SECURE ROUTING AND BROADCAST AUTHENTICATION IN HETEROGENEOUS SENSOR NETWORK

By Aasma Abid

MS thesis submitted to faculty of Information Security Department Military College of Signals, National University of Sciences and Technology, Pakistan in partial fulfillment of the requirements for the degree of MS in Information Security

January 2011

ABSTRACT

Security of sensor network communication architecture relies on its routing scheme. Previously security was not issue of routing protocols. But now security is adequate for routing. Homogeneous sensor networks are less efficient due to limitations of scalability and are prone to routing attacks due to resource constraints. Heterogeneous sensor networks have proven to be more secure and scalable. Sensor network used is heterogeneous. Routing tables are generated with multipath routing. Base station generates inter-cluster routes and cluster head generates intra-cluster routes. This minimizes the computation load on cluster nodes. Light-weight broadcast authentication is used for secure routing table generation and data communication; and to reduce communication overhead. Goal of proposed scheme is to introduce security architecture with in routing protocol. Security analysis shows that scheme is secure against routing attacks and is tolerant to compromised nodes. Moreover damage due to compromised nodes is only confined locally.

DEDICATION

This research work is dedicated to my parents, for their never ending love, faith and motivation, especially to my husband and my little daughter. I would also take the opportunity to thank my in-laws for their constant support and encouragement.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

ACKNOWLEDGEMENTS

First of all, all Praises to Almighty Allah for His Blessing and Mercy. Indeed, His Fovour was the most important source for the completion of the present thesis.

I would like to thank my Advisor Brig (R) Dr.Mukhtar Hussain for his time, guidance related to present work. In fact this thesis is an out come of his concern that leads me to the completion of this research work. He has been very cooperative throughout the thesis work.

More over I am also thankful to Dr.Firdous Kausar whose guidance lead me to publish my research paper in 12th International conference on Network Based Information System NBiS at Perdue University, Perdue USA.

I am also thankful to all of the guidance committee members, Lec Ayesha Naureen, Lec Ahmad Raza Cheema and Lt Col Dr M. Arif Wahla, they have been very helpful.

TABLE OF CONTENTS

List of Figures

List of Tables

INTRODUCTION

1.1. Overview

Sensor networks are becoming viable solution to many challenging problems, and security issues pertaining to wireless sensor networks are in lime light. HSN, Heterogeneous sensor networks are considered excellent visions in applications like monitoring, tracking and distributed sensing at sensitive borders of countries in military and civil environment. HSNs are more scalable than homogeneous wireless sensor networks. Due to resource limitation; strong cryptographic algorithms for encryption, decryption and authentication in routing protocols cannot be deployed on individual sensor nodes [1, 2, 3]. For this reason HSN is one of better option, because sensors are heterogeneous in terms of the resources they posses. HSN increases network reliability. Heterogeneity can triple the average delivery rate and lifetime of large battery powered sensors network. HSN work in clusters, therefore improves energy efficiency of the network [4-5] but security protocol for HSN are just few.

Secure routing and data forwarding is essential service for enabling communication in sensor networks. Unattended and wireless nature of the network makes it vulnerable to multiple attacks. Unfortunately current routing protocols suffer from many security vulnerabilities like eavesdropping, broadcast authentication, DoS style attacks, and malicious nodes injecting spurious information, resulting in routing inconsistency and susceptible to replay attacks. [6]. Unshielded sensor network leaves nodes vulnerable to physical compromise; adversary may capture nodes, analyze them and replicate those nodes, placing these replicas in strategic locations within network secretly to initiate malicious activity.

Problem under consideration is resource limitation of wireless sensor network, due to which heterogeneous nodes are introduced in the network. We used the approach of hierarchical HSN.

Intrusion detection in the HSN is in fact difficult task to be performed in limited time. Because the parameter required for anomaly-based intrusion detection are not known prior to nodes. Concluding these is time consuming which may lead to demolishes whole network.

Proposed scheme uses multi-path routing, to bypass the intruders. Multiple routes are discovered between each sender and receiver. Further, the broadcast nature of the wireless communication medium significantly enhances the capabilities of an intruder for DoS style attack, by repeatedly sending same message or advertises bogus routing information. Challenge is to propose a secure routing scheme for static hierarchical HSN with the risk of physical security and resource limitations; and can provide light weight broadcast authentication.

1.2. Problem Statement

Sensor networks suffer from routing attacks, due to limitation of sensor nodes resources, and risk of physical security. Secure routing and data forwarding is core concern in civil and military applications of sensor networks. Due to resource limitations; strong cryptographic algorithms for encryption, decryption and authentication in routing protocols cannot be deployed on individual sensor nodes. Heterogeneous sensor networks show better performance because sensors are heterogeneous in terms of the resources they posses. Broadcast authentication is probably one of the most critical security primitive in sensor networks, Extensive research have used only symmetric primitives to achieve light weight broadcast authentication in sensor networks.

1.3. Objectives

Objective of this research work is to have a critical analysis of existing routing schemes being secure with the limitations of sensor networks. Focus of research is to propose secure routing with light weight broadcast authentication scheme in heterogeneous sensor network environment. Research work also provides analysis and comparison of proposed scheme with the existing routing scheme.

1.4. Organization of Research Work

This research work has been divided in chapters. Proceeding chapter 2 will discuss Sensor network environment, limitations of homogeneous sensor networks. Why heterogeneous sensor network environment been chosen, routing protocols, limitations of existing routing protocols. Chapter 3 presents our proposed secure routing and broadcast authentication scheme. Chapter 4 discusses design of our proposed scheme how it is been implemented on TinyOS, flow graph of the scheme and working scenarios. Then actual implementation of proposed scheme using TinyOS simulation results is included in Chapter 5 and performance and security analysis is discussed. Chapter 6 presents, concluding remarks along with future work in this research area.

LITERATURE SURVEY

2.1 Introduction

Overall literature that has been reviewed for this thesis research work has been included in this chapter. Including Wireless sensor networks, their application, limitations, security architectures, routing protocols and the most important one is the secure routing protocol that has been used for comparison.

Wireless Sensor Network (WSN) was emerged from the advancement of integration between tiny embedded processors, wireless interfaces and micro-sensors which were based on MEMS. WSNs comprises of large number of heterogeneous sensor devices. Network consists of complex sensor nodes with the capabilities of storage, processing and communication. These nodes have ability to monitor the physical environment through adhoc deployment of numerous tiny nodes networked together intelligently. [7]

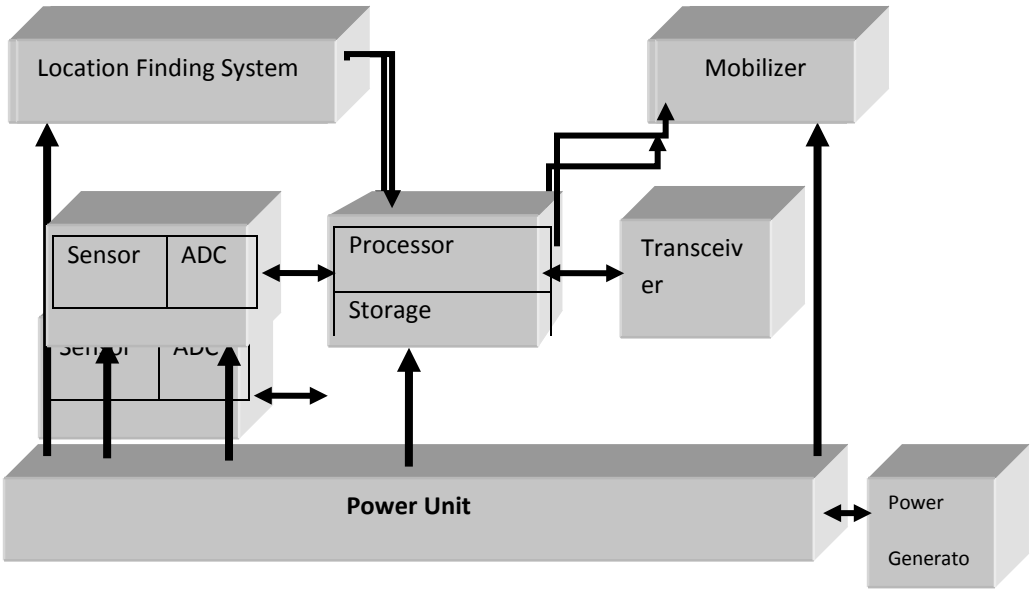

Figure: 2.1 Sensor Node Components

2.2 Applications of Wireless Sensor Network

These can be used in structural health monitoring, Wireless sensing on machines will allow assets to be inspected when the sensors indicate that there may be a problem, reducing the cost of maintenance and preventing catastrophic failure in the event that damage is detected. It will also reduce the initial deployment costs, as the cost of installing long cable runs is often prohibitive [8]. Wireless sensors are used for industrial automation; use of wireless sensors allows for rapid installation of sensing equipment and allows access to locations that would not be practical if cables were attached. Examples include health monitoring, environment monitoring, location-based services for logistics, and health care.

2.3 Design Challenges

WSN is challenging and unique from research point of view due to the limitations of energy constraint there is trade off between performance and life time, remote deployments lead to self healing and self-organizing, scalability leads to large number of nodes. One of the design issues is heterogeneity because nodes or devices are of varied capabilities; sensors have different modalities and hierarchical deployments. Another issue is of adaptability i.e. adjusting to operating conditions and changes in application requirements.

But the most important design issue is security and privacy, because sensor networks are used for potentially sensitive information and these networks are working in hostile environment. [9]

2.4 Security Requirements

Designing a secure sensor network communication protocol is hard; as it requires data secrecy, availability, authentication, and replay protection. With the limitations of sensor i.e. low energy consumption.

2.5 Existing Security Architecture

Security architecture for secure communication analyzed during research is TinySec, SPINS, and the most recent one MiniSec [10, 15, 16].These security architectures suffer from different vulnerabilities.

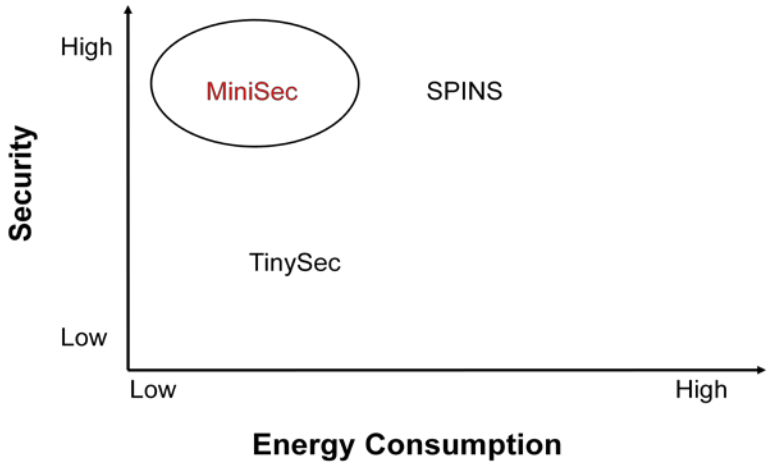

Table 1: Limitations of Different Security Architecture

	Pros	**Cons**
TinySec	No counter resynchronization No stored state	Packet overhead No replay protection
SPINS	No packet overhead Replay protection	Counter resynchronization O(*n*) state
MiniSec	No packet overhead Implicit counter resynchronization	Loose time synchronization

	Constant state Probabilistic replay protection	

TinySec being the most popular secure link layer protocol. It was designed for Mica2 motes in 2003 when memory constraints were much more severe issue then they are today. Now currently developed motes have increased memory size but energy constraints remains as issue as ever. TinySec provides authentication and data integrity with low power consumption but does not provide replay protection. Utilization of single network wide key, which leads to complete network compromise even if single node is compromised.

SPINS has two phases SNEP and µTESLA. Security issues like confidentiality, data authentication and data freshness or replay protection is provided by SNEP. Most important security issue is broadcast authentication and it is provided by µTESLA. SPINs being more secure architecture than TinySec, suffer from high energy consumption, and counter resynchronization. SPINS provide replay protection at the expense of storing per sender state; limits network scalability.

MiniSec is network layer architecture with low energy consumption and high security mechanisms providing data secrecy, authentication and replay protection. Most important aspect of MiniSec is broadcast authentication and replay protection using Bloom Filters and rate control mechanism. Also provide implicit counter resynchronization. Counter value is not appended to packet decreasing packet overhead [10].

SPINS introduced secure broadcast authentication using µ*TESLA* [15] and data secrecy using Message authentication code MAC; calculated using keyed one way hash functions. Our scheme makes use µ*TESLA* from SPINS and Bloom filters from MiniSec [10] for broadcast authentication and replay protection during routing. MAC is used for verification of local connectivity information transferred to base station.

2.6 Sensor network routing architecture

Secure routing is demand of secure data communication. Route computation for routing protocols may be on-demand or prior. On-demand routing protocols computes routes

when required by the node, it may suffer from the packet loss of discovered routes when required, and nodes cannot forward messages till the time routes are discovered. It may add unusual delay. Similarly prior route computation has overhead of pre-route discovery, many of those routes may not be used; also route maintenance is required continuously [3]. Our scheme utilizes prior route computation, computing routing tables before data forwarding.

2.7 Heterogeneous sensor network

Previous sensor network platforms like mica and mica2 use non-standard, platform specific radios, problem is that these two types are not interoperable with one another. Most of the existing sensor networks have homogeneous nodes. IEEE approved the 802.15.4 radio MAC (Medium Access Control) and a physical layer standard, in 2003. These standards are designed explicitly for Low Rate Wireless Personal Area Networks (LRWPANs).

Now there are new sensor network platforms like micaz and telos and they support this new radio standard, and allow these heterogeneous nodes to communicate with each other. With this now hierarchical heterogeneous sensor networks are designed. This leads to improved scalable, flexible and long life sensor network. Also many sensor network designers already require some degree of heterogeneity for development and testing in simulated frameworks such as the TinyOS Simulator and EMStar [17].

we are considering Tmotes in our research these are Low Power Wireless Sensor Module, TelosB is IEEE 802.15.4 compliant, having 250 kbps, High Data Rate Radio , TI MSP430 microcontroller with 10kB RAM, integrated onboard antenna , Data collection and programming via USB uses Open-source operating system like TinyOS with Optional integrated temperature and humidity sensor. These are developed and published to the research community by UC Berkeley. [18]

2.8 Analyzed Routing Protocols and their Limitations

Abu-Ghazaleh et al [11] have proposed geographic routing protocol with multipath routing, tolerate packet DOS attack for packet dropping, on the basis of reputation of one-hop neighbors. Misbehaving neighbor nodes are avoided by discovering new paths to sink using reputed and verified locations information. A node only needs to store geographic location of one hop neighbors and trust factor. Although geographic routing [12, 13] seems to be very

attractive from the point of scalability of sensor network and storage efficiency, but require each node to be aware of its geographic location and require GPS with extra power and additional hardware, and doesn't work indoors. Therefore this scheme suffers from single path routing. Our scheme use multipath routing mechanism between same sender and receiver for hierarchical HSNs.

Another secure routing protocol is ARRIVE [2]. This algorithm has robust routing in WSNs, follow tree based topology. It works on localized observation of neighbor nodes for packet forwarding decision and nodes forward message to its parent and all of the neighbor nodes with threshold value of higher reputation [14].

Yet another secure routing protocol is INSENS (Intrusion Tolerant Routing Protocol for WSNs) construct routing tables for every node, minimizes storage, computational and communication on sensor nodes. It works in phases of pre-deployment, route discovery and data forwarding. But this protocol has several drawbacks like high overhead due to secure route discovery phase. All node to node communication is via base station. Increases burden on base station. And it increases chances of single point of failure incase base station is compromised. Due to which network is not scalable [1]. Our scheme proposes routing for hierarchical HSN with high power nodes for computations and data transfer with in clusters and low power nodes for sensing. Therefore computational burden on base station reduces.

2.9 Routing Attacks

Various kinds of attacks on sensor network areas follow:

- Spoofed, altered or replayed routing attack.
- Selective forwarding
- Sinkhole attacks
- Sybil attacks
- Wormholes
- HELLO flood attack

2.10 Development Environment

The power of TinyOS coupled with NesC on TOSSIM simulator is utilized to develop proposed security architecture for HSNs. This platform has been particularly selected because of its capability of being tested on real sensor networks test bed and real time demonstration on

sensor nodes. TinyOS started as collaboration between the University of California, Berkeley in co-operation with Intel Research and Crossbow Technology**,** and has since grown to be an international consortium, the TinyOS Alliance.

2.10 Overview

This Chapter gives an overview of the wireless sensor networks, challenges faced by wireless devices and wireless sensor network. Further security architectures and their pros and cons are reviewed in detail. Existing secure routing protocols are reviewed. INSENS is the secure routing protocol, which is used for comparison with our scheme.

PROPOSED ROUTING SCHEME

3.1 Introduction

First problem under consideration is resource limitation of wireless sensor network, due to which heterogeneous nodes are introduced in the network. All heavy computations are performed on the cluster head being more powerful nodes. Instead of having centralized base station being the only point of computation and communication, more over single point of failure if compromised, we used the approach of hierarchical HSN.

Intrusion detection in the HSN is in fact difficult task to be performed in limited time. Because the parameter required for anomaly-based intrusion detection are not known prior to nodes. Concluding theses is time consuming which may lead to demolishes whole network.

Proposed scheme uses multi-path routing, to bypass the intruders. Multi path routing have multiple advantages, it provide reliability i.e. the probability that a message generated at one place in the network can actually be routed to the intended destination. It reduces Routing Overhead and provides Security.

Multiple routes are discovered between each sender and receiver. These paths are independent of each other i.e. common nodes between same sender and receivers are as least as possible. And message is forwarded on multiple routes, to avoid route jamming due to compromised node. First path is calculated using Shortest Path tree (SPT). Nodes can have second or third path using other cluster members and through other CH to reach its CH.

Third, the broadcast nature of the wireless communication medium significantly enhances the capabilities of an intruder for DoS style attack, by repeatedly sending same message or advertises bogus routing information. Bloom filters are used for the purpose of broadcast authentication and replay protection [10].

3.2 Network Model

Heterogeneous sensor network is under consideration. Network includes BS and two types of nodes working in clusters. These nodes are different on the basis of resources they posses. Small number of high power nodes working as cluster head, CH and large number of low power nodes working as cluster nodes CN. CHs have large storage, communication and computational capacity. Cluster members have limited storage, communication and computational capacity. It is assumed that these nodes are deployed uniformly and randomly in sensor network. Figure 3.1 shows network model with star gate used as CH and Tmotes used as cluster members. Nodes use multi-hopping to communicate with the closest CH rooted toward BS.

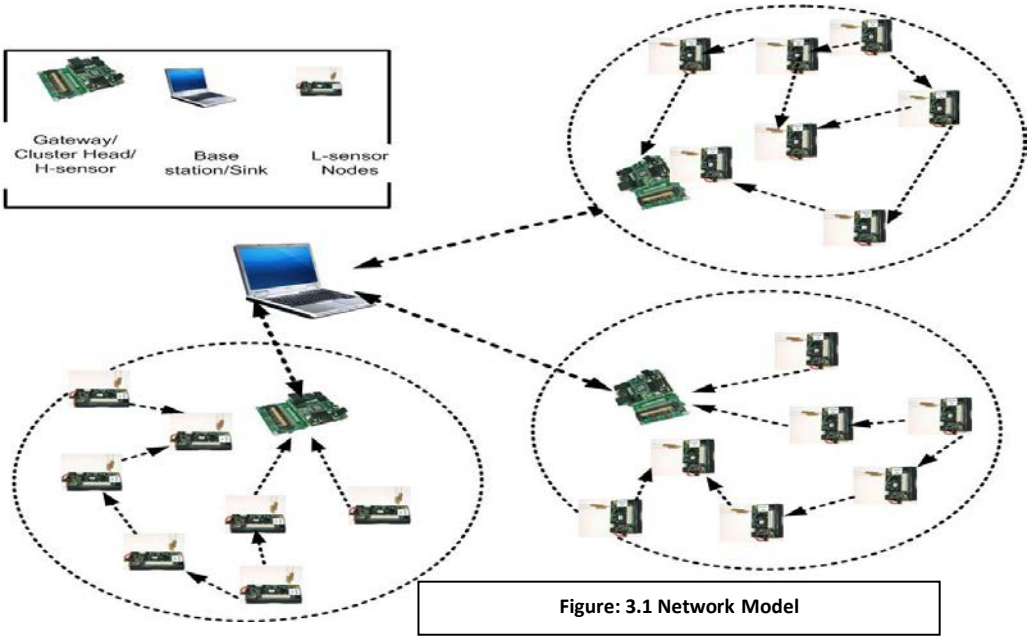

Figure: 3.1 Network Model

CN can communicate with each other after routing table generation and sharing pair wise keys. Due to these, cluster formation data is processed locally and reduces communication load and computation load on centralized BS. It is assumed that CHs can communicate directly with other CHs. It is assumed that BS shares unique secret key with each CH and CH are pre-loaded with keys that they shares with their CNs, following scheme proposed by Kausar et al. [20-21] on Key Management and Secure Routing in Heterogeneous Sensor Networks. CHs are assumed to be equipped with tamper resistant hardware. CNs have chance to be compromised

during route discovery or routing table generation phase, but intruder will only have access to one secret key that it shares with CH, rather than secret key of whole network. Each CN is programmed with only one key to authenticate its CH. Cluster formation follow scheme proposed by Du and Lin in [22].

3.2.1 Threat Model

It is assumed that CN can be compromised and adversary can have access to cryptographic data stored on node. CNs can be reprogrammed and redeployed in the network. Stajano and Anderson [23] described some traditional trends of security in hierarchy of CIA: Confidentiality, integrity and Authentication. But in case of compromised nodes authenticity is the major issue and confidentiality becomes minor one. Adversary can launch attacks like eavesdropping, message replay, bogus routing, selective forwarding and sinkhole attack DoS attack and jamming.

3.3 Bloom Filters

Bloom filters are space efficient data structures for membership addition and query, stored by all sender and receiver nodes. It stores set of entries to support membership queries, having zero false positive rate and minimum false negative rate. BFs are used to represent a set U= $\{u_1, u_2, u_3, \ldots\ldots u_n\}$ with *n* entries and is an array of *m-bits*, set to zero initially. BFs uses *k* independent random hash functions say $H_1, H_2, \ldots\ldots\ldots H_k$ where $\{0 \leq H < m-1\}$. For each entry u_j ,the bits $H_i(u_j)$ are set to 1 for $1 \leq j \leq n$ and $1 \leq i \leq k$. To check if a new entry *x* is in U, we check whether all *Hi*(*x*)-th bits in the array are 1. If not, *x* is absolutely not in *U.* Mitzenmacher showed that false positive rate of Compressed BF is less than normal BF [20]. Actually it reduces the communication over head over the expense of storing BF at both sender and receiver end, with compression and decompression requirement.

Bloom filter are excellent data structures, which is randomized i.e. it uses randomly selected hash functions. Therefore it has some false positive rate, i.e. probability that it may incorrectly return that an element is in a set when it is not there. This probability of false positive can be made sufficiently small and space saving is significant enough that that Bloom filters are considered useful. The probability of false positive rate can be calculated in straight forward fashion, with assumption that hash functions are perfectly random. All the elements of

set U are hashed into Bloom filter, then the probability that a specific bit is still '0' is previously calculated by Almeida et al where they evaluated the statistics behind Bloom filters:

$$\left(1-\frac{1}{m}\right)^{kn} \approx e^{-\frac{kn}{m}})$$

We let $p = e^{\frac{-kn}{m}}$. Then the probability of false positive is:

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1-e^{-\frac{kn}{m}}\right)^{k} = (1-p)^{k}$$

Now let f=false positive rate.

$$f = \left(1-e^{-\frac{kn}{m}}\right)^{k} = (1-p)^{k}$$

There are three performance metrics for the Bloom filters that can be trade off: these are computation time (depending upon number of hash function 'k'), size (depending upon the array size 'm'), and probability of error (depending upon its false positive rate 'f ').

If *m* and *n* are known to us then, optimized number of hash functions *k* can be found to minimize the false positive rate *f*. using two parameters, i.e. more hash functions gives more chance to find a 0 bit for the element that is not a member of S. otherwise using fewer hash functions increases the part or fraction of zero bits in array. Therefore required is the optimal number of hash function.

3.4 Notations and Terms

Table.3.1 Notation Table

BS	Base Station
CH	Cluster Head

CN	Cluster Node
MACR	Message Authentication Code Request
MACRR	Message Authentication Code Response
BF	Bloom Filter
U	set of *μTESLA* instances
Ly	L
$K_{CHi,CNj}$	pair wise key between Cluster Head and cluster node
C_x	Counter value at x interval
E	Epoch

3.5 Proposed Scheme

This scheme presents the implementation of security architecture within routing protocol design, instead of having separate protocols for efficient routing and intrusion detection systems.

Our scheme consists of two parts: a) route discovery b) data forwarding. It is assumed that CNs use shared pair wise keys for authentication of messages transferred between CH and CNs. Further, CNs are pre-loaded with BFs, having hashed *μTESLA* instances required for one way broadcast authentication.

3.5.1 Route Discovery

This part of propose scheme includes building routing tables, for each node. There are three phases, in the first phase, *route request* message is broadcasted from the BS to all of the CHs, and then CHs re-broadcast this message to their entire CNs. In the second phase, all CNs send their topology information back to CH via *route response message*. In third phase, as each CH knows the topology of its cluster, computes forwarding tables locally for each CN based on the information received in the previous phase. As a consequence, CHs generate pair-wise keys for each pair of neighbor nodes that are on same path. Then CHs send routing tables and pair-wise keys to each CN and aggregated data to the BS.

3.5.1.1 First Phase

To create the network topology, BS commences first phase, as it needs to construct forwarding tables, or whenever there is some change in topology of network due to addition or removal of nodes from network. As shown in Figure 1, CH broadcasts ROUTE REQUEST message and CN receiving that message for the first time rebroadcasts that message. ROUTE REQUEST message broadcasted by node say 'A' contain path from CH to that node. Node 'A' appends its identity to the message. Node 'A' also stores identity of the sender of message to its array of neighbor nodes.

1. *CH_x => *: Route Request (Id_{Lx})*
2. *CN_y: adds the Id CH_x to neighbors List*
3. *CN_y =>*: Route Request appending its own identity (Id_{Lx}, Id_{Ly}…)(But for the first time only)*
4. *CN_y: repeat 2 step for each Route Request message*

Figure: 3.2. Neighbor Node Discovery Phase

Receiving a duplicate request message node 'A' does not broadcast that message but stores identity of sender to its array of neighbors. Nodes may receive request from multiple CH, but response to the one with strong signal strength and stores other as a backup. This is a type of neighborhood discovery phase, to let nodes know of its neighbors and assists to forward feedback message containing topology of cluster from each CN back to CH. In response to this single ROUTE Discovery message CH discovers multiple routes to any destination.

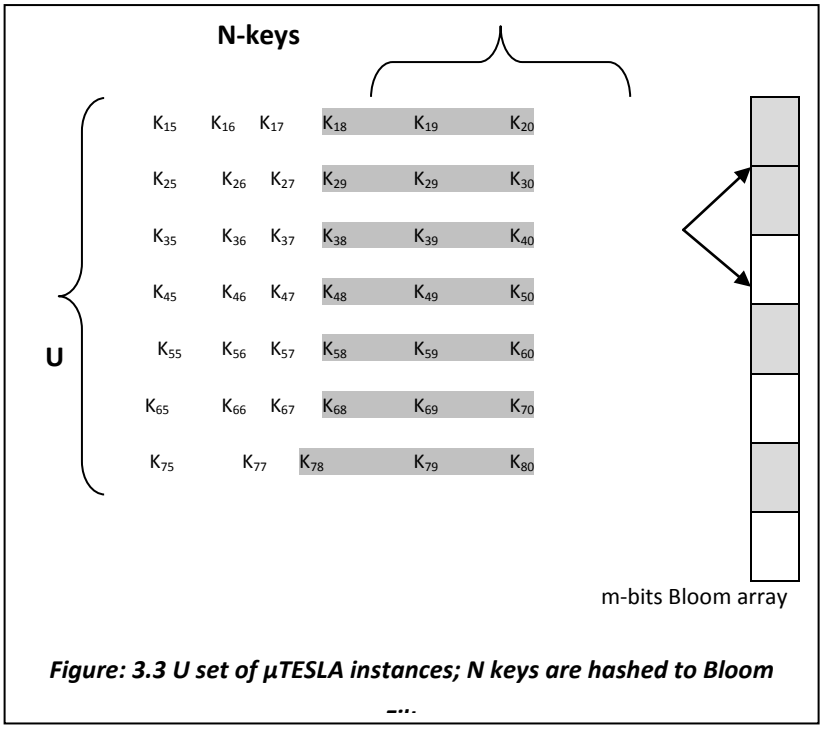

Figure: 3.3 U set of µTESLA instances; N keys are hashed to Bloom

Broadcast Authentication

In order to check if the ROUTE REQUEST message is initiated by the BS at CH level and from CH at CN level and to restrict DOS flooding attacks, there is a need of broadcast authentication. For this purpose, we are using hybrid technique of *µTESLA* and *Bloom Filters.* It is assumed that all CH and CN have one BF pre-loaded with keyed hashed values for authentication purpose. Further, we consider set U of independent *µTESLA* instances as shown in Figure 3. Last *N* keys of µ*TESLA* are hashed and entered to m-bits BF. On receiving ROUTE REQUEST message receiver check for the instance (ki, ki+1,…..,kn) used to generate message authentication code MACR for ROUTE REQUEST by applying one way hash function on it. It then query BF for the presence of hash of instance, if query returns true, message is forwarded, if false message is dropped and identity of sender is stored for future reference. BS sends BF of next *U,* µ*TESLA* instances, before end of previous instances.

Whole network time is divided into epoch and message counter at sender node is reset in the beginning of each epoch. The number of broadcast messages by each node per epoch is bounded to *k*. so that if CN is compromised and sends messages at very fast rate, the upstream node will forward that message at its normal defined rate. This prevents the network jamming.

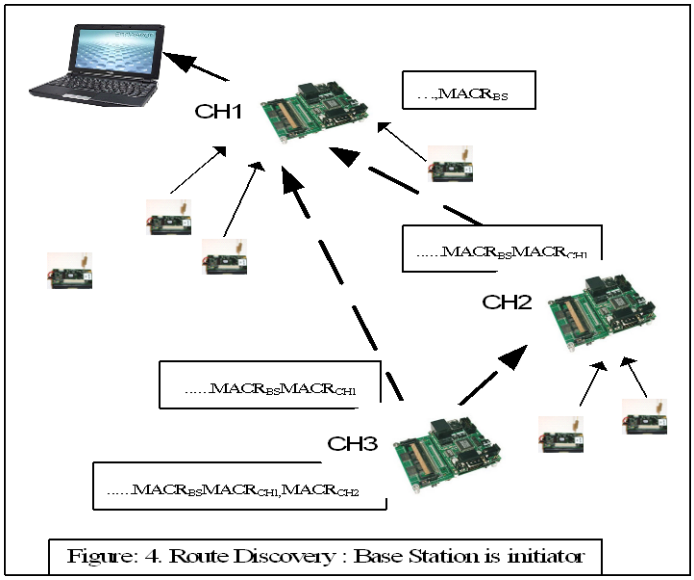

Figure: 4. Route Discovery : Base Station is initiator

Further if adversary or compromised node replay old legitimate ROUTE REQUEST message, it is prevented again using BF. Each node have two BF for two epochs i.e. E_i current epoch and E_{i-1} the previous.

All of the valid messages received in epoch E_i are stored in BF_i, using one way hash function on counter value and source identity ($C_x||Id$). And BF_{i-1} has all the legitimate messages of previous epoch E_{i-1}. At the start of epoch E_{i+1} all messages of BF_{i-1} are drooped to accept messages of $E_{i+1.}$

To help prevent malicious node entering fake path or bogus route discovery messages, each message is appended with MACR calculated using $K_{CHi,CNj}$ before forwarding ROUTE REQUEST message to next node. Figure.4 shows that CH-3 receives ROUTE REQUEST message from CH-1 with MACR of BS and CH-1 and from CH-2 with MACR of base station, CH-1and CH-2.

16-byte MACR is calculated by node say 'A', this is calculated over the complete message, including node 'A' own appended identity, but if all CNs append 16-bytes MACR to the ROUTE REQUEST message, overhead increases, for this purpose only last say 64-bits are appended to message. Also this overhead is only for the period of route discovery and routing table formation phase.

MAC is generated using secret key of CN 'A' using following values.

$MACR_A$ = MAC (AM, Len, path, key_A)

Where *AM* is active message type, to help CNs know that CH is collecting topology information for building the routing table, it shows that this is ROUTE REQUEST message, '*Len*' field contains the length of the path, , *path* field contains path from CH to the current CN.

MAC message authentication code is used at this stage for the integrity of the paths entered by the CNs, if compromised node adds a fake path in the ROUTE REQUEST packet, its MAC can not be verified by the CH and it will be discarded; CNs ID is stored by the CH for future reference.

3.5.1.2 Second Phase

In this phase, ROUTE RESPONSE is returned back to the initiator of Route discovery; in figure.5 CN, 'D' sends its local connectivity information (their identities Id_i, $MACR_i$)., path-

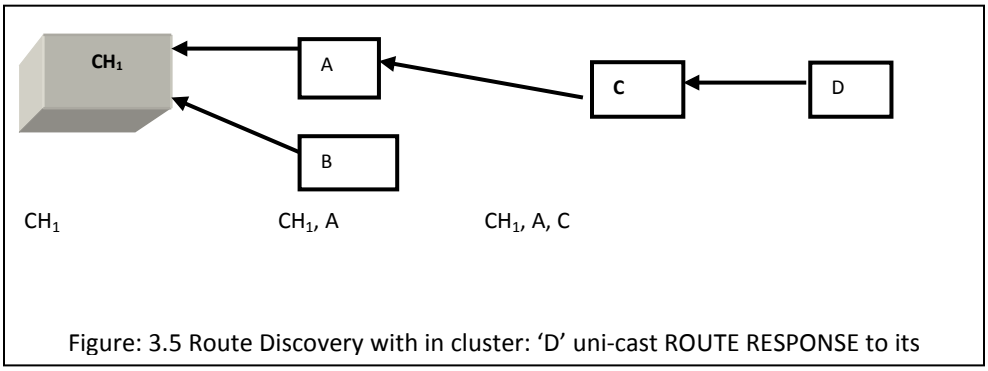

Figure: 3.5 Route Discovery with in cluster: 'D' uni-cast ROUTE RESPONSE to its

sequence from upstream neighbor that broadcasted this ROUTE REQUEST message to node, and *chosen-parent* (i.e. $MACR_p$) one that first broadcasted the ROUTE REQUEST message to CN 'D'. With in cluster all CN are rooted toward CH. There is certain timeout interval between generation of ROUTE REQUEST and ROUTE RESPONSE messages, during which nodes hear local broadcast of ROUTE REQUEST message from its upstream and downstream neighbors.

Figure.5 shows CN 'D' receives ROUTE REQUEST from CN 'C' then 'C' becomes chosen-parent for 'D' and path received from CN 'C' is **CH_1->A->C** then path-sequence which is returned in the ROUTE RESPONSE from 'D' includes **CH_1->A->C->D.**

For the secrecy of REQUEST RESPONSE returned to CH, 16 byte MACRR of message is calculated and appended to message. Similar to MACR only last 64 bits are appended to reduce memory over head.

Path-seq = (Id_A, Len, $path_A$, $MACR_A$)

nbr-info= (Id_A. $MACR_A$, Id_B. $MACR_B$,……, Id_N. $MACR_N$)

$MACRR_A$= MAC (AM, path-seq, nbr_info, key_A)

Using MACRR local topology of network is securely transferred to CH so that it can construct routing tables for all CNs, CH do communicate with other CHs for their local connectivity information, and to find out multi-paths, complete connectivity might not be available to CH due to the presence of compromised nodes in the network, but it is assured that the information transferred is true and is secure and verified by the CH. And in case if compromised node injected some false neighbor information, it is detected by CH when it counters checks the MACR of neighbor information list of compromised node and list of all the neighbors of compromised node. Message is forwarded only to parent node as multi paths are not available at this stage.

3.5.1.3 Third Phase

This is routing table generation phase and propagation of these tables to each CN. CH after broadcasting ROUTE REQUEST message waits for certain time out period, during which they receives local connectivity information of all CNs, through their ROUTE RESPONSE messages. Security mechanisms during first two phases guarantee, information that reaches BS via CH is secure.

Further CH computes MACRR for each ROUTE RESPONSE message, and compares that with the received one, if two matches then it go further for neighbor's connectivity information. Similarly CH sends its cluster connectivity information to BS. MACRR is to guarantee that neighbor connectivity information that reaches CH is correct and all neighbors have broadcasted ROUTE REQUEST message. CN one hop away from CH sends their ROUTE RESPONSE messages direct to CH other sends via CNs on route.

CH sends ROUTE RESPONSE message back to BS appending its MACRR BS counter checks for the neighbor list of different CHs and their CNs, also BS generate multi-paths between same sender and receiver using any of Shortest Path algorithm. BS generates and sends ROUTING TABLES with neighbor information of CHs.

Finally CH generates routing table and pair wise keys for cluster members and send that to each CN. ROUTING TABLE of each node is encrypted using key, $K_{CHi,CNj}$, then one way keyed MAC is generated on that message, and appended.

3.5.2 Data Forwarding

Using these tables' nodes can communicate through their neighbor nodes on route directly using pair wise keys generated by CH. This key is used simply to encrypt message it sends, to destination nodes. Message includes sender Id, receiver Id, and destination Id. On receiving message CN check for destination Id, if it present in neighbor list, forwards message to next CN. At this stage multi-path have been discovered and message is forwarded on all paths available; to prevent message from jamming attack, selective forwarding or sinkhole attack. Message integrity is assured by encrypting it; authentication and replay protection is guaranteed using μ*TESLA* and BFs. If adversary compromises any of CN during data forwarding, it can only decrypt message sent to that CN. Message from CH to BS is also forwarded on multi paths with CHs on routing table list. Figure.4 shows two routes available for $\mathbf{CH_3}$ in its routing table: $\mathbf{CH_3}$->CH_1->BS **or** $\mathbf{CH_3}$-> CH_2->CH_1->BS.

Within cluster only CH is allowed to update CNs routing table and also send updates to BS. These are authenticated updates, encrypted by $K_{CHi,CNj}$.

3.6 Overview

This chapter presents the secure routing scheme with messages broadcasted by base station and cluster head; which are checked for authentication at cluster nodes. Then secure route discovery using keyed MAC and encrypting the messages. MAC been verified at CH. Therefore data been transferred and route discovered is secure. Replay protection and DOS style flooding attacks are avoided using BF and control flow of messages. Further maximum computations are performed at CH and only aggregated data is transferred to BS, saving its computational load. Cluster are smaller therefore there is less computation burden on the CH.

Design and Implementation Architecture

4.1 Introduction

This chapter deals with the actual implementation of our proposed scheme. Proposed scheme is about secure routing with broadcast authentication in wireless sensor network. Sensor network works in insecure environment therefore security is the core issue, and being energy, memory and power constrained, it is more challenging to design secure routing scheme for these. Our proposed scheme work in two parts: a) Route discovery b) Data forwarding. First part deals with the secure route discovery, it includes three phases: first phase deals with Route request Messages being broadcasted. Second phase deals with route response messages which as the name indicate are in response to route request messages. These are uni-casted by each node. Third phase deals with the routing table generation.

Proposed scheme is implemented on a network of 10 sensor nodes. Development tool as described earlier in chapter 2 is TinyOS using NesC language. TinyOS is open source operating system specifically designed for low power wireless devices, like sensor networks, PAN (Personal Area Network), and smart buildings. TinyOS is embedded operating system written in NesC language as a set of cooperating tasks and processes. TOSSIM simulator is used for simulation of network. Secure Routing and broadcast authentication scheme is implemented in two parts route discovery and data forwarding.

Network hierarchy is heterogeneous sensor network; with one base station BS, few cluster heads CH and more cluster nodes CNs. Tmotes are assumed to be CNs. Tmotes are small embedded devices which are capable of measuring several features in the surrounding environment such as temperature and light. Tmotes are equipped with a radio transmitter which can be used to communicate between Tmotes. Tmotes are controlled by TinyOS. Important consideration of scheme is that during simulation first node indexed ‘0’ is always base station. Then second node indexed ‘1’ is cluster head CH and then rest of the nodes are cluster nodes. Simulation is implemented for one to two clusters. This hierarchy is hard coded when

number of nodes is five. Index ‘0’ is BS; index ‘1’ is CH and rest of ‘2’, ‘3’, and ‘4’. When more nodes are added to network next coming node become CH and further three becomes CNs. One more change that has been made using TinyOS is that message length TOSH_DATA_LENGTH in file AM.h has been changed to 100 i.e. maximum.

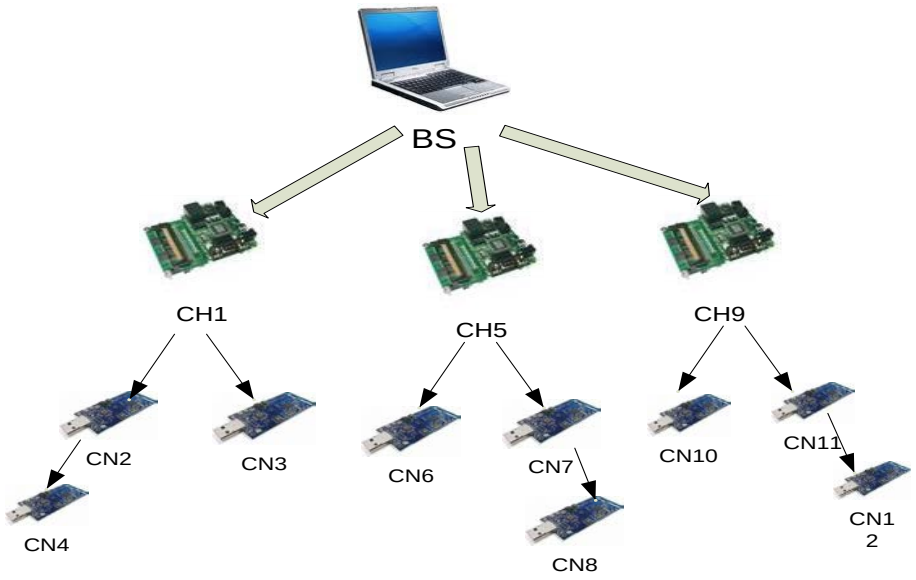

Figure 4.1 Hierarchical HSN

4.2 Implementation of Secure Routing Scheme

It consists of two phases’ pre deployment and post deployment. In pre deployment phase base station is pre-loaded with keys that it shares with CHs, and CHs are pre loaded with key that it shares with CNs following the scheme proposed by Kausar et al. Cluster heads CHs are equipped with tamper resistant hardware. Each CN shares unique secret key with cluster head. In this case if CN is compromised only one key is compromised instead of the whole network as in case of INSENS. Each node is equipped with three bloom filters. One for broadcast authentication, hashed with µTESLA instances on receiving of key node compute hash and check for its presence in BF if found key is valid and MAC calculated with this key is also valid therefore message is authenticated, and rest of two for replay protection. Bloom filters are simple arrays.

4.2.1Configuration of Bloom Filter

As discussed in chapter 3 there are three performance metrics for the Bloom filters that can be trade off: these are computation time (depending upon number of hash function 'k'), size (depending upon the array size 'm'), and probability of error (depending upon its false positive rate 'f ').

As bloom filters are configured by two parameters size *m* and number of hash functions *k*. under pessimistic assumption of the hardware and network activity, using bloom filters for checking replay attacks, we can achieve 1% of false positive rate by using m=18 bytes, and k=8 hash functions when epoch time t=1s. The false positive rate of BF can be calculated based on number of stored items. For this we can use approach of Mark Luk used in MiniSec [10]. If we upper bound the average number of packets received (p_u) in one epoch of length t_e, these are known to us e.g. regular heart beats. BF can be configured accordingly.

Assume t_l be the lower bound of node's life time, E_c be the energy of node battery and E_p be the energy consumption for receiving one packet then. If all energy of battery is consumed in receiving packet then maximum number of packets received over the life time of node is E_c/E_p.

Then the number of packets received in one epoch:

$$p_u = \frac{E_c t_e}{E_p t_l}$$

The probability 'p' of false positive in BF after inserting n elements as calculated by Ameida et al.'s is:

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^k$$

4.3 Communication during Route Discovery

Three types of messages are communicated during route discovery. In phases one of route discovery, route request message is initially broadcasted by BS to CH then CH broadcast this message with its own MAC, as shown in flow diagram.

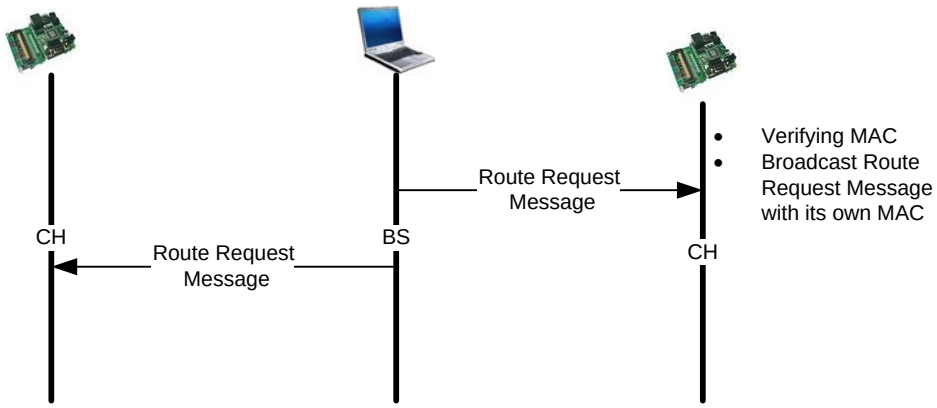

Figure: 4.2 Base Station to Cluster head communication

As discussed earlier in chapter 3 on receiving route request message CH will generate its route request message with its own keyed MACR and broadcast that to CNs. On receiving that message node first check message type if its broadcast then, check Path id in Path field if its own Id found it will not rebroadcast message, otherwise after receiving key calculate its hash and check in Bloom filter if entry found there calculate MACR and compare with received one if matches, node append its Id to Path field, increment in Length field so number of hopes message had traveled can be checked. Calculate MACR over the new message and rebroadcast the message.

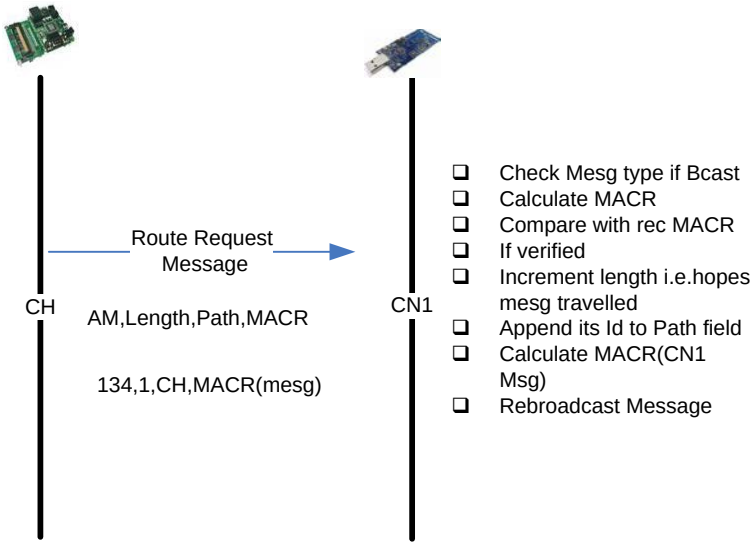

Figure 4.3: CH to CN communication on Broadcasting Route Request Message

After receiving the route request message CNs will wait for 500ms, so that it can receive route request message from all upstream and downstream nodes, after 500ms CN will uni-cast the route response message over the path from which it received the message.

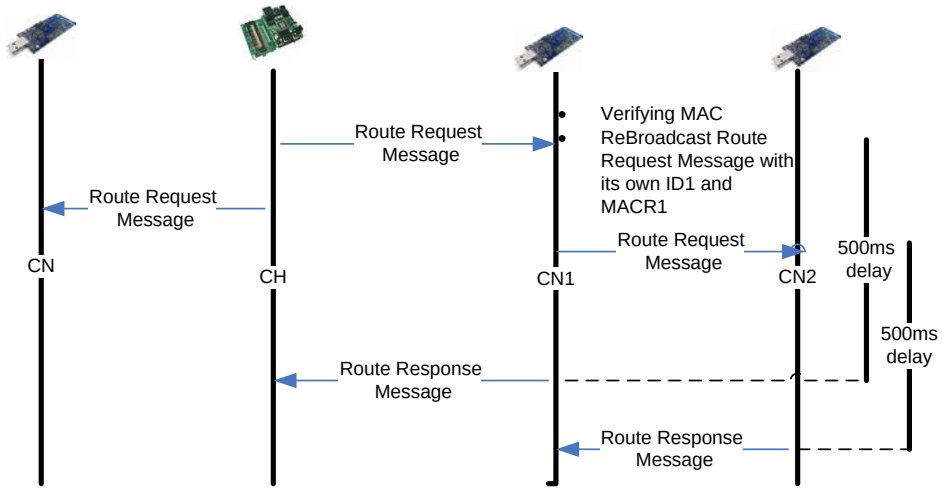

Figure 4.4: Delay between receiving Route Request Message and sending Route Response Message

4.3.1 Basic Flow of Route Request Message:

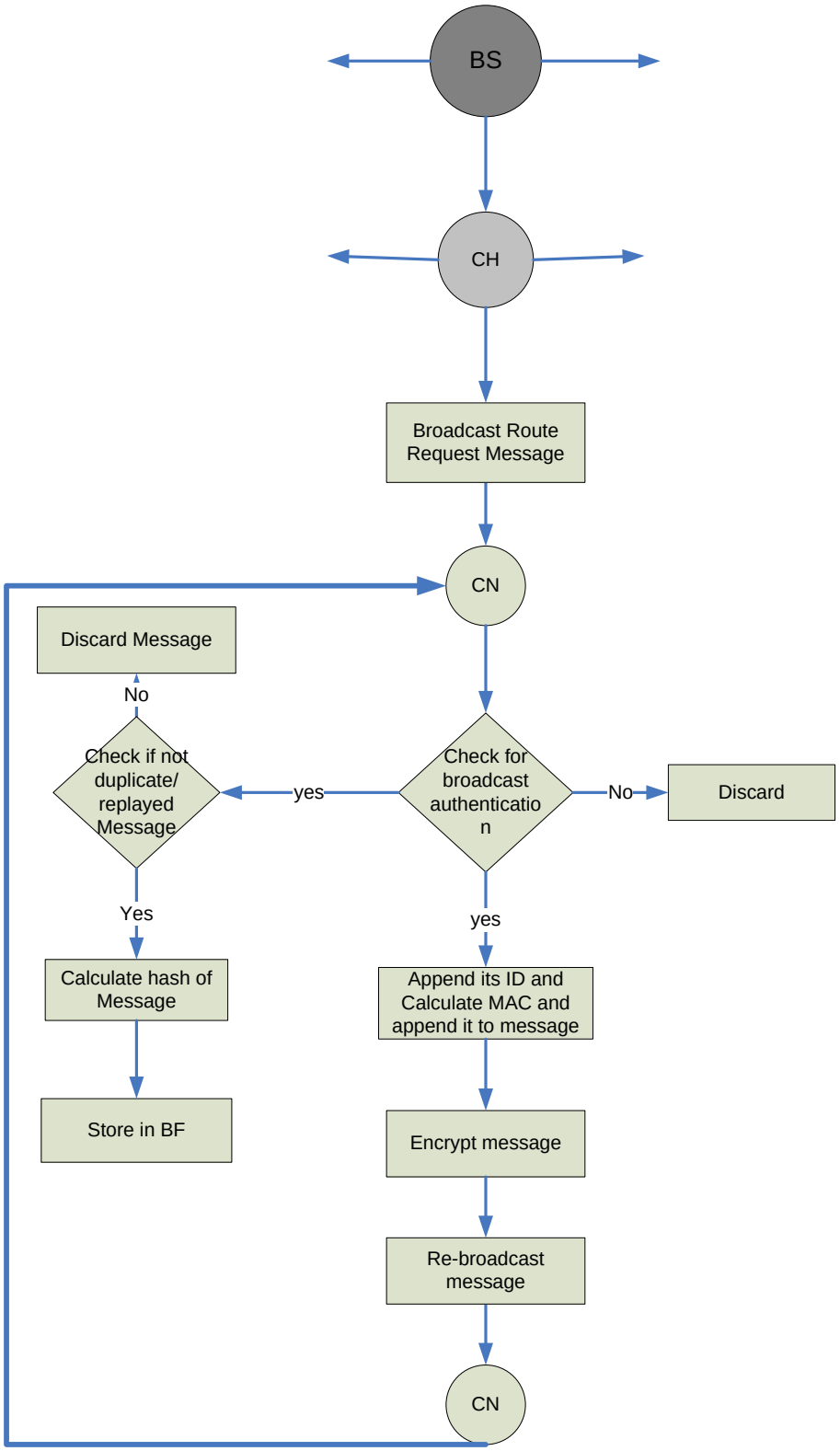

Figure: 4.5flow of Route Request Message

4.3.2 Basic Flow of Route Response Message

Cluster node CN after receiving route request message wait for 500ms and then uni-cast the route response message, on receiving the route response message CN identify the sending node as its parent node and place parent identification information i.e. MACRp in the parent_info field of route response message, CN already have this information from the parent original request message, this determine which upstream neighbor is the parent who should forward the message next, message contain Id of sending node, path_seq as received in route request message, nbr_info including MACR of all immediate nodes, encryption is applied over the path_seq and nbr_info fields and then calculate MACRR over the complete message. This MACRp addressing function selects the specific parent from all upstream nodes to forward this response message. When upstream node hear the local broadcast message whose MACRp does not matches with its own MACR, then it knows that its not parent and should not forward message. If the two MACRp match, then node knows that that it is the selected parent and forward message.

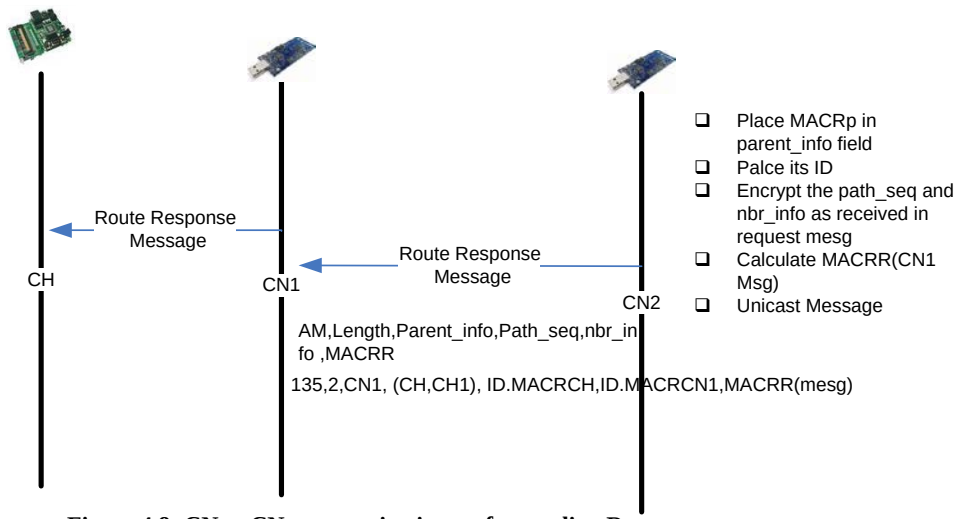

Figure 4.8: CN to CN communication on forwarding Route Response Message

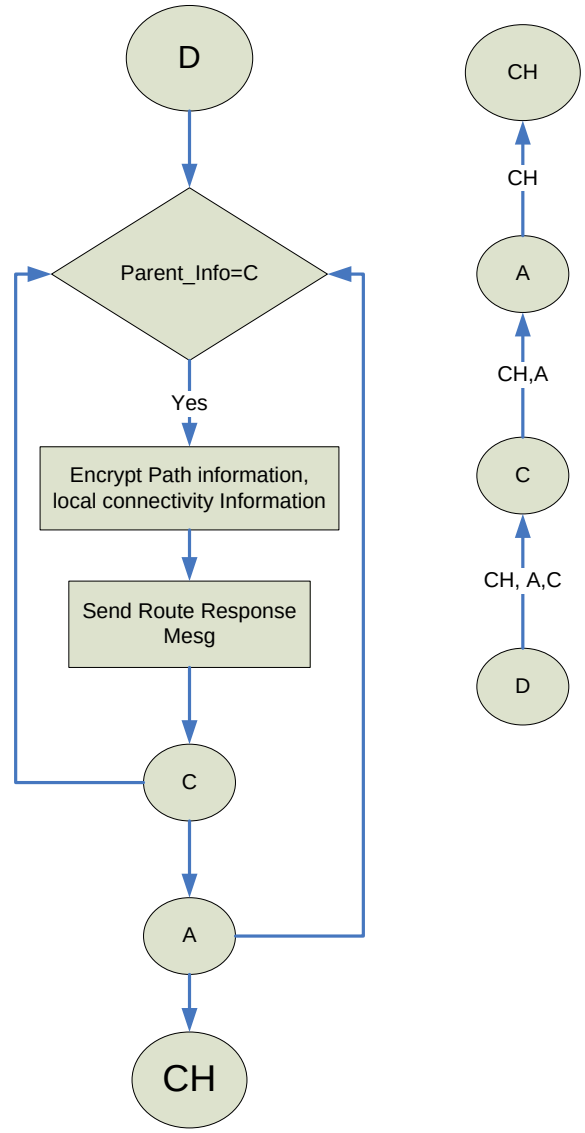

Figure: Route Response Message

Figure 4.7Route Response Message

4.3.3 Basic Flow of Routing Table Generation

After receiving route response message, CH computes the MACRR and verifies that there is a match. If there is a match; then CH match the node list as neighbors with the information it receives from different nodes. The MACRR is the proof that nodes heard each other. From the connectivity information CH computes the routing tables of each cluster node.

And then generate pair wise keys for the neighbor nodes so that they can communicate with each other without directing message via CH. This reduces the communication over head of overall network.

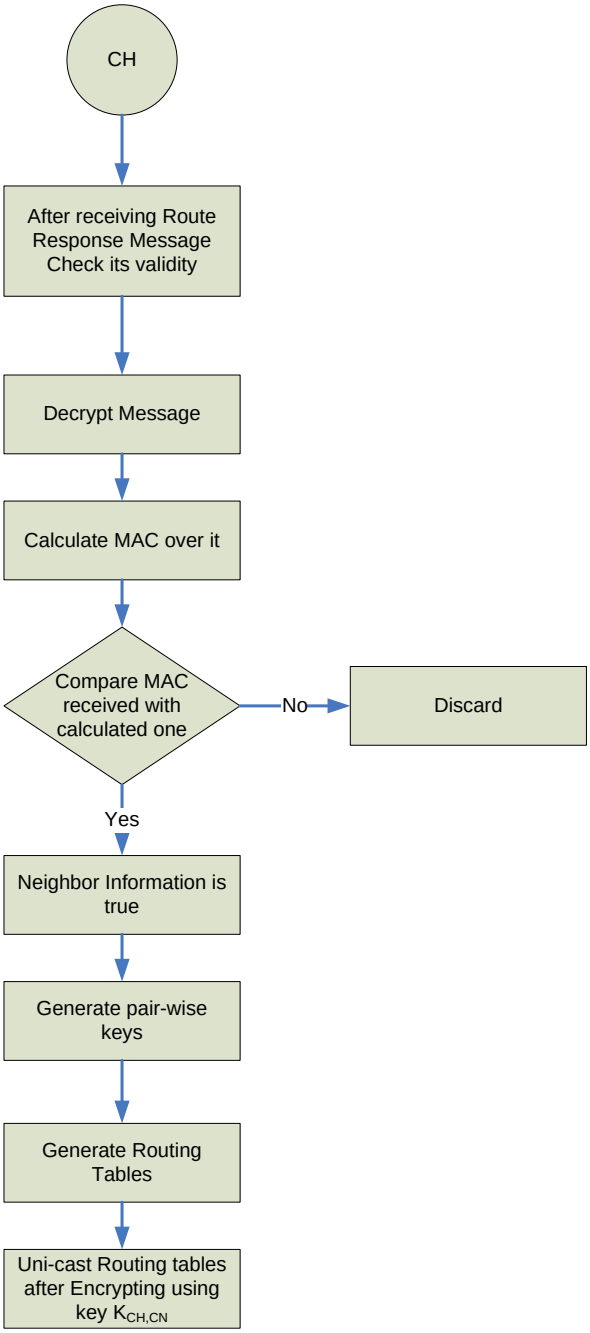

Figure: 4.8 Routing Table generation

4.4 Messages used in TinyOS

4.4.1 RouteReqMsg

After the start of simulation this is the first message, which is broadcasted for the route discovery. On receiving this message CH broadcast the route request message to its cluster members, and cluster nodes on receiving this rebroadcast the message.

RouteReqMsg	
PARAMETERS	**LENGTH**
1- AM	Uint8
2- Length	Uint8
3- Path	Uint8 (array of length 16)
4- MACR	Uint8 (array of length 16)
5- rec_mac	Uint8 (array of length 16)
PURPOSE: To inform CN that route request is broadcasted to generate routing tables	

4.4.2 RouteRespMsg

RouteRespMsg	
PARAMETERS	**LENGTH**
1-AM	Uint8
2- Parent	Uint8 (array of length 16)
6- Path_seq	Uint8 (array of length 16)

7- Nbr_info	Uint8 (array of length 16)
8- MACRR	Uint8 (array of length 16)
PURPOSE: To inform CH of local connectivity information	

4.4.3 Routing Table forwarding Message

RouteTblMsg	
PARAMETERS	**LENGTH**
1-AM	Uint8
2- dest_add	Uint8 (array of length 16)
3- tabl	Uint8 (array of length 16)
4- imd_snd	Uint8 (array of length 16)
4- MACG	Uint8 (array of length 16)
PURPOSE: To generate routing table and pair wise keys for CNs	

4.4.4 Key Disclosure Message

KeydisclosMsg	
PARAMETERS	**LENGTH**
1- id	Uint8
2- ID_key	Uint8
3- MAC	Uint8 (array of length 16)
PURPOSE: Key disclosure to CHs	

Figure 4.5: tinyviz simulation

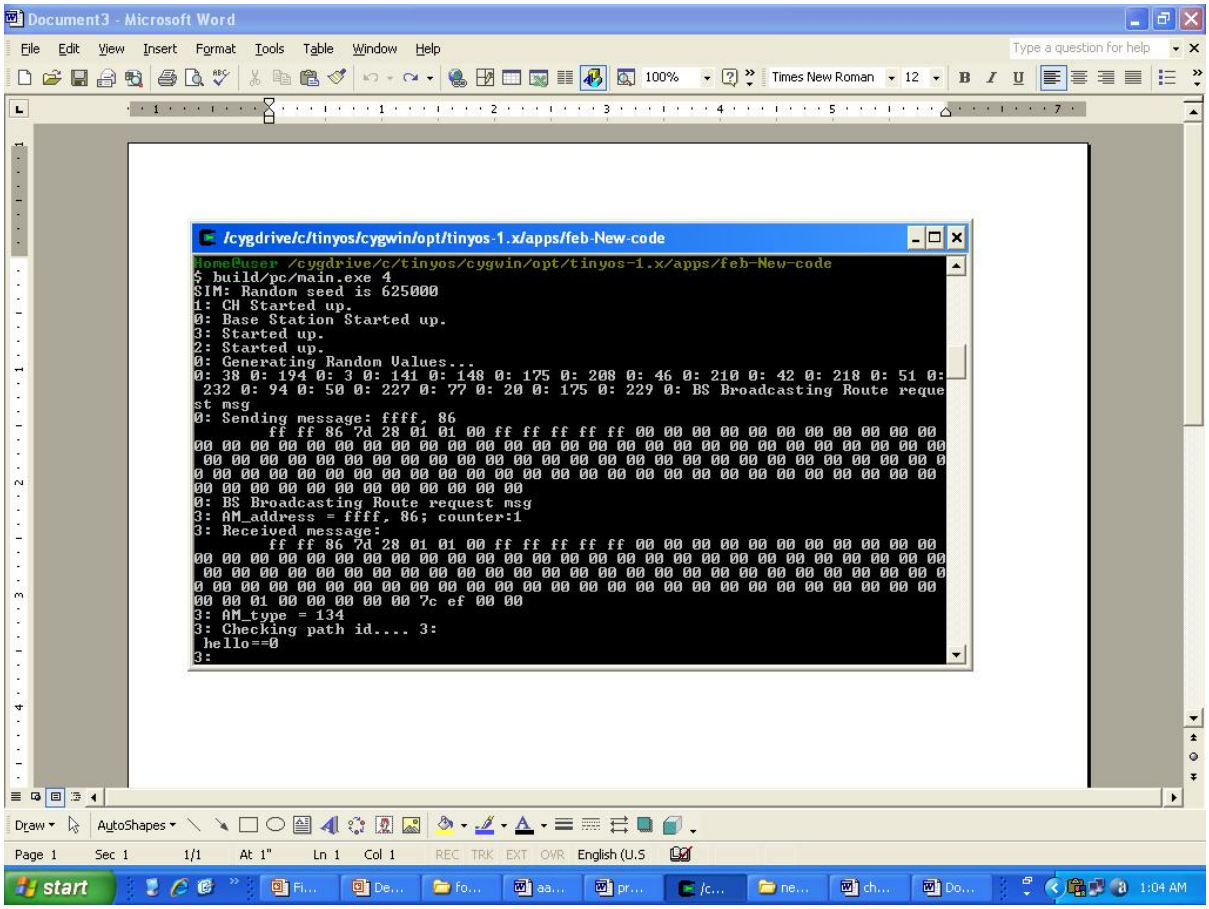

Figure 4.6: cygwin bash output

4.5 Overview

This chapter includes few of the messages included in simulation which is implemented in TinyOS. Hierarchy of the messages which leads to routing table generation are included. Parameters used for implementation purpose

Implementation Performance and Security Analysis

5.1 Introduction

This chapter include the implementation, performance and security analysis of our propose scheme with the existing routing and secure routing protocols. Security parameters added, techniques used for calculating those security measures and then their analysis and result is included.

5.2 Calculating HMAC

For authentication of message keyed hash authentication is used. For this some suitable algorithm was required. Purpose of any hashing algorithm is; if digest and the corresponding message from which the digest is derived are available; it should be computationally infeasible to construct a different message with the same digest. We are using HMAC-MD5 for calculating MAC of the message at all stages. MD5 is basically a hashing algorithm, it takes a message of 264bits and reduces its size to a digest of 128bits i.e. 16 bytes. MD5 is a development of algorithm invented in 1990 by Ronald Rivest called MD4. MD4 was flawed therefore some revisions were made to it and as result christened MD5 developed. We are using 8bytes of calculated MAC for message communication.

Another more secure hashing algorithm is SHA1, this was developed by NIST, and it produces 160-bit hash of 264bits message , which definitely is larger then MD5, therefore its storage and computational requirement is larger then MD5. And it is slower then MD5.

Security Strength of HMAC

It is studied that security strength of MAC depends on its length. Mostly protocols use 8 or 16 bytes MAC but some protocols like TinySec uses 4-bytes MAC. As discussed earlier that we are considering TelosB motes as cluster nodes, and we are using 8-bytes MAC. Because of two reasons:

- Because wireless sensor networks cannot afford sending 16 bytes MAC additional with the message packet. As the default packet size of TinyOS is 36bytes, 16-bytes MAC is too large for it.
- It is observed that on Micas2 platform with radio CC100, it takes 20 months by an attacker to forge 4-byte MAC. While TelosB with radio CC2420 it takes 3 months to forge 4-bytes MAC, because band with of the radio is 6 times larger then Mica2.

Since we are using 8-bytes of MAC; this will significantly increases the security strength of the proposed scheme.

Performance

We have studied several different MAC algorithms and observed that memory requirement of block cipher based MAC is low, as compared to the hashed based. But hash based algorithms has shorter computation time therefore they consume less energy which is major limitation of sensors networks. It is also observed that HMAC takes much ROM space because of MD5 implementation; but it consumes less energy especially on TelosB motes.

5.3 Encryption/Decryption

For the confidentiality of the messages; messages are encrypted for this purpose RC5 encryption /decryption algorithm is used. RC5 is chosen because of its small code size and high efficiency. RC5 does not rely on multiplication, and does not require large tables. However RC5 does use-32 bit data-dependent rotates. We have evaluated several other algorithms for encryption/decryption purpose but; AES algorithm Rijndael uses over 800 bytes of lookup tables, not suitable for memory constraint environment. Yet another algorithm DES block cipher require 512-entry S-box table and 256 entry table for various permutations, which is again not feasible for sensors with limited memory and computation resources. Therefore RC5 proves to be best option. And further to save code space, we have used the same function for both encryption and decryption. Important property of this block cipher is that; it is stream cipher in nature; therefore the size of cipher-text remains same as the size of plaintext. Which proves to be best quality in our working environment;

because message sending and receiving becomes very expensive in terms of energy the uses. Also long messages suffer from data corruption.

We have used RC5 which is default implementation of TinyOS present in TinySec. RC5 with 12-rounds and with message size of 36 bytes is used. Key size is 64-bits i.e. 8 bytes.

5.4 Network Setup Time

To measure the network setup time, we calculated interval between the time base station broadcasts its route request message and the time it receives routing table receive messages form cluster heads, as CH are going to forward routing tables. Assuming network to be dense, i.e. every node has several neighbors. Factors effecting the network setup time will be then: 1) execution time of cryptographic algorithms like RC5 and MD5, 2) execution time of packet processing and 3) waiting time including delay time of 300ms to 500ms by nodes, route response message waiting time and CH and base station waiting time. In our case CH waits 300ms to 500ms after receiving route response message packet. This time is reset with every route response message packet. When no more route response message arrives, CH will time out and compute routing table for its cluster. Every cluster node CN also waits for 500ms. CH then uni-casts the custom routing table for each node. CH give delay of 100ms between sending each routing table and pair wise keys for neighbor CNs. Computation time of RC5 based cryptographic algorithm is relatively short.

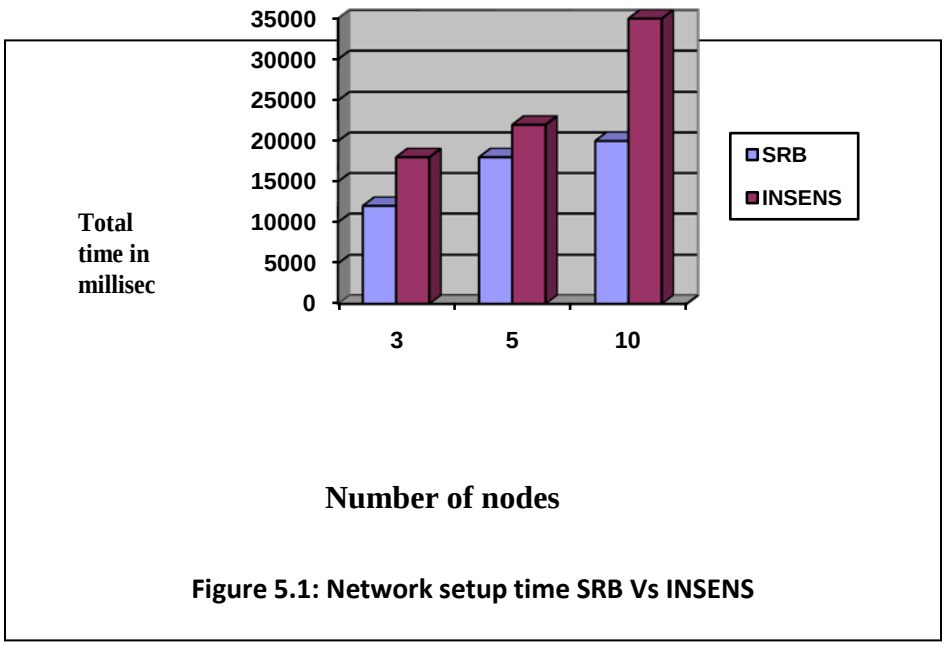

Figure 5.1: Network setup time SRB Vs INSENS

In case of basic INSENS protocol all of the nodes send their data to base station and then base station perform all heavy computations. But in our case CH computes the data locally, routing table generation computation are performed locally. This has significantly reduced the network setup time e.g. time taken by INSENS for network setup of three nodes is approximately equal to network setup time with six nodes in our case. CHs perform computations simultaneously therefore computations are timely done.

5.5 Proposed Scheme Overhead

5.5.1 Packet overhead

For comparison of this secure routing scheme we consider simple routing scheme; in which base station sends route request that is forwarded by all nodes once. And each of those node save identity of the node from which it receives the route request message. After delay say of 300ms nodes forward their response message to its identified parent. And message finally reaches base station, all nodes updates routing tables. Nodes forward response message after appending neighbor information. Base station receives the response message from all sensor nodes and computes network topology. Here it is seen that total number of messages or packets exchanged in this non secure routing protocol 2K, where K is the number of nodes in the network.

In comparison to this our proposed scheme sends more packets, and this difference increases with number of sensor nodes. This difference is due to attribute of security involved. But still due to cluster formation overhead is less as compared to secure routing protocol INSENS. Where if there are 10 nodes in network, then to generate routing table for node 7 hopes away, one packet may need to follow 6 or 7 hopes i.e. 7 packets for route request and seven 7 for route response and 7 for routing table receiving. But in case of our proposed scheme nodes are maximum two hopes from CH. Therefore routing table generation for node 2 hopes away need only 6 packets. But in case of INSENS it needs 7+7+7=21 packets. We have measured message overhead for different number of nodes for many sensor networks. Figure plot average

number of messages exchanged during route discovery as function of number of nodes in the network; for INSENS and our heterogeneous sensor network.

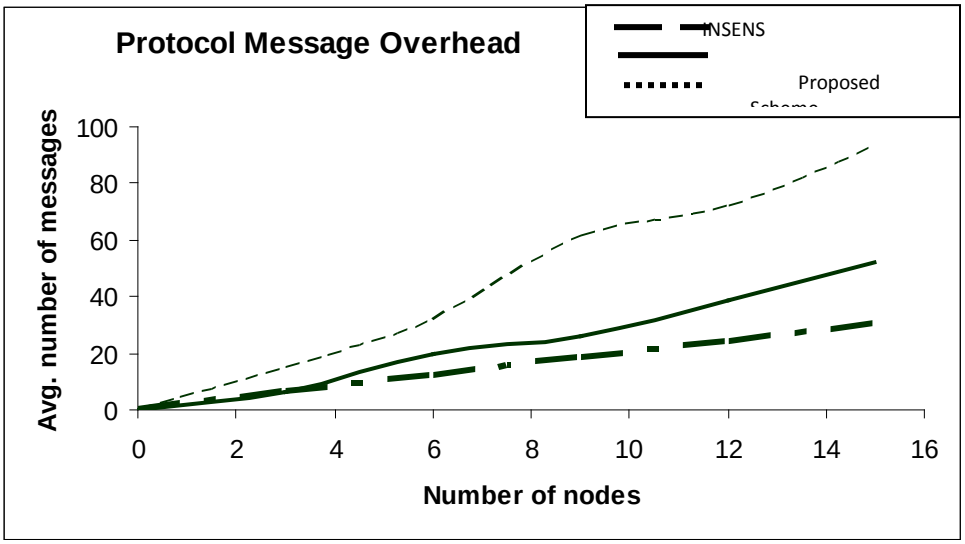

Figure 5.2: Routing over head of our proposed Scheme Vs secure three phase routing protocol INSENS Vs highly optimistic insecure single phase protocol

Broadcast authentication

Using bloom filter for broadcast authentication we have calculated total n=128 µTESL instances inset 'U'. We are considering the size of bit map bloom filter to be m=2048 bits. And hash functions to be used for this purpose are 2. We can compare the communication overhead when W=32 µTESLA instances are used. Last N keys to be hashed are N=n/W = 4. Using the normal µTESLA technique to preload all commitments it takes 4096 bits i.e. 32*128=512 bytes. In case of our scheme with bit map size 2048 it takes total of 256 bytes.

To verify new key chain overhead is Nk= 4*2= 8 hash operations.

The probability 'p' of false positive in BF after inserting n elements as calculated by Ameida et al.'s is:

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k}$$

We have calculated false positive rate 'f' for different values of hash functions.

$$f = (1-p)^k$$

Array bits per element	**m/n**	32	32	32	32	32
Hash functions	**k**	5	4	3	2	1
False positive rate	**f**	0.025	0.026	0.03	0.05	0.1

Table 5.1: false positive rate with at most thirty two bits per item

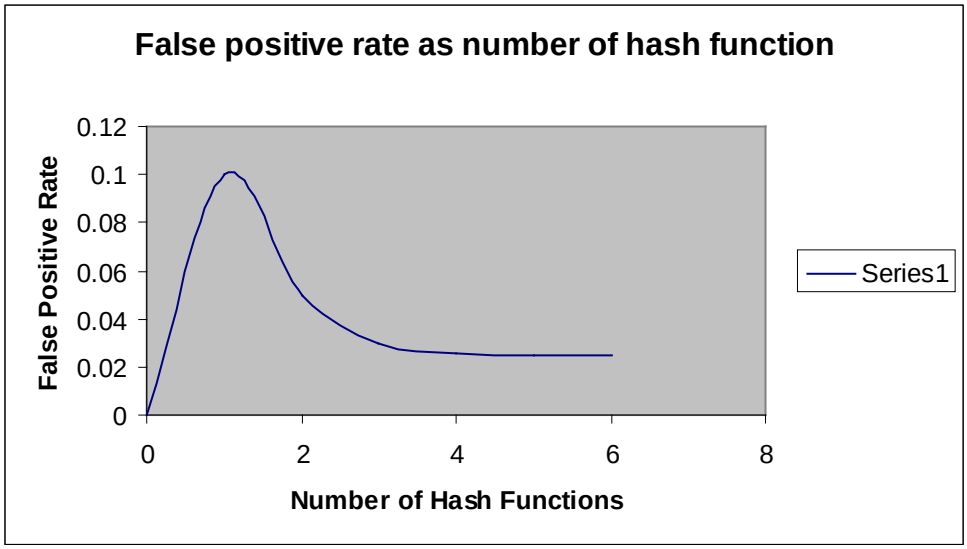

Figure 5.3: False positive rate as a function of number of hash functions using 32 bits per element

The adversary chance to break the system of broadcast authentication is by forging key chains which are mapped on to bloom filter bitmap; can be analyzed. Adversary can not reverse the hash function. It can break it using brute force; continuously generating a longer key chain till the time last N keys are all false positive in the bloom filter. Figure 5.4 shows the adversary chance to forge valid µTESLA instance. The chance of adversary to break calculated [20] is:

$$Adv = f^{N} = (1-p)^{Nk}$$

This figure gives the idea that with more entries in the bloom filter become more secure. And adversary's chance is less to break key chain. With more hash functions it become more difficult to break the chain.

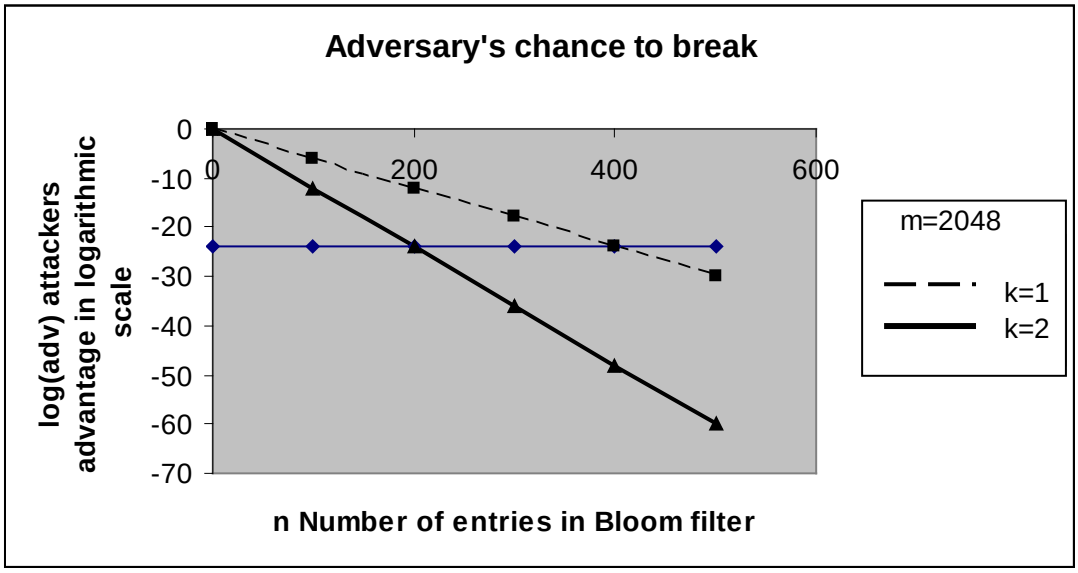

Figure 5.5 Adversary's chance to break broadcast authentication when m=2048 and W=32

Computation and Communication Overhead

Communication overhead has been reduced significantly, for the two reasons:

- First only 8-bytes out of 16-bytes calculated MAC is appended with the message.
- Secondly all traffic is not forwarded to base station, instead CN sends their connectivity information to CHs, and CHs calculate and verify the MAC of all nodes in its range. For example if there are 11 nodes in the network; 1 BS, 3 cluster heads, and rest CN cluster nodes. Each CH has to calculate and verify only max of 3 to 4 MAC. And have to generate pair-wise key for them. In case of Basic INSENS protocol BS have to calculate and verify all 10 MAC from the nodes. In our scheme; BS has to calculate only MAC of two CHs.

5.6 Security Analysis

The security analysis of our proposed scheme is in comparison with the existing secure routing protocol INSENS. During ROUTE REQUEST broadcast, malicious node can launch multiple attacks. Node receiving ROUTE REQUEST first checks if it's a valid message, by computing its keyed MAC. If message is not valid, it is dropped. Secondly receiver checks that message is not replayed. For this receiver queries corresponding BF for the presence of message, if query return true, message is dropped and if false it means it's new and is accepted and added to BF. BFs stores messages of current and previous epoch and therefore secure nodes from replay attacks. INSENS does not provide light weight broadcast authentication and replay protection.

INSENS does not provide any rate control method for message broadcast to prevent DoS attack. Number of message broadcast is controlled by (1) rate at which data is transferred (2) number of packets per epoch.

Compromising Probability

Resilience of our scheme against compromised node attack; if node is compromised, its cryptographic data is captured, but CNs only have limited number of keys that it shares with its neighbors and one secret key that it shares with CH. after the route discovery phase; each CH knows the local topology of its cluster, therefore CH sends the pair-wise keys for each pair of neighbor nodes that are on same path. These are sent to CNs along with their routing tables. CN use this pair-wise key to encrypt message it send to its neighbor. We analyze node compromise e.g. if CN say 'B' is compromised. Then effect of this node on the rest of network i.e. any two other CNs that are not compromised, probability that attacker can decrypt communication among say node 'C' and 'D'. This is called "**compromising probability**".

In our scheme each CN share uniqe key with its CH preloaded. After receiving pair-wise keys, the entire communicating neighbor CNs has different shared keys. Therefore compromising 'B' does not effect the communication between other CNs. Therefore impact of attack is locally and limited. In case of basic INSENS protocol, nodes do not share pair wise keys. Nodes can communicate only through BS, in case BS is compromised whole network is compromised.

To protect nodes against eaves-dropping during ROUTE RESPONSE phase only path _ seq and nbr _ info are encrypted using key $KCH_i;CN_j$, except its own identity in path _ info field.

Identity field is not encrypted for two reasons, to check for replayed ROUTE RESPONSE message, and to let CH know whose connectivity information it is.

In case of INSENS each node sends ROUTE RESPONSE to BS with complete MAC of all nodes, for verification. Message size increases therefore message is fragmented and work load and communication overhead is increased on BS. In our scheme only authenticated aggregated data is sent to BS from CH. To avoid sinkhole or wormhole attack CNs sends data only to its parent nodes and CH sends data to BS. Other nodes are not allowed to send data directly to BS. INSENSE is prone to sinkhole or wormhole attack. Adversary cannot launch Sybil attack, each nodes shares unique secret key with its CH and pair wise key among its neighbor nodes, therefore cannot posses multiple identities.

CONCLUSION

This research presents a secure routing scheme for HSN in order to improve security at design level and provide lightweight broadcast authentication using µTESLA and Compressed Bloom Filters. Broadcast authentication is essential to defense against compromised node entering bogus routing information, which is a major limitation of INSENS. Routing tables are generated for each node, which improves performance in case node is compromised or during jamming in some part of network. Further, computation and storage burden from CNs is reduced by computing forwarding tables at CH and BS. Analysis shows that proposed scheme is robust against different routing attacks.

6.1 Future Work

Proposed scheme is more secure than the previous one but still, there is need to have protocol more secure and energy efficient, in terms of computation. Also scheme is routing attack tolerant. Protocol is required for routing attack detection. Our scheme is using bloom filter for broadcast authentication along with µTESLA, in future instead of these standards bloom filters Compressed Bloom filters can be used, as they provide less false positive rate, and communication overhead is also less.

This scheme is implemented and analyzed in TinyOS using TOSSIM simulator, although this is for network analysis still to get accurate results there is need of actual implementation of the scheme in real environment. So that secure network can be implemented.

Bibliography

[1]. J. J. Deng, R. Han, and S. Mishra. Intrusion-tolerant routing for wireless sensor networks. *Elsevier Journal on Computer Communications*, 2005.

[2]. Chris Karlof, Yaping Li, and Joe Polastre. ARRIVE: Algorithm for robust routing in volatile environments. Technical Report UCB/CSD-03-1233, University of California at Berkeley, May 2002.

[3]. J. Deng, R. Han, and S. Mishra. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *IEEE IPSN*, April 2003.

[4]. V. Mhatre, C. P. Rosenberg, D. Kofman, *et al*, "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," *IEEE Transactions on Mobile Computing*, Jan. 2005, Vol. 4, No. 1, pp 4-15.

[5]. M. Yarvis, N. Kushalnagar, H. Singh, *et al.*, "Exploiting Heterogeneity in Sensor Networks," *Proc. of IEEE INFOCOM 05*, Miami, FL, Mar. 2005.

[6]. C. Karlof and D. Wagner, "Secure routing in sensor networks: Attacks and countermeasures," in *Proc. IEEE 1st Int. Workshop Sensor Network Protocols Applications*, May 2003, pp. 113–127.

[7]. F. L. LEWIS "Wireless sensor network", To appear in *Smart Environments: Technologies, Protocols, and Applications* ed. D.J. Cook and S.K. Das, John Wiley, New York, 2004.

[8]. *Chris Townsend, Steven Arms, MicroStrain, Inc.* "*Wireless Sensor Networks:* Principles and Applications", WilsonChapter22.indd

[9]. Katia Obraczka , "Wireless sensor Network Tutorial" Department of Computer Engineering, University of California, Santa Cruz, May 2006

[10]. M. Luk, G. Mezzour, A. Perrig, and V. Gligor. "MiniSec: A Secure Sensor Network Communication Architecture."In *Proceedings of the Sixth International Conference on Information Processing in Sensor Networks (IPSN 2007)*, April 2007

[11]. N. Abu-Ghazaleh, K. D. Kang, and K. Liu. "Towards Resilient Geographic Routing in Wireless Sensor Networks". In 1st ACM Workshop on QoS and Security for Wireless and Mobile Networks (Held in Conjunction with ACM/IEEE MSWiM 2005), Oct. 2005.

[12]. Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," 2001.

[13]. Brad Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Mobile Computing and Networking*, 2000, pp. 243–254.

[14]. E. Sabbah, A. Majeed, K. Don Kang, K. Liu, and Nael AbuGhazaleh, “An Application Driven Perspective on Wireless Sensor Network Security,” In Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks. Terromolinos, Spain-2006

[15]. Chris Karlof, Naveen Sastry, and David Wagner. TinySec: A link layer security architecture for wireless sensor networks. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.

[16]. Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual International Conference on Mobile Computing and Networks (MobiCom)*, pages 189–199, Rome, Italy, July 2001.

[17]. Kevin Chang, David Gay “Language Support for Interoperable Messaging in Sensor Networks”, proceeding of the 2005 workshop on software an compilers for embedded systems, ACM New York, NY, USA 2005.

[18]. http://www.willow.co.uk/html/telosb_mote_platform.html

[19]. http://en.wikipedia.org/wiki/TinyOS

[20]. Mitzenmacher, M.: Compressed Bloom Filters. IEEE/ACM Trans.Netw. 10, 604 (2002).

[21]. S. Hussain, F. Kausar, A. Mehmood “An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks,” International Conference on Communications and Mobile Computing Proceedings of the 2007 international conference on Wireless communications and mobile computing. Pages: 388 – 392. Honolulu, Hawaii, USA

[22]. X. Du and F. Lin, “Maintaining Differentiated Coverage in Heterogeneous Sensor Networks,” *EURASIP Journal on Wireless Communications and Networking*, Issue 4, pp 565–572, 2005.

[23]. Frank Stajano and Ross J. Anderson, “The resurrecting duckling: Security issues for ad-hoc wireless networks,” in *Seventh International Security Protocols Workshop*, 1999, pp. 172–194.

[24]. D. Ganesan, R. Govindan, S. Shenker and D. Estrin, "Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks," Mobile Computing and Communication Review (MC2R) Vol 1. No.2. 2002.