate increases, so did the number of bits transmitted. As the key size increases, the number of bits transmitted increases as well. While the key size has a linear increase, the result on the number of bits transmitted is not linear because when the key size is 512 bits and the WSN size is 1,000 nodes, the Secure Lock message size exceeds the maximum transfer unit size. This causes the Secure Lock message to be broken up across multiple packets, incurring additional overhead.

The most interesting insight gained from removing the WSN size from the ANOVA tables is the impact of the rekey protocol on the number of bits transmitted. The pair-wise protocol is outperformed in every case. Secure Lock yields the fewest bits transmitted for networks of 100 or less nodes. The hierarchical protocol performs best for the largest network of 1,000 nodes. When the network size is 500 nodes, the best rekey protocol depends on the size of

the key. For largest key size, hierarchical performs best, while Secure Lock performs best for the two smaller key sizes of 128 and 256 bits.

Table 15 shows the mean number of bits transmitted for all levels of WSN size, key size, and rekey protocol. Appendix D presents this data with the confidence intervals including Figure 37, a plot of all the data shown in Table 15.

Table 15. Mean Number of Bits Transmitted

| | | **Rekey Protocol** | | **( x 10,000)** |
|---|---|---|---|---|
| **WSN Size** | **Key Size** | **Pair-wise** | **Hierarchical** | **Secure Lock** |
| 40 | 128 | 9.8 | 19.7 | 4.8 |
| | 256 | 12.1 | 24.2 | 7.0 |
| | 512 | 16.5 | 33.2 | 11.5 |
| 100 | 128 | 52.4 | 67.4 | 18.5 |
| | 256 | 64.4 | 82.9 | 30.5 |
| | 512 | 88.3 | 113.7 | 54.4 |
| 500 | 128 | 1207.0 | 516.1 | 306.9 |
| | 256 | 1482.9 | 634.1 | 586.4 |
| | 512 | 2034.7 | 870.1 | 1158.7 |
| 1000 | 128 | 4777.1 | 1202.4 | 1156.0 |
| | 256 | 5869.1 | 1477.2 | 2287.4 |
| | 512 | 8052.9 | 2026.9 | 4551.3 |

Table 16 presents the analysis of the data shown in Table 15, where the hierarchical and Secure Lock rekey protocols are presented as a percentage of the baseline rekey protocol, pair-wise. As seen in earlier analysis, hierarchical performs worse than pair-wise for the two smallest network sizes of 40 and 100 nodes. Secure Lock always outperforms pair-wise, and is usually the best performing with the exception of three configurations where hierarchical performs best. The best performing rekey protocol is shaded in grey.

Table 16. Mean Number of Bits Transmitted: Percentage of Baseline (pair-wise)

| | | Rekey Protocol | |
|---|---|---|---|
| **WSN Size** | **Key Size** | **Hierarchical** | **Secure Lock** |
| 40 | 128 | 200.68 % | 48.85 % |
| | 256 | 200.68 % | 58.38 % |
| | 512 | 200.68 % | 69.68 % |
| 100 | 128 | 128.72 % | 35.28 % |
| | 256 | 128.72 % | 47.32 % |
| | 512 | 128.72 % | 61.61 % |
| 500 | 128 | 42.76 % | 25.42 % |
| | 256 | 42.76 % | 39.54 % |
| | 512 | 42.76 % | 56.95 % |
| 1000 | 128 | 25.17 % | 24.20 % |
| | 256 | 25.17 % | 38.97 % |
| | 512 | 25.17 % | 56.52 % |

#### 4.2.1.3 Additional Observations

It should be noted that the implementation of the hierarchical rekey protocol in the simulation places it at a disadvantage. Because nodes are randomly assigned to clusters, each hierarchical tree has excess capacity. For example, in a network of 40 nodes, there should be an average of four nodes per cluster. However, in practice, it is not uncommon for a cluster to have more than eight nodes assigned. For pair-wise and Secure Lock, this does not present a problem since each protocol can handle any number of nodes within a cluster. However, hierarchical trees are based on powers of two. For example, to ensure sufficient capacity in all ten clusters of a 40-node network, each hierarchical cluster has a capacity of 16 nodes and the entire network has a capacity of 160 nodes, even though not all are used. The excess capacity in this case requires two additional keys to be transmitted (six instead of four) per rekey operation per cluster. If nodes can be efficiently assigned to hierarchical clusters thereby eliminating this wasted space, hierarchical rekeying can perform more efficiently when nodes depart. If realized

in the 40 node case, hierarchical rekeying would require approximately 66.7% of the bits transmitted and the entries in Table 16 for a WSN size of 40 nodes would improve from 200% to 134%. Since each hierarchical tree has excess capacity, hierarchical rekeying will improve for all WSN sizes, but the improvement will be less for larger networks than the smaller ones since the number of keys eliminated becomes less significant compared to the number of keys required to maintain the hierarchical tree. These gains are offset by the management and coordination communications needed to organize the nodes properly to fill the hierarchical trees efficiently and may not be worth the communication and processing cost. This is left for future work.

It should also be noted that Secure Lock has an advantage in the simulation. The Matlab simulation does not accommodate retransmissions of rekey messages. As noted in previous research [AnM08], Secure Lock performs best when all nodes receive the Secure Lock in a single transmission. If the Secure Lock message must be re-transmitted, the large size of the Secure Lock message hampers performance. The application of Secure Lock in a network simulator that accommodates network conditions requiring retransmissions is left for future work.

### 4.2.2 Analysis of Battery Power Consumed

In a wireless sensor network, preservation of battery power is an important consideration since nodes cease to function when the limited battery power is expended. This section analyzes the impact of the Section 4.2.1.2 results on the battery life span in a WSN.

Given that a typical WSN node uses two AA batteries [Cro08] and those batteries have a capacity of 2.85 mAh [Ene08], the total battery capacity is known. The power consumption of a typical WSN node is known to be 17 mA when transmitting at 3 dBm and 16 mA when receiving

[Cro08]. Using the mean number of bits transmitted from Table 15, the amount of time required to transmit rekey messages from the communications relay is

Time to transmit (relay) or receive (WSN) = number of bits / link capacity (2)

Given the link capacity of 250 kb/s [Cro08], the time for each case is shown in Table 17. The time to transmit from the communications relay is the same as required by the WSN to receive the transmission.

Table 17. Time Required (seconds) to Transmit the Mean Number of Bits From the Relay

| | | **Rekey Protocol** | | |
|---|---|---|---|---|
| **WSN Size** | **Key Size** | **Pair-wise** | **Hierarchical** | **Secure Lock** |
| 40 | 128 | 0.39 | 0.79 | 0.19 |
| | 256 | 0.48 | 0.97 | 0.28 |
| | 512 | 0.66 | 1.33 | 0.46 |
| 100 | 128 | 2.10 | 2.70 | 0.74 |
| | 256 | 2.57 | 3.31 | 1.22 |
| | 512 | 3.53 | 4.55 | 2.18 |
| 500 | 128 | 48.28 | 20.65 | 12.28 |
| | 256 | 59.32 | 25.36 | 23.45 |
| | 512 | 81.39 | 34.80 | 46.35 |
| 1000 | 128 | 191.09 | 48.10 | 46.24 |
| | 256 | 234.76 | 59.09 | 91.50 |
| | 512 | 322.12 | 81.08 | 182.05 |

Given the time required for the transmission, the amount of battery power consumed by the receiving WSN nodes is calculated by multiplying the transmission time by the current required by the WSN node in receive mode. Finally, the percentage of battery life savings of the hierarchical and Secure Lock rekey protocols over the baseline pair-wise is

% Battery Life Savings = (pair-wise power – other power)/Total battery life * 100 (3)

where the "other power" is power consumption for the hierarchical or Secure Lock case and the total battery life is 10,260 mA-secs given by 2.85 mAh * 60 min/hr * 60 sec/hr. The percentage-

of-battery-life-saved results are shown in Table 18. Positive numbers reflect savings over pair-wise, while negative numbers indicate pair-wise consumed less power. The best performer is shaded grey.

Table 18. Percentage of Total Battery Life Saved in Comparison to Pair-wise Keying (RX)

| | | Rekey Protocol | |
|---|---|---|---|
| **WSN Size** | **Key Size** | **Hierarchical** | **Secure Lock** |
| 40 | 128 | -0.06 % | 0.03 % |
| | 256 | -0.08 % | 0.03 % |
| | 512 | -0.10 % | 0.03 % |
| 100 | 128 | -0.09 % | 0.21 % |
| | 256 | -0.12 % | 0.21 % |
| | 512 | -0.16 % | 0.21 % |
| 500 | 128 | 4.31 % | 5.61 % |
| | 256 | 5.29 % | 5.59 % |
| | 512 | 7.26 % | 5.46 % |
| 1000 | 128 | 22.30 % | 22.59 % |
| | 256 | 27.40 % | 22.34 % |
| | 512 | 37.59 % | 21.84 % |

The battery power saved ranges from 0.03% to 37.59% of the batteries' total life span by using a more efficient rekey protocol instead of pair-wise.

If a WSN node has to retransmit the rekey messages, the savings are similar. Using the same approach outlined above, but with a power consumption of 17mA at the transmitting WSN node, Table 19 shows the same type of result as Table 18 for a WSN node transmitting at 3 dBm. The battery life savings in a node that transmits the rekey message ranges from 0.0333% to 39.9385%.

Table 19. Percentage of Total Battery Life Saved in Comparison to Pair-wise Keying (TX)

| | | Rekey Protocol | |
|---|---|---|---|
| **WSN Size** | **Key Size** | **Hierarchical** | **Secure Lock** |
| 40 | 128 | -0.07 % | 0.03 % |
| | 256 | -0.08 % | 0.03 % |
| | 512 | -0.11 % | 0.03 % |
| 100 | 128 | -0.10 % | 0.22 % |
| | 256 | -0.12 % | 0.22 % |
| | 512 | -0.17 % | 0.22 % |
| 500 | 128 | 4.58 % | 5.97 % |
| | 256 | 5.63 % | 5.94 % |
| | 512 | 7.72 % | 5.81 % |
| 1000 | 128 | 23.69 % | 24.00 % |
| | 256 | 29.11 % | 23.74 % |
| | 512 | 39.94 % | 23.21 % |

### 4.2.3 Memory Occupied Within the WSN Devices

Because of the limited memory in the WSN devices, the amount of memory required to store the keys or received message containing the keys is an important consideration. Table 20 presents the maximum amount of memory in bytes occupied in a WSN node during a rekey operation. Each entry represents the number of bytes occupied by the newly transmitted key(s) or the Secure Lock message.

In Table 20, for a given number of rekey recipients (cluster size) and the key size in bits, the amount of memory in bytes is shown for each rekey protocol. Pair-wise is the best performer every time, since it requires only the new SEK to be sent to each node. The number of keys held by nodes under the hierarchical rekey protocol depends on the cluster size and the implementation within the simulation. Secure Lock starts off reasonably close to hierarchical, but does not scale well [AnM08] and increases dramatically up to almost 100 times of the amount of memory occupied by pair-wise keying.

Table 20. Memory (bytes) Occupied by Rekey Message

| | | Rekey Protocol | | |
|---|---|---|---|---|
| **Cluster Size** | **Key Size** | **Pair-wise** | **Hierarchical** | **Secure Lock** |
| 4 | 128 | 16 | 80 | 64 |
| | 256 | 32 | 160 | 128 |
| | 512 | 64 | 320 | 255 |
| 10 | 128 | 16 | 96 | 158 |
| | 256 | 32 | 192 | 319 |
| | 512 | 64 | 384 | 639 |
| 50 | 128 | 16 | 128 | 794 |
| | 256 | 32 | 256 | 1593 |
| | 512 | 64 | 512 | 3194 |
| 100 | 128 | 16 | 144 | 1587 |
| | 256 | 32 | 288 | 3188 |
| | 512 | 64 | 576 | 6388 |

Table 21 reflects the same results as Table 20, but by percentage of increase over pair-wise keying. Secure Lock has the smallest increase over pair-wise for the smallest cluster size of 4 nodes, corresponding to the WSN size of 40 nodes. An improvement in the efficiency of the hierarchical assignment as mentioned in Section 4.2.1.3 would allow hierarchical to almost match (300%) or beat (200%) Secure Lock for the smallest cluster size case. For all other cases, hierarchical exhibits the least increase in occupied memory, but efficiency gains can be seen here as well. Each key eliminated from the hierarchical tree reduces the entries in the hierarchical column of Table 21 by one. Secure Lock shows a maximum increase of 9,880% over pair-wise keying. The 6.2 KB required by Secure Lock represents 4.84% of the total data memory space available in the Crossbow IRIS [Cro08], taking away memory for measurements or other data storage requirements.

Table 21. Percentage of Increase in Memory Space Used Over Pair-wise Keying

| | | Rekey Protocol | |
|---|---|---|---|
| **Cluster Size** | **Key Size** | **Hierarchical** | **Secure Lock** |
| 4 | 128 | 400 % | 297 % |
| | 256 | 400 % | 298 % |
| | 512 | 400 % | 299 % |
| 10 | 128 | 500 % | 888 % |
| | 256 | 500 % | 895 % |
| | 512 | 500 % | 898 % |
| 50 | 128 | 700 % | 4861 % |
| | 256 | 700 % | 4879 % |
| | 512 | 700 % | 4890 % |
| 100 | 128 | 800 % | 9820 % |
| | 256 | 800 % | 9861 % |
| | 512 | 800 % | 9880 % |

### 4.3 Overall Analysis

Analysis of all of the collected simulation data finds that the size of the WSN is the primary statistically-significant factor in the number of bits transmitted to rekey a WSN. Once the WSN size is accounted for, the rekeying protocol is the next largest statistically-significant contributor.

Comparing Secure Lock and pair-wise keying, Secure Lock provides the following performance gains over pair-wise keying:

- 3.32 - 75.80% reduction of the number of bits transmitted
- 0.03 - 22.59% reduction in WSN receiver battery power consumption
- 0.03 - 24.00% reduction in WSN transmitter battery power consumption

However, Secure Lock consumes 297 - 9,880% more memory than pair-wise keying.

Comparing hierarchical and pair-wise keying, hierarchical provides the following performance gains when the WSN size is 500 or 1,000 nodes:

- 57.24 - 74.83% reduction of the number of bits transmitted
- 4.31 - 37.59% reduction in WSN receiver battery power consumption
- 4.58 - 39.94% reduction in WSN transmitter battery power consumption

However, for WSN sizes of 40 or 100 nodes, hierarchical performed worse than pair-wise. For all WSN sizes, hierarchical used 400 - 800% more memory than pair-wise keying.

Even so, the security of Secure Lock is not as robust as the other keying techniques [AnM08], due to the way Secure Lock "locks" the rekey message. Each participant in the network is assigned a unique, relatively prime number as a pre-shared secret. The numbers are relatively prime to the other assigned numbers in network. Because the field of numbers is reduced from all possible combinations to only the relatively prime numbers, the strength of the lock is not as great as other encryption methods. Because of this, Secure Lock is more vulnerable to brute force attacks.

### 4.4 Summary

This chapter presents and analyzes the data collected from the Matlab simulation of three different rekeying protocols applied to wireless sensor networks using the Hubenko architecture. The validation of the simulation is presented followed by analysis of the 1440 data points collected in terms of bits transmitted and battery power conserved. An overall analysis is presented along with several observations.

# V. Conclusions and Recommendations

This chapter summarizes the conclusions of the research. Section 5.1 presents the conclusions from the experimental results. The significance of this research is discussed in Section 5.2. Section 5.3 describes recommendations for areas of future research. Finally, Section 5.4 summarizes the chapter.

## 5.1 Conclusions of Research

The following conclusions are drawn from analysis of 1,440 data points collected from 144 distinct network simulations.

The Hubenko architecture can be successfully applied to WSNs. The benefit of the Hubenko architecture over clustering alone increases as the mobility of nodes between clusters increases. Mobility does not have to be physical movement; it can also be logical movement of nodes as they are re-assigned to different clusters to realize higher efficiency or some operational effect.

The baseline Hubenko architecture rekeying performance can be improved by using either Secure Lock or hierarchical rekeying. The best choice depends on the size of the wireless sensor network and the size of the key used. For small networks with shorter key lengths, Secure Lock performs better than hierarchical. For larger networks with longer key lengths, hierarchical is the best performer. Compared to the baseline pair-wise keying, there is a 3.32% to 75.80% reduction in the number of bits transmitted for rekey operations. Table 22 summarizes the results, showing the best rekey protocol for each combination of WSN size and key size.

Table 22. Best Rekey Protocol for given WSN size and Key Size

| WSN Size | Key Size | | |
|---|---|---|---|
| | 128 | 256 | 512 |
| 40 | Secure Lock | Secure Lock | Secure Lock |
| 100 | Secure Lock | Secure Lock | Secure Lock |
| 500 | Secure Lock | Secure Lock | Hierarchical |
| 1000 | Secure Lock | Hierarchical | Hierarchical |

Significant battery power savings can be realized through the Secure Lock or hierarchical rekeying instead of pair-wise. Compared to the baseline pair-wise keying, savings range from 0.03% to 37.59% in receiver battery power saved and from 0.03% to 39.94% transmitter battery power saved.

Secure Lock places additional resource requirements on the WSN nodes. The additional memory required ranges from 297% to 9,880% over pair-wise keying. To unlock and extract the new key from a Secure Lock message, the WSN node must divide the arbitrary precision number (which is up to 51,110 bits in this simulation) by its Secure Lock ID to obtain a remainder and perform a bit-wise XOR. On the other hand, hierarchical keying's additional memory space requirements range from 400% to 800% more than pair-wise keying, and hierarchical requires only the bit-wise XOR for the WSN node to decrypt the key.

Based on the number of bits transmitted, the savings in battery power and the amount of memory space required, it is clear that hierarchical keying is the best approach for network sizes of 1,000 nodes or more utilizing a key with 128 bits or more in length. For smaller network sizes, Secure Lock may be beneficial, but any savings over hierarchical keying are offset by the reduced security and the increased complexity of Secure Lock.

## 5.2 Significance of Research

This research provides alternatives to the baseline Hubenko architecture pair-wise rekeying approach that result in valuable WSN battery savings. Since the WSN nodes' life is limited by the battery lifetime, even modest savings can be important for increasing the operational lifetime of a WSN.

The application of the Hubenko architecture to WSNs is significant. In WSNs that are mobile, or adapt to external conditions, the Hubenko architecture has been shown in previous research to reduce network rekey transmissions significantly for networks with internally mobile nodes [Phi08, Hub08].

This research is the first to apply the Hubenko architecture to WSNs. It is also the first to apply the Secure Lock and hierarchical rekey protocols to the Hubenko architecture.

## 5.3 Recommendations for Future Research

One area that requires additional work is the efficiency of hierarchical method. The initial assignment of nodes to hierarchical trees within the clusters may be worth additional communications cost to improve efficiency. Also, investigation into efficiently trimming the hierarchical tree as nodes depart, efficiently assigning nodes to the hierarchical tree to maximize efficiency may result in the hierarchical method performing better.

Another area for additional research includes using a network simulator, such as NS2 or OPNET to model physical layer effects. This would allow modeling of network retransmissions caused by a variety of reasons, based on the random placement of nodes within the physical space of the network.

In order for Secure Lock to be used in WSNs, the WSN devices must be capable of unlocking the Secure Lock message. There is no evidence that Secure Lock has been

implemented on WSN devices. A Secure Lock message to rekey 100 nodes with a 512-bit key requires 51,100 bits [AnM08]. While this will fit into the memory of the WSN devices used in this research, numbers of this size far exceed the double-precision limitation of 64 bits. One approach may be to implement Secure Lock with the Java Big.Integer class within TinyOS, to obtain the required arbitrary precision [AnM08].

Future efforts could develop a fully defined protocol based on the Hubenko architecture and one rekey protocol, such as hierarchical. The complete protocol should also handle unexpected occurrences, such as nodes not requesting a departure from the group, with the system detecting the absence and initiating a rekey automatically.

### 5.4 Summary

This chapter presented and discussed the conclusions of this research. The significance of the research is discussed as well as several recommendations for future research.

## Appendix A. Creation of Secure Locks

This appendix describes how to create and use Secure Lock to secure messages within a network.

The first step in using Secure Lock is to assign a unique, relatively prime ID number N to every node within the network [ChC89]. Relatively prime numbers are sets of numbers that have a greatest common denominator of one. Since the pool of numbers used when assigning numbers is based on the number of bits of the message size used, there may be more relatively prime numbers for a given range than true prime numbers.

The next step is to pick a subset, n, from all available nodes in the network that are authorized to properly unlock the message. Next, for each authorized recipient, the remainder used by the Chinese Remainder Theorem, $R_i$ is found through a bit-wise exclusive-or (XOR) operation between the node's assigned number, $N_i$ and the message to be sent, M as shown in [ChC89, Kob98] where

$$R_i = N_i \text{ XOR } M \tag{A1}$$

These remainders, $R_i$, are then used to create a set of equations, known as congruent equations, to solve for the common solution, which is the Secure Lock, X. The set of congruent equations is

$$X = R_i \bmod N_i \qquad \text{for all } i=1 \text{ to } n \tag{A2}$$

For this set of congruent equations, the common solution, X is somewhere between zero and M-1 where

$$M = \Pi N_i \qquad \text{for } i=1 \text{ to } n \tag{A3}$$

Based on the Chinese Remainder Theorem [Den82], the following steps [Kob98] show how

to generate the common solution, X, which is the Secure Lock [ChC89]:

a) Compute M.

b) For each i, compute $M_i = M/N_i$. Where $N_i$ is the prime ID of Node i

c) For each i, find the least positive residue, $R_i$, of $M_i$ modulo $N_i$.

d) For all i, find the least positive $y_i$ that satisfies $y_i M_i = 1 \bmod N_i$.

e) For each i, compute $R_i M_i y_i$.

f) Add all of the numbers from step (e).

g) Find the least non-negative residue modulo M of the result from (f). This is X.

Once computed, the Secure Lock, X, is transmitted to all nodes in the network.

## Appendix B. Communications Link Budget

This appendix describes the link budgets between a WSN node and a communications relay as well as a link between the communications relay and an UAV.

The transmission range of the WSN nodes is an important factor in the design of the overall network.

Using the Crossbow IRIS as the simulated device, the range depends on the output power of the IRIS, the gain of the transmitting antenna, the path loss, the gain of the receiving antenna, and finally, the receiver sensitivity.

The received power is [Ada09]

$$P_R = P_T + G_T - L + G_R \tag{B1}$$

where

$P_R$ is the received signal power in dBm

$P_T$ is the transmitter output power in dBm

$G_T$ is the transmitter antenna gain in dBi

$G_R$ is the receiver antenna gain in dBi

L is the link loss in dB

Using the IRIS's maximum power of 3 dBm [Cro08], whip antenna gains of 0 dBi [Ada09], and a receiver sensitivity of -100 dBm [Cro08, Sim08], the maximum link loss allowed is 103 dBm.

The dominate loss in a link is usually path loss, Lp is a function of the transmission frequency, F in MHz and the distance, D between the transmitter and receiver in kilometers [Ada09].

$$L_P = 32.44 + 20 \text{ Log } D + 20 \text{ Log } F \quad \text{(B2)}$$

Using a transmit frequency of 2.4 GHz and the maximum link loss of 103 dBm, the maximum distance, D is 1.41 km.

However, there are additional losses in the link. The first is the normal atmospheric loss which varies based on the distance of the link. Normal atmospheric attenuation is about 0.01 dB per km [Ada09]. The other source of loss is due to the amount of moisture in the air. If it is raining heavily, the link experiences additional losses up to 0.03 dB per km. Accounting for the normal atmospheric effects, D drops to 1.4 km.

The maximum distance of 1.4 km is the theoretical distance a link can be established between the transmitter and receiver. However, a link margin of 5 dB is normally added to ensure the link works as expected. Accounting for atmospheric effects and adding the link margin of 5 dB, the maximum theoretical distance drops to 789 meters. The IRIS specification sheet indicates a maximum range in excess of 300 meters [Cro08].

The maximum distance of 789 meters is not much range to establish communications beyond the WSN. Unless the GACA-GKM can be placed within range of the WSN, a relay must be added to the network. The relay requires a more powerful radio, which requires more battery power.

The UAVs listed in Table 3 carry the Common Data Link (CDL) radio suite, which provides VHF and UHF radio relay access to ground forces [OSD04]. L-3 Communications makes a small package (3" x 5" x 1.25") transceiver with a 3.7 W transmitter [L3C08] that is suitable for ground use. Paired with 5 dBi omni-directional antennas, the total link budget for equation (B1) is now 145 dBm with a transmission frequency of 15 GHz. The theoretical range

of this radio link is 12.2 km or 39,650 feet accounting for all factors mentioned above except for rain.

This range allows for the relay to communicate with the MQ-1 and MQ-9 UAVs operating below their maximum altitude ceiling, but is still out of reach of the Global Hawk, which commonly operates at attitudes above 40,000 feet.

Once the signal reaches the UAV, the CDL suite allows for the signal to be re-broadcast to ground stations or up to a satellite link, which allows for the network's GACA-GKM to be placed anywhere in the world.

While the equipment exists for this link to be formed, it is not a practical solution for an unattended WSN. First, the relay must be located within the range of the WSN, with a WSN node physically attached to it via Ethernet cable. While the CDL radio is small, it is still much larger than the WSN nodes. Finally, the power requirements of the radio will drain the radio's lithium battery [SLB05] within five hours of continuous transmission. Even with a 10% duty cycle on the radio, the battery will last only 2 days, compared to the 30 day simulated life of the WSN nodes.

As described in Section 1.2, one potential application of the network architecture is a replacement for a field of landmines. With the CDL communications relay(s) located on the friendly edge of the field, hundreds or thousands of WSN nodes can be scattered in front of the relays. The nodes operate as a WSN, with the cluster leaders channeling messages to the CDL communications relay. The relays can either be powered by batteries and serviced every other day, or powered by an external source. Alternatively, if the WSN field is used to protect the perimeter of friendly installation with computer network connectivity to spare, externally

powered WSN gateways can be used to provide the interface between the IP network and the WSN field.

The network modeled in this research is independent of the type of communication relay or WSN gateway used. This is left to the network designer for the specific application as required.

## Appendix C. Plots of Log Number of Bits Transmitted

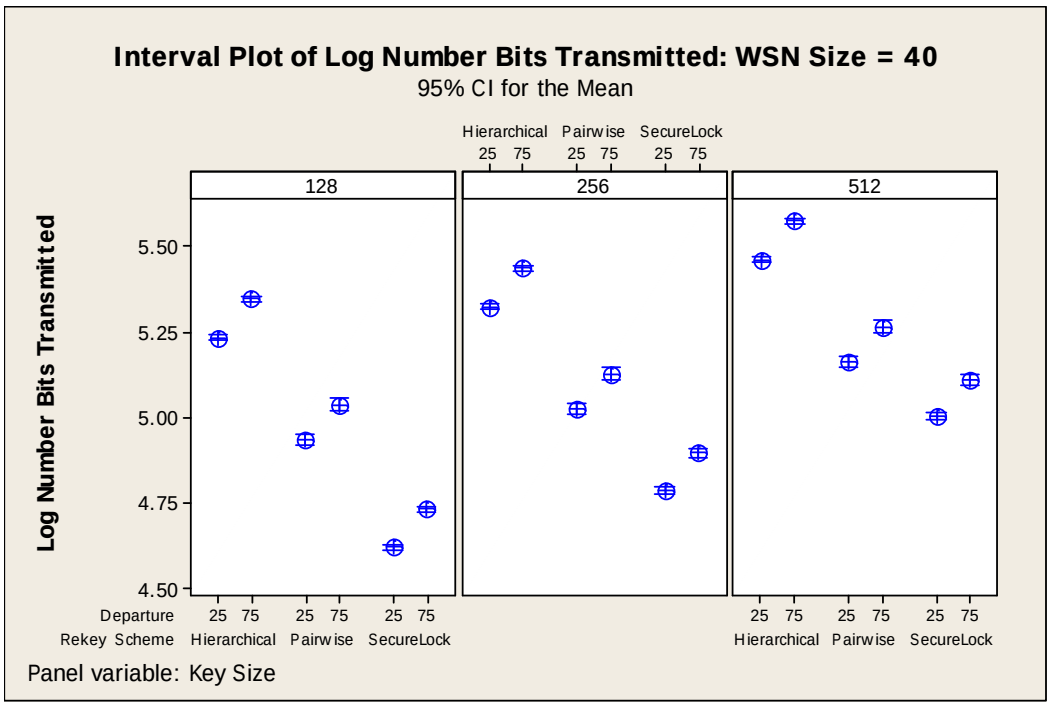


Figure 29. Plot of Log Number of Bits Transmitted for WSN Size = 40

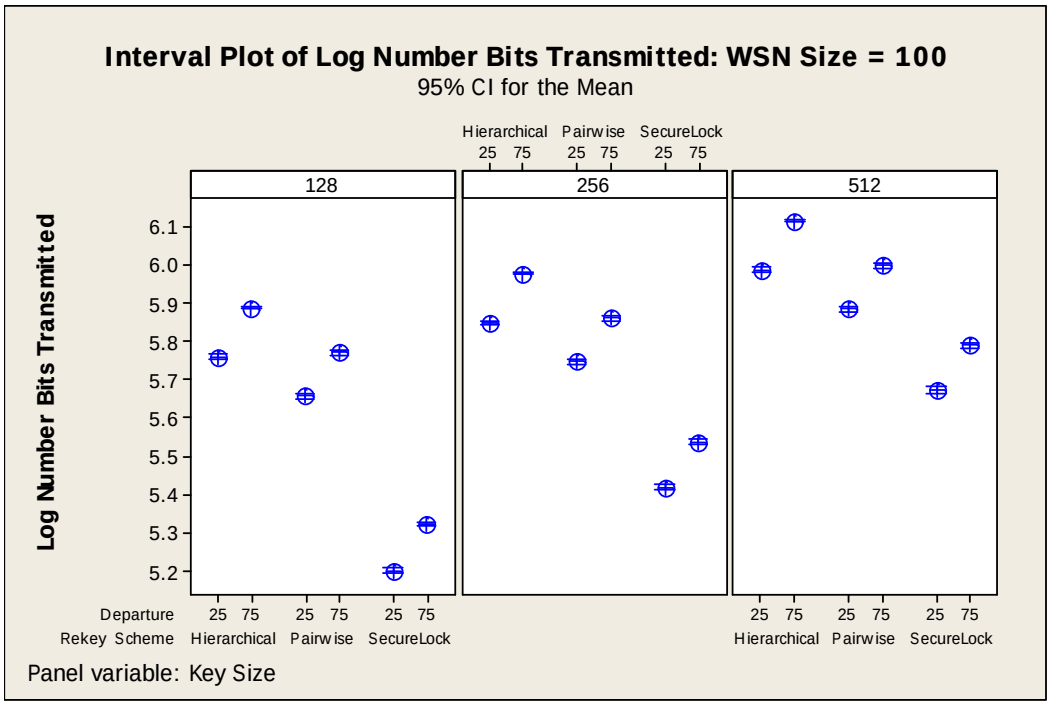


Figure 30. Plot of Log Number of Bits Transmitted for WSN Size = 100

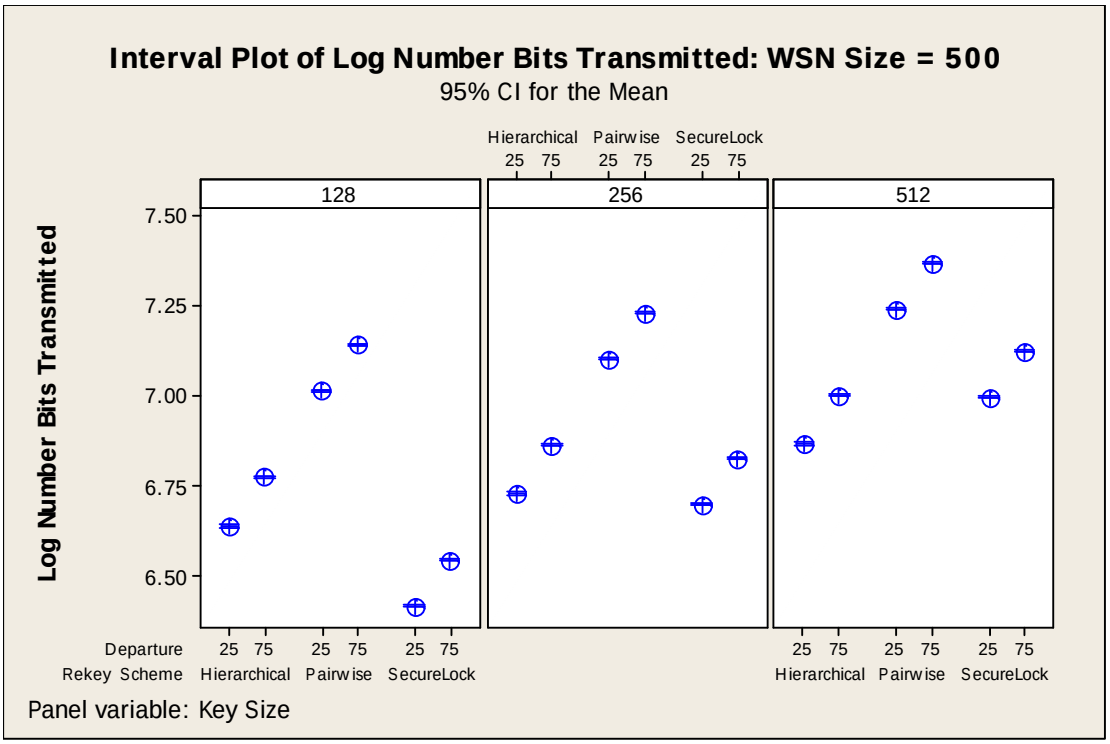


Figure 31. Plot of Log Number of Bits Transmitted for WSN Size = 500

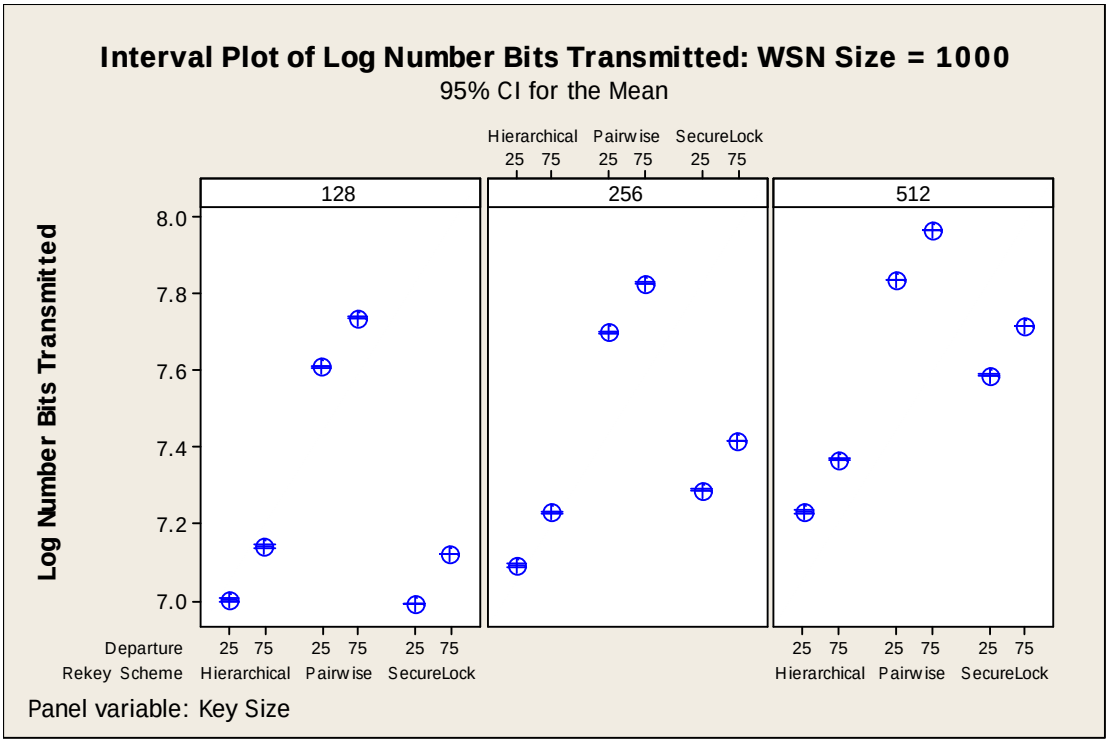


Figure 32. Plot of Log Number of Bits Transmitted for WSN Size = 1000

## Appendix D. Plots of Mean Number of Bits Transmitted

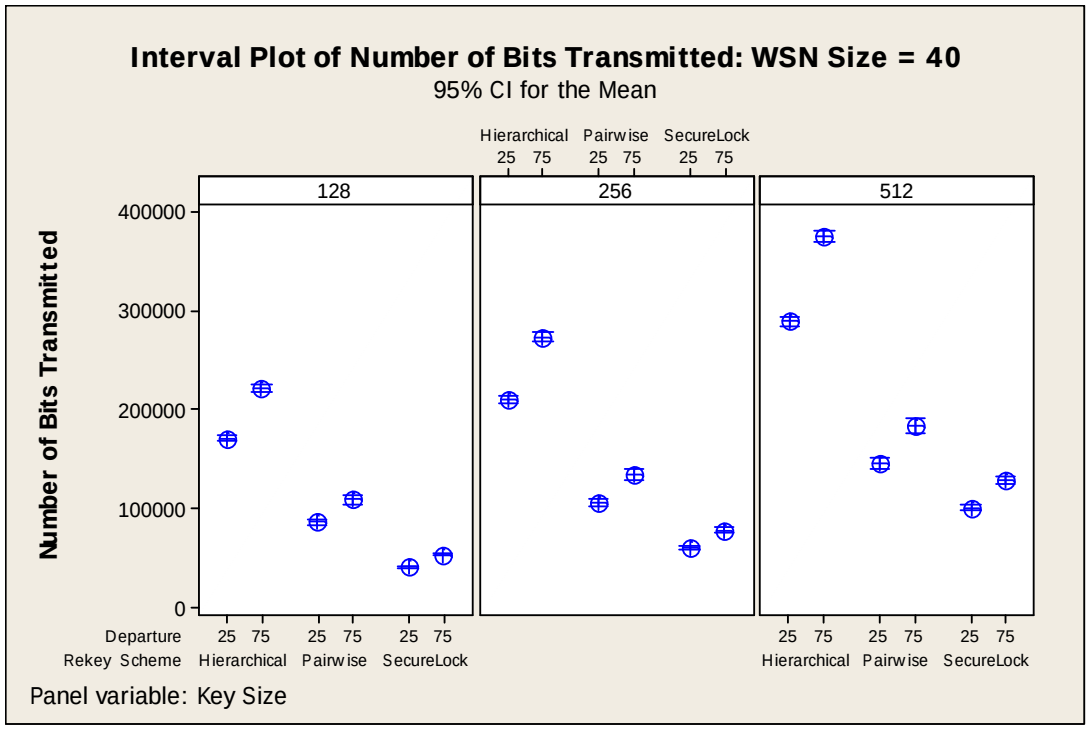


Figure 33. Plot of Number of Bits Transmitted for WSN Size = 40

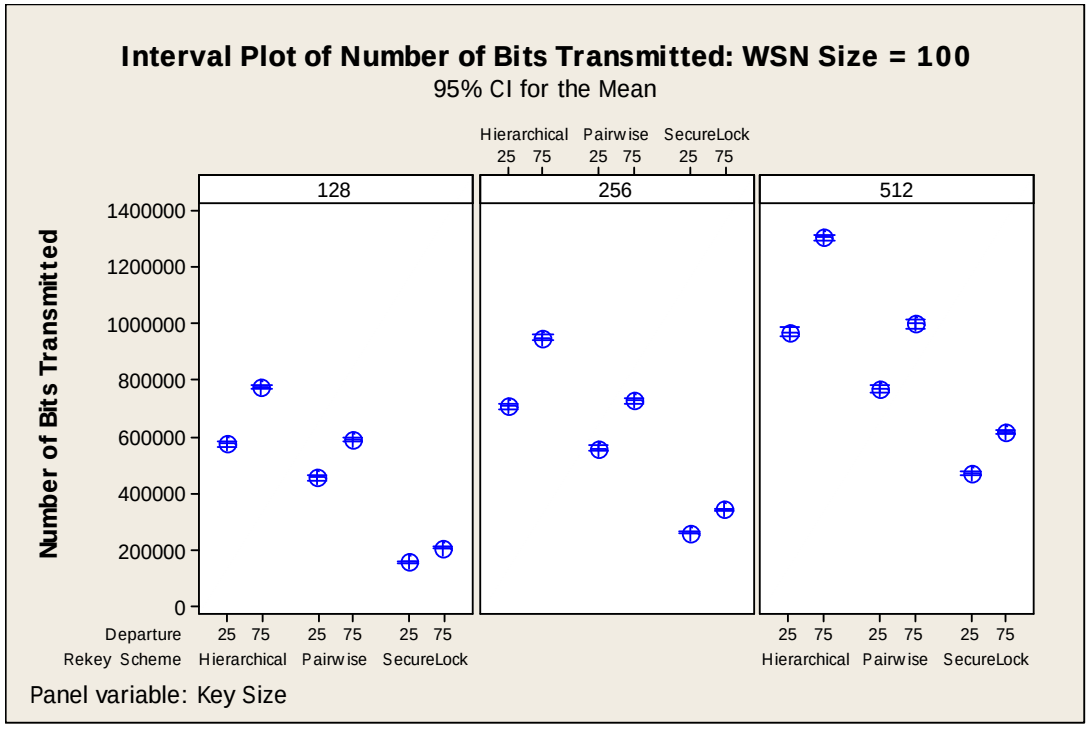


Figure 34. Plot of Number of Bits Transmitted for WSN Size = 100

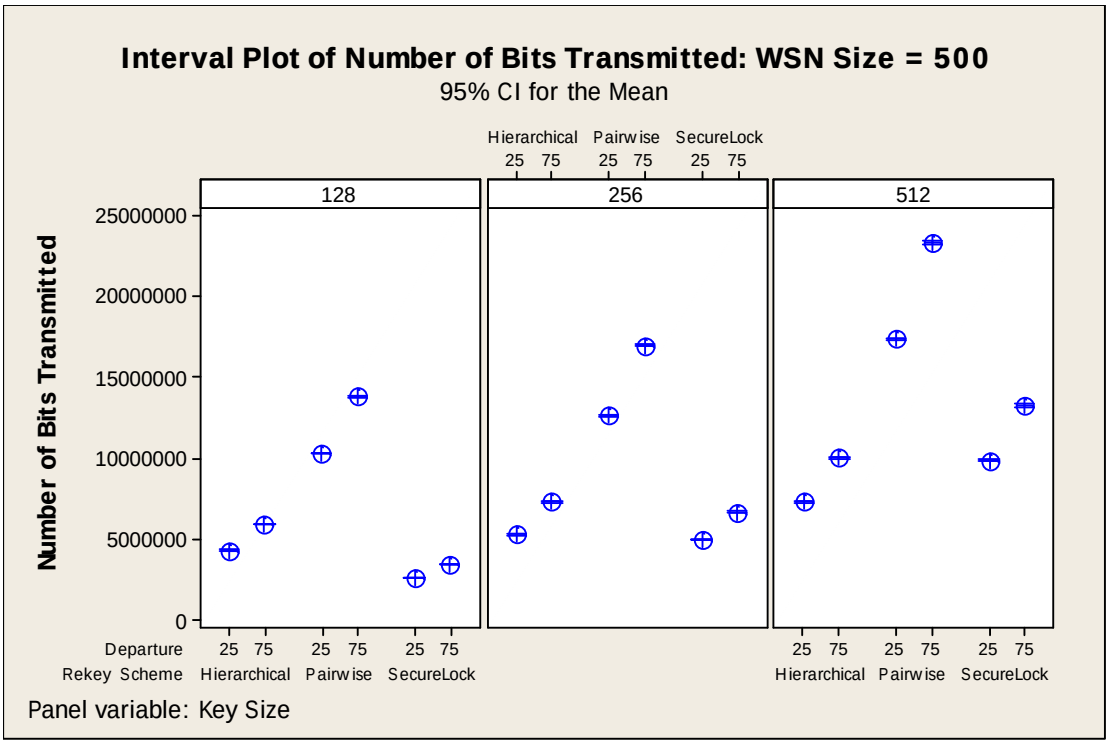


Figure 35. Plot of Number of Bits Transmitted for WSN Size = 500

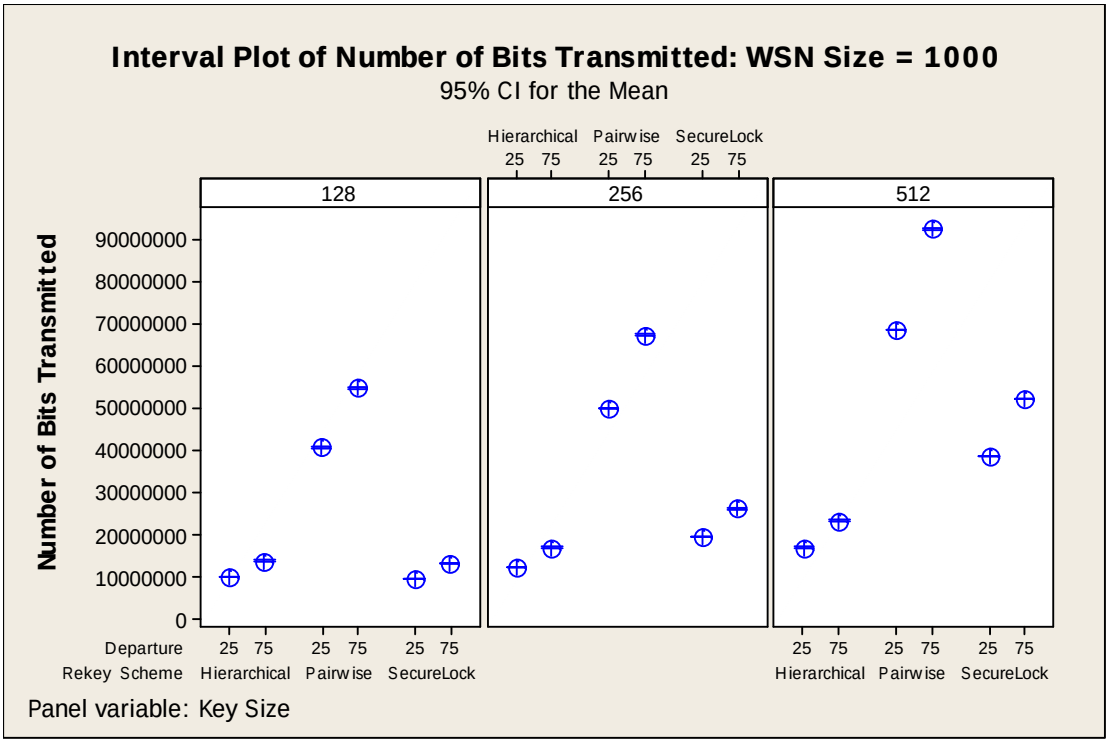


Figure 36. Plot of Number of Bits Transmitted for WSN Size = 1000

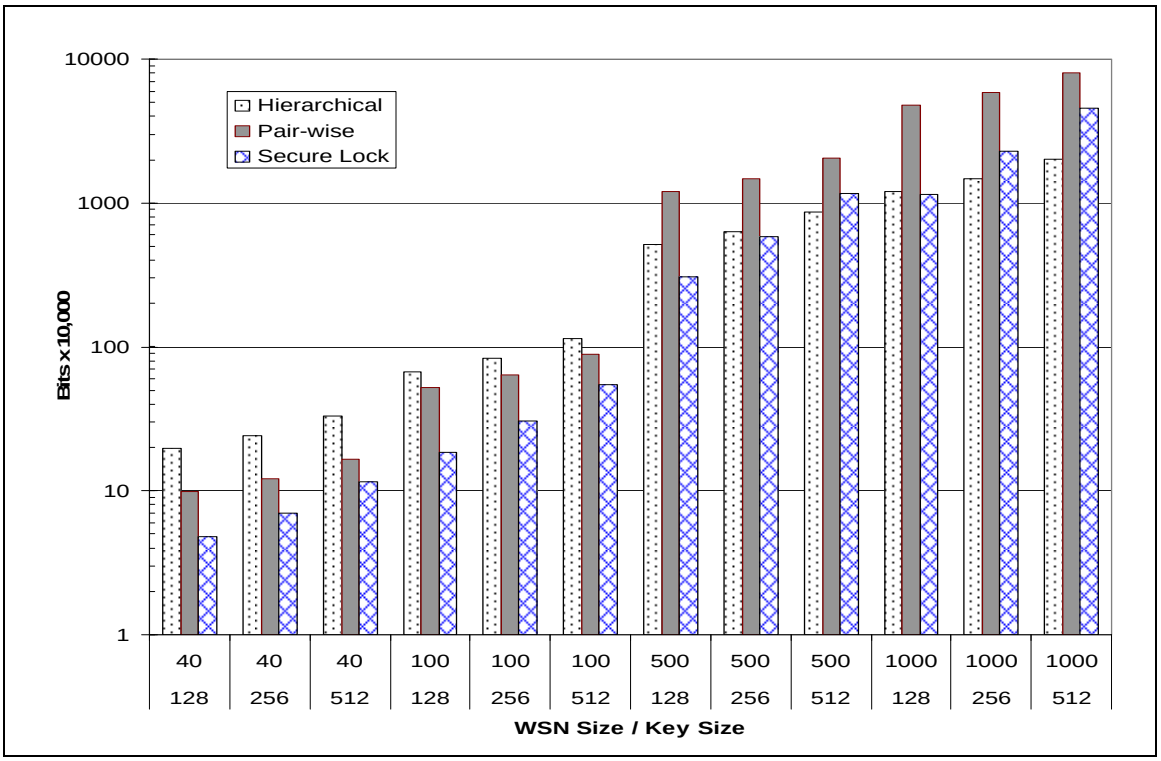


Figure 37. Plot of Data Presented in Table 15.

## Bibliography


[Ada09] D. Adamy. *EW 103: Tactical Battlefield Communications Electronic Warfare,* Boston, MA: Artech House, pp. 1-152, 2009.

[AMC07] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, pp. 32-39, December 2007.

[AnM08] C. J. Antosh, and B. Mullins, "The Scalability of Secure Lock," *Proceedings of the 27th IEEE International Performance Computing and Communications Conference,* pp. 507-512, December 2008.

[Atm08] Atmel. (2008). *ATMega 1281 Product Card.* Retrieved 2 August 2008 from http://www.atmel.com/dyn/products/product_card.asp?PN=ATmega1281.

[BaB02] S. Banerjee, and B. Bhattacharjee. "Scalable Secure Group Communication Over IP Multicast," *IEEE Journal on Selected Areas of Communciations,* vol. 20, no. 8, pp. 1511-1527, October 2002.

[CAA08] T. Claveirole, M. D. Amorium, M. Abdalla,, and Y. Viniotis, "Securing wireless sensor networks against aggregator compromises," *IEEE Communications Magazine,* vol. 46, no. 4, pp. 134-141, April 2008.

[ChC89] G. H. Chiou, and W. T. Chen, "Secure Broadcasting Using the Secure Lock." *Transactions on Software Engineering,* vol. 15, no. 8, pp. 929-934, August 1989.

[Cro08] Crossbow. (2008), *IRIS Datasheet*. Retrieved 2 August 2008 from http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/IRIS_Datasheet.pdf.

[Den82] D.E. Denning, *Cryptography and Data Security.* Reading, MA: Addison-Wesley, pp. 35-48, 1982.

[DoD08] Department of Defense. *Department of defense dictionary of military and associated terms.* Joint Publication 1-02. Washington: Headquarters, Department of Defense, 12 April 2001, as amended through 26 August 2008. Retrieved 17 September 2008 from: http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf.

[Ene08] Energizer Corp. (2008), *E91 Product Datasheet.* Retrieved 25 July 2008 from http://data.energizer.com/PDFs/E91.pdf.

[FMA08] R. Fantacci, L. Maccari, P. N. Ayuso, and R. M. Gasca, "Efficient packet filtering in wireless ad hoc networks," *IEEE Communications Magazine,* vol. 46, no. 2, pp. 104-110, February 2008.

[GAA08] General Atomics Aeronautical Systems. (2008). *Predator-B / MQ-9 Reaper: Persistent Multi-Mission ISR.* Retrieved 2 August 2008 from http://www.ga-asi.com/products/pdf/Predator_B.pdf.

[HaH03] C. E. Haave, and P. M. Haun. *A-10s over Kosovo: The victory of airpower over a fielded army as told by the airmen who fought in operation allied force*. Alabama: Air University Press, 2003.

[Hub08] V. P. Hubenko. *Secure and efficient communications for global information grid users via cooperating space assets.* PhD Dissertation, Dept of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. 2008.

[HRB08] V.P. Hubenko, R. Raines, R. Baldwin, B. Mullins. R. Mills, and M. Grimaila. "A Secure and Efficient Satellite-based Multicast Architecture," *IEEE Radio and Wireless Symposium,* pp. 227-230, 2008.

[IbM07] J. Ibriq and I. Mahgoub. "A Hierarchical Key Establishment Protocol for Wireless Sensor Networks," *21st International Conference on Advanced Networking and Applications,* pp. 210-219, 2007.

[Ian08] B. Ianotta, "Hand-held device OK'd to handle secret data," *C4ISR Journal,* vol. 7, no. 4, p 8, May 2008.

[ICB06] International Campaign to Ban Landmines. (2006). "Convention on the Prohibition of the Use, Stockpiling, Production, and Transfer of Anti-Personnel Mines and on Their Destruction," Retrieved 14 November 2008, from: http://www.icbl.org/content/download/7050/165094/file/treatyenglish.pdf.

[L3C04] L 3 Communications Corp. (2004). "The Battlefield Anti-Intrusion System, AN/PRS-9," Retrieved 10 November 2008 from http://www.l-3com.com/cs-east/pdf/bais.pdf.

[L3C04b] L 3 Communications Corp. (2004). "The Remotely Monitored Battlefield Sensor System-II, AN/GSR-8(V)," Retrieved 10 November 2008 from http://www.l-3com.com/cs-east/pdf/rembassii.pdf.

[L3C08] L 3 Communications Corp. (2008). "Miniature CDL Transceiver (Mini-CDL-200)," Retrieved 10 January 2009 from http://www.l-3com.com/csw/Product/docs/MiniCDLTransceiver.pdf.

[Jai91] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Littleton, MA: John Wiley. 1991.

[Jez08] A. Jeziorski, "BAMS to the finish line: Three-way race will shape US maritime surveillance," *C4ISR Journal,* vol. 7, no. 4, pp. 16-17, May 2008.

[JuA02] P. Judge, and M. Ammar, "Gothic: A group access control architecture for secure multicast and anycast," *IEEE Infocom,* pp. 1547-1556, July 2002.

[KMB07] J. T. Kautz, B. E. Mullins, R. O. Baldwin, S. R. Graham, "An adaptable energy-efficient medium access control protocol for wireless sensor networks," *Proceedings of the 40th Annual Hawaii International Conference on System Sciences,* 2007.

[Kob98] N. Koblitz, *Algebraic Aspects of Cryptography*, 2nd Ed, Berlin Heidelberg, Germany: Springer Verlag, pp. 27-33, 1998.

[Kru98] P. Kruus, "A Survey of Multicast Security Issues and Architectures," *Proceedings of the 21st National Information Systems Security Conference*, pp. 503-511, 1998.

[KuR05] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet* (3rd Edition). Boston: Pearson Education, 2005.

[MOV96] A. Menezes, P. V. Oorschot, and S. Vanstone, *Handbook of applied cryptography*. United States of America: CRC Press, 1996.

[MDM07] J. V. D. Merwe, D. Dawoud, and S. McDonald, "A survey on peer-to-peer key management for mobile ad hoc networks," *ACM Computing Surveys,* vol. 39, no. 1, Apr 2007.

[Mit97] S. Mittra, "Iolus: A framework for scalable secure multicasting," *Proceedings of the ACM SIGCOMM,* pp. 1-12, Sep 1997.

[MRR99] M. J. Moyer, J. Rao, and P. Rohatgi, "A survey of security issues in multicast communications," *IEEE Network*, pp. 12-23, Nov/Dec 1999.

[NoG07] Northrop Grumman. (2007). *HALE factsheet.* Retrieved 1 June 2008, from http://www.is.northropgrumman.com/systems/system_pdfs/HALE_Factsheet.pdf.

[OSD04] Office of the Secretary of Defense. (2004). "Defense Science Board Study on Unmanned Aerial Vehicles and Uninhabited Combat Aerial Vehicles." Retreived on 1 April 2008 from: http://www.acq.osd.mil/dsb/reports/uav.pdf.

[PPS08] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, J. Hubaux, "Secure neighborhood discovery: A fundamental element for mobile ad hoc networking," *IEEE Communications Magazine,* vol. 46, no. 2, pp. 132-139, Feb 2008.

[Per08] C. Perrin. (2008). *How to Spoof a MAC Address.* Retrieved 28 September 2008, from http://blogs.techrepublic.com.com/security/?p=395.

[PST02] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *Wireless Networks,* vol. 8, no. 5, pp. 521-534, 2002.

[Phi08] A. Phillips. *A secure group communication architecture for a swarm of autonomous unmanned aerial vehicles.* MSECE Thesis, Dept of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. 2008.

[SLB05] Saft Lithium Battery Group. (2005, Dec). "BA 5590 High Capacity Lithium Sulfur Dioxide Primary Battery System." Retrieved 10 January 2009 from: http://www.saftbatteries.com/doc/Documents/primary/Cube536/BA5590_HC_1205.370233fe-c19d-42c9-bab6-694ed6a067e5.pdf.

[SaM00] L. H. Sahasrabuddhe, and B. Mukherjee. "Multicast routing algorithms and protocols: A tutorial," *IEEE Network*, pp. 90-102, Jan/Feb 2000.

[Sim08] SimPhonics, Inc. (2008) "VComm User's Manual, V1.25", Retrieved 10 January 2009 from: http://www.simphonics.com/products/software/VComm/Docs/VComm%20User%20Manual.doc.

[Sch08] E. Schechter, "Rush to the sky: US army quickens the pace of its new UAV program," *C4ISR Journal,* vol. 7, no. 5, pp. 24-25, Jun 2008.

[Ste94] W.R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols.* Reading, Massachusetts: Addison-Wesley-Longman Inc, pp. 169-186, 1994.

[SuR07] Y. Sun, and K. J. R. Liu. "Hierarchical Group Access Control for Secure Multicast Communications," *IEEE/ACM Transactions on Networking,* vol. 15, no. 6, pp. 1514-1526, 2007.

[TGC08] Teal Group Corporation. (2008). *World unmanned aerial systems: Market profile and forecast*. Retrieved 1 June 2008 from http://www.tealgroup.com/.

[Ten01] D. Tennenhouse. (2001). "Largest Tiny Network Yet," retrieved 14 November 2008 from: http://webs.cs.berkeley.edu/800demo/.

[Tho02] S.A. Thomas. *IP switching and routing essentials: Understanding RIP, OSPF, BGP, MPLS, CR-LDP and RSVP-TE.* United States of America: John Wiley and Sons, pp. 61-192, 2002.

[USA05] United States Air Force. (2005) *The U.S. air force remotely piloted aircraft and unmanned aerial vehicle strategic vision,2005*. Retrieved 3 April 2008 from http://www.af.mil/shared/media/document/AFD-060322-009.pdf.

[USA08] United States Air Force. (2008). *Factsheets.* Retrieved 1 April 2008, from http://www.af.mil/factsheets/.

[WAR06] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials,* vol. 8, no. 2, pp. 2-36, 2006.

## Vita

Captain Cory J. Antosh was born in Racine, Wisconsin in 1970 and graduated from Canyon Del Oro High School in Tucson, Arizona in 1988. After graduation, he enlisted in the US Air Force and was trained as a Secure Communications Systems Maintainer. He maintained voice and data communications equipment while assigned to fixed and mobile communications units at Seymour Johnson AFB, North Carolina; Osan Airbase, South Korea; and Eglin AFB, Florida. In 1997, he was selected for the Airman's Education and Commissioning Program, which sent him to the University of Arizona to study Electrical Engineering, graduating in December 2000. While at the University of Arizona, he was inducted into Eta Kappa Nu and also became a member of the IEEE. After graduation, he attended Officer's Training School at Maxwell AFB, Alabama and was commissioned in April 2001.

After commissioning, his first assignment was to the 31st Communications Squadron, Aviano AB, Italy, where he was placed in charge of the base computer network and later became the Plans and Programs Flight Commander. In February 2003, he was selected to be the 31st Mission Support Group Executive Officer. In April 2004, he was reassigned to the Air Education and Training Command's Studies and Analysis Squadron at Randolph AFB, Texas, where he oversaw operational testing of new training technologies as the Technology and Innovation Flight Commander. In 2006, Capt Antosh deployed to Camp Victory, Iraq where he planned and oversaw the restoration of Iraqi civilian and military communications while assigned to the Multi-National Corps-Iraq. In August of 2007, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the United States Air Force Academy in Colorado to be an instructor of electrical engineering.

# REPORT DOCUMENTATION PAGE

*Form Approved OMB No. 074-0188*



**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From – To)* |
|---|---|---|
| 26-03-2009 | Master's Thesis | August 2007 – March 2009 |

**4. TITLE AND SUBTITLE**

The Evaluation of Rekeying Protocols Within the Hubenko Architecture as Applied to Wireless Sensor Networks

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**5. AUTHOR(S)**

Antosh, Cory J., Captain, USAF

**5d. PROJECT NUMBER**
09-310

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)**

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFIT/GE/ENG/09-04

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

United States Air Force Academy, UAV Research Group
Contacts: Daniel Pack, Ph.D.
Mailing Address: USAFA/DFEC, 2354 Fairchild Drive, Suite 2F6
USAF Academy, Colorado Springs, CO 80840
Email: Daniel.Pack@usafa.edu
Phone: (719) 333-6967 DSN: 333-6967 Fax: (719) 333-3756

**10. SPONSOR/MONITOR'S ACRONYM(S)**
USAFA/DFEC

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

This thesis investigates the impact of using three different rekeying protocols–pair-wise, hierarchical, and Secure Lock within a wireless sensor network (WSN) under the Hubenko architecture. Using a Matlab computer simulation, the impact of the three rekeying protocols on the number of bits transmitted across the network and the amount of battery power consumed in WSN nodes during rekey operations is investigated. Baseline pair-wise rekeying performance can be improved by using either Secure Lock or hierarchical rekeying. The best choice depends on the size of the WSN and the size of the key used. Hierarchical rekeying is the best choice for networks with 500 or more nodes using a key size of 512 bits. It is also the best choice for a network of 1,000 nodes using a 256-bit key. For smaller networks with shorter key sizes, Secure Lock is the best choice. Overall, the number of bits transmitted for rekey operations can be reduced 3.32% to 75.80% and the battery power savings range from 0.03% to 39.94% compared to pair-wise keying. Based on the number of bits transmitted, the savings in battery power and the amount of memory required, hierarchical keying is clearly the best approach for network sizes of 1,000 nodes or more utilizing a key with 128 bits or more in length. For smaller network sizes, Secure Lock can be beneficial, but any savings over hierarchical keying are offset by the weaker security scheme and increased complexity of Secure Lock.

**15. SUBJECT TERMS**
**wireless sensor network, group key management, security, multicast**

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **REPORT** | **ABSTRACT** | **c. THIS PAGE** | | | Barry E. Mullins, Ph.D. (ENG) |
| **U** | **U** | **U** | **UU** | 104 | **19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 6565; e-mail: Barry.Mullins@afit.edu |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18