Spring 2011

CENTRALIZED SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS

Li Yang
San Jose State University

CENTRALIZED SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS

Presented to
The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Li Yang
June 2010

Li Yang has passed the defense for the project Centralized Security Protocol for Wireless Sensor Network.

Professor Melody Moh — Date: 2/24/2011

Professor Jon Pearce — Date: 2-24-11

Professor Mark Stamp — Date: 2/24/11

NOTE: The advisor should send the final report to the graduate coordinator so
that the student can be cleared for graduation

Acknowledgement

A special thank you Dr. Melody Moh for guidance throughout the project.

The Designated Committee Approves

CENTRALISED SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS

by
Li Yang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE
SAN JOSÉ STATE UNIVERSITY

June 2010

Dr. Moh Department of Computer Science

Dr. Pearce Department of Computer Science

Dr. Stamp Department of Computer Science

Abstract

Wireless Sensor Networks (WSN) is an exciting new technology with applications in military, industry, and healthcare. These applications manage sensitive information in potentially hostile environments. Security is a necessity, but building a WSN protocol is difficult. Nodes are energy and memory constrained devices intended to last months. Attackers are physically able to compromise nodes and attack the network from within. The solution is Centralized Secure Low Energy Adaptive Clustering Hierarchy (CSLEACH). CSLEACH provides security, energy efficiency, and memory efficiency. CSLEACH takes a centralized approach by leveraging the gateways resources to extend the life of a network as well as provide trust management. Using a custom event based simulator, I am able to show CSLEACH's trust protocol is more energy efficient and requires less memory per node than Trust-based LEACH (TLEACH). In terms of security, CSLEACH is able to protect against a wide range of attacks from spoofed messages to compromised node attacks and it provides confidentiality, authentication, integrity and freshness.

Table of Contents

List of Acronyms

ACK - Acknowledgments
BER - Bit Error Rates
CBC - Cipher Block Chaining
CDMA - Code Division Multiple Access
CH - Cluster Head
CM - Cluster Member
CSLEACH - Centralized Secure Low Energy Adaptive Clustering Hierarchy
CTT - Clusterhead Trust Threshold
DOS - Denial of Service
ECC - Elliptic Curve Cryptography
EKG - Electrocardiogram
KDC - Key Distribution Center
LEACH - Low Energy Adaptive Clustering Hierarchy
MAC - Message Authentication Code
MTT - Member Trust Threshold
NACK - Negative Acknowledgements
NSTT - Neighbor Situational Trust Table
SHT - Second Hand Trust
TCIV - Trust Check Initialization Vector
TC - Trust Check
TLEACH - Trust-based Low Energy Adaptive Clustering Hierarchy
TTP - Trusted Third Party
WSN - Wireless Sensor Networks

List of Figures

List of Tables

Chapter 1 : Introduction

A Wireless Sensor Network (WSN) is a collection of microcontroller devices designed to accumulate sensed data through wireless communication. Equipped with transceiver, microcontroller, memory, and battery, sensor nodes collect various forms of data from a sensor module. Early research focused primarily on energy efficient solutions. Recently, security is becoming as important a topic. WSNs have potential in medical, industrial and military applications. These applications have urgent need to protect confidential data. However, developing a secure WSN protocol is not easy. Sensor nodes operate in remote and sometimes hazardous environments inaccessible to humans. Nodes must function without renewable energy sources for months. Additionally, nodes may number in the thousands so slight changes to the cost of an individual node can cause dramatic changes in the overall cost of the network. As a result memory is limited. The purpose of this research is to develop a protocol capable of satisfying the needs for security, yet remain energy and memory efficient. The scope of this project will include researching WSN protocols, developing a new protocol, and analyzing the new protocol in terms of network performance, memory requirements, energy consumption, and most importantly security. Analysis of encryption algorithms is outside the scope of the project. The new protocol introduced is called Centralized Secure Low Energy Adaptive Clustering Hierarchy (CSLEACH).

So what are sensor networks? Sensor networks are networks of sensor nodes or motes capable of performing automated monitoring or detection. Motes are devices equipped with special sensor modules such as an electrocardiogram (EKG**)**, motion

sensor, or pressure sensor. Sensor nodes scattered throughout a region transmit data to a gateway (aka controller or base station). The gateway is responsible for organizing and transmitting data through the internet where the data is reaches a final destination for storage or processing.

Chapter 2 : Background and Related Studies

2.1 Security Attacks

WSN face unique set of security challenges **[30]**. WSN not only need confidentiality, authentication and data integrity, but trust as well. Nodes deploy in hostile environments where attackers can physically tamper with nodes. Nodes must be produced cheaply to be cost-effective; therefore nodes are severely underpowered compared to laptop class attackers. Below is an overview of potential attacks.

Hello Flood

The hello flood attacks nodes using a powerful transmitter by advertising routes to the gateway. Nodes receiving the message see the attacker as a nearby node with a short route to the gateway, but the attacker is actually outside the transmission range of most nodes. Neighboring nodes become confused when data sent to the advertised route disappear. The hello flood also works with replayed messages **[19]**.

Spoofing/Message Altering

Spoofed and altered messages are simple attacks that modify messages to confuse message recipients. Altered messages can spread false routing information to cause bad routing decisions. Bad routing in WSN translates to longer paths and wasted energy. This attack can be defeated by an integrity check such a Message Authentication Code (MAC).

Replay Attack

A replay attack captures and retransmits a message. Replay attacks are unaffected by encryption. A nonce or timestamp is necessary to counter replayed messages. Timestamps are preferred by WSN because they require fewer messages.

Sybil Attack

The Sybil Attack is a class of attacks that target trust based protocols. The Sybil Attack relies on the ability to forge or mimic node identifications in order to produce a large set of identifications to leverage a trust based system. By sending false trust messages from a large set of nodes, the attacker can reduce the trust of innocent nodes. Sybil is preventable with a key registration system.

Wormhole

A wormhole is a coordinated attack between two attackers capable of communicating through other means than the normal communication. An example would be two computers at opposite ends of the network, communicating through a different frequency. The attackers share information only available to the other node. The attackers then advertise a better route than the ones available, causing neighboring nodes to use the attacker as an intermediary hop. This attack sets-up other attacks such as selective forwarding.

Selective Forwarding

Selective Forwarding works when an attacking node places itself in the routing path of another node. The attacker then chooses which packets to forward to the next hop and which packets to drop. The most basic selective forwarding attack is a sinkhole. A sinkhole drops all arriving packets. Often routing protocols detect sinkholes as broken links and attempt to avoid the link.

Compromised Nodes

It is hard to imagine someone physically breaking into a home computer to attack the network, but this is the reality for WSN **[27]**. Imagine a sensor node deployed on the battlefield to detect enemy movement. Attackers have physical access to the deployed nodes. Once a node is compromised, the attacker has access to privileged information, such as keys. How do we distinguish which nodes are compromised? This is where trust protocols come in. Trust protocols have long existed for Ad-Hoc networks **[11][15]** . Many trust based protocols use monitoring similar to watchdog **[23]**. The watchdog monitors neighboring nodes for "misbehaviors" which are reported and evaluated. A neighbors trust value entry is used to determine whether a neighbor is part of a trusted route. Trust is often established through direct monitoring or distribution of trust tables called Second Hand Trust (SHT).

Trust based protocols are not attacker proof, rather they are best effort attempts at intrusion detection. Trust protocols often rely on special knowledge to determine "misbehaviors" which usually means knowing the definition for legal application data. Trust protocols are subject to myriad of problems, one of which is lying. Compromised

nodes can collude to victimize innocent nodes by passing false second hand trust values. Other problems include false positives and misdetections. Existing trust protocols for Ad-Hoc networks rely on flooding to distribute trust. Flooding is unsuitable for WSN because of the energy wasted with redundant transmissions. In the next section, we will see and example of a WSN trust based protocol.

2.2 Security Considerations in WSN

Existing WSN security protocols use variations of symmetric key, MAC and pre-distributed key schemes to provide confidentiality, data integrity and authentication **[18]**. The reason many protocols converge to similar solutions is because of the lack of alternatives.

Public key cryptography provides authentication and confidentiality. Asynchronous feature in public key is useful for distributing keys and for broadcast authentication. The high energy and processing overhead eliminates public key cryptography as an option. Elliptic Curve Cryptography (ECC) is a new way to do public key. ECC reduces key sizes while still providing the same level of security **[32]**. Unfortunately, ECC is still too computationally expensive compared to symmetric key cryptography. As a result block ciphers dominate majority of WSN protocols with extensive research into energy performance of block ciphers **[1][10][17]**.

Traditional key exchange protocols use public keys. Most WSN protocols resort to some form of pre-distributed keys **[7].** Pre-distribution schemes can be categorized as single key, pair-wise and random-key. In single key pre-distribution, all nodes in the network share a single key. If the single key is ever made public, the entire network is

compromised. In the basic pair-wise scheme, each node must store keys for n-1 neighbors. This approach requires large amounts of memory to store keys. In random key pre distribution, nodes are assigned a random subset of keys from a key pool. Two nodes are allowed to communicate if they have matching keys. It only takes a small subset of keys to compromise the entire network.

2.3 WSN Security Protocols

SPINS is a protocol developed to solve the particularly difficult WSN problem of broadcast authentication **[28]**. SPINS is built of two protocols called SNEP and μTesla. SNEP provides security between two nodes, while μTesla provides broadcast authentication using symmetric keys. SNEP uses block ciphers to encrypt messages in Cipher Block Chaining (CBC) mode. μTesla provides broadcast authentication using a delay strategy. μTesla begins with the gateway generating a key chain by continuously applying a hash function and reversing the order of the keys. Each node entering into the network must be bootstrapped with a key in the keychain. The bootstrapped key is a commitment to the key chain because subsequent keys can be authenticated with repeated applications of the hash functions to return to the initial key value. The network is synchronized by intervals to which a new key is bound to. Packets send during an interval contain a MAC encrypted with the intervals key. After each interval, the gateway releases another key. A node can validate the key by applying the hash function to obtain the previous rounds key.

μTesla does have its flaws. Because nodes must buffer data before keys are revealed, attackers can send random messages to overflow the nodes buffer. The receiving node is unable to determine which messages are from the gateway until the key is revealed.

2.4 LEACH-based Protocols and Security Enhancements

The basics of security are confidentiality, data integrity, and authentication, but in the world of WSN, energy is always the first priority. Early protocols prolong the operating lifetime of a network with clustering, multihop, and energy aware routing **[6] [8]**. These strategies focus on reducing transmission costs because transmission energy increases exponentially with distance. While these protocols are not designed for security, they do provide a useful energy efficient template to develop a new protocol. Numerous low energy protocols exist, but we will turn our attention to one specific protocol, LEACH.

LEACH

Low Energy Adaptive Cluster Hierarchy (LEACH) is amongst one of the earliest energy efficient protocols developed for WSN **[13]**. LEACH is organized into the three stages cluster set-up, schedule creation, and data transmission (aka steady state). Nodes form clusters under a cluster head (CH). A CH is responsible for coordinating transmission schedules and aggregating data. LEACH elects CHs by probabilistically self electing nodes. Candidates advertise their candidacy to neighboring nodes. Non-CH nodes select the closest CH based on the strongest signal strength. Non-CHs respond with a cluster join message to become cluster members (CM). CH is responsible for

organizing CMs by providing a time schedule. Once clusters are organized, each cluster can simultaneously collect sensor data from its members. This is possible with different code division multiple access (CDMA) codes. A CH aggregates the data before sending it to the gateway. Data aggregation saves energy by compressing data before transmission.

LEACH Based Security Protocols

SC-LEACH is a LEACH based protocol designed to optimize LEACH by fixing the fundamental problems related to random CH election. SC-LEACH uses a pre-distributed key ring that is used to coordinate secure communication between a CH and CM. SC-LEACH uses symmetric key cryptography along with a nonce to protect against replay attacks **[16]**.

Sec-LEACH uses random key pre-distribution scheme to coordinate clusters **[26]**. A key pool of randomly generated keys and ids are generated at the start of the network. Nodes are assigned a string of keys selected by a pseudo random number generator. Each node is also assigned a pair-wise key shared with the gateway. Nodes join clusters to which they share a common key. Armor LEACH is another security protocol based on Sec-LEACH **[2]**.

TLEACH

TLEACH is a WSN trust protocol **[31]**. TLEACH contains two main components, the Monitoring Module and the Trust Evaluation Module. Each node also maintains a Neighbor Situational Trust Table (NSTT) filled with trust value entries for each pair of node ids and situational operations. Situational operations, such as data sensing and

routing, each have an individual trust value because nodes may not behave maliciously for all operations. The Monitoring Module is responsible for detecting a neighbor's “misbehaviors”. The Trust Evaluation Module evaluates which actions are safe to take based on NSTT trust values. Like LEACH, clusters are formed through self election. Instead of signal strength, TLEACH selects a CH based on the CH candidate with the highest trust value. TLEACH’s transmission period is separated into multiple turns with each turn ending in a trust update slot. During an assigned transmission timeslot, CMs transmit data to their CH. When a node is not transmitting, the node probabilistically determines if it will monitor a transmitting neighbor. Whenever a monitoring node detects misbehaviors, the Monitoring Module files a misbehavior report tallying the number of misbehaviors and good behaviors. The trust update slot allows the CH to share its trust values with its CMs in a SHT message. Nodes update their NSTT with the SHT and the misbehavior reports.

Chapter 3 : CSLEACH

In this paper I introduce a LEACH based security protocol called Centralized Secure LEACH. The motivation behind this project is the need for security and resource efficiency in a WSN protocol. When building a WSN protocol, it is understood the gateway cannot be compromised because the network cannot function without a single point to collect data. Additionally, the gateway is unique because the gateway can be more resource abundant than a sensor node. These resources include a rechargeable battery, larger memory and greater processing power. To take advantage of these features, CSLEACH utilizes the gateway for key management, and trust management. CSLEACH builds on the LEACH algorithm by adding authentication, confidentiality, integrity, freshness and trust. Like LEACH, each sensor node is able to directly transmit to the gateway. Using a Key Distribution Center (KDC) approach, each node shares a unique private key with the gateway. CSLEACH uses single key pre-distribution to share a gateway private key that is used for broadcast authentication.

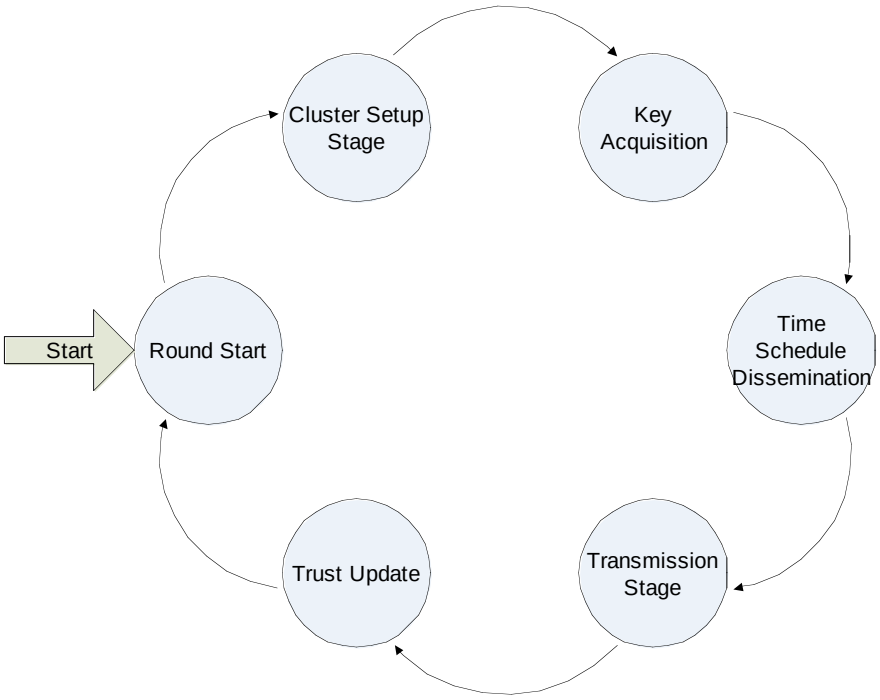

Figure 1 CSLEACH state diagram.

CSLEACH is separated into the stages Round Start, Cluster Setup, Key Acquisition, Time Schedule Dissemination, Transmission Stage, and Trust Update. Below are detailed descriptions of each stage.

3.1 CSLEACH Details

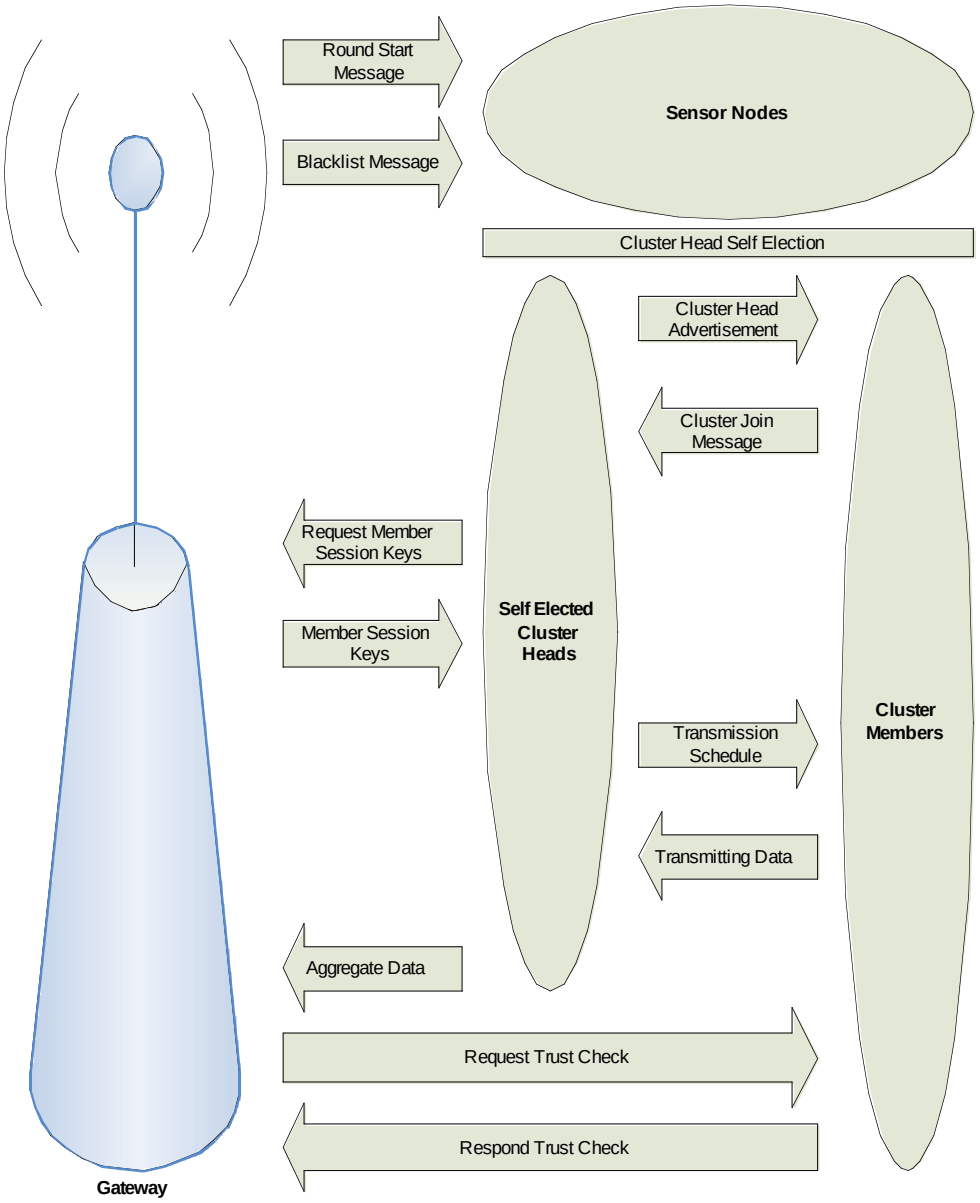

Figure 2 CSLEACH Message Transmissions

CSLEACH is organized into periods called rounds. At the start of each round, session keys are distributed to prevent stale keys. Sensor nodes each possess two permanent keys, a Gateway Private Key ($\mathbf{K_{CTRL}}$) and a Node Private Key ($\mathbf{K_P}$). As nodes initially enter the network, they enter in a receiving state. Nodes wait patiently for a message from the gateway which indicates the beginning of a round.

Round Start

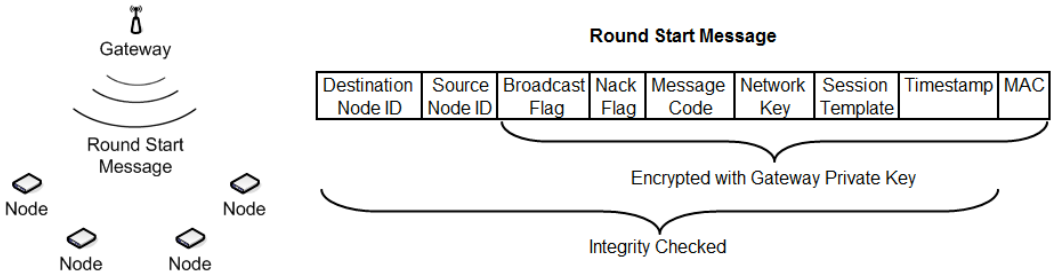

Figure 3 Round Start Message Frame

Each round is triggered by a Round Start Message from the gateway which functions as a synchronization message. The message distributes a Session Template (**T**) and an Network Key ($\mathbf{K_N}$) used to cheaply produce Session Keys and MAC Keys. The Session Key ($\mathbf{K_S}$) encrypts communications between nodes and the gateway, and communications between a cluster head (**CH**) and a cluster member (**CM**). The MAC Key ($\mathbf{K_{MAC}}$) is used to encode a MAC to provide integrity protection.

$$\mathbf{K_S = HMAC\ (T, K_P)}$$

$$\mathbf{K_{MAC} = K_S \oplus K_N}$$

T is hashed with a HMAC using the key $\mathbf{K_P}$. As long as the $\mathbf{K_P}$ is kept safe, a new session key can be generated each round. Similarly, a Gateway Session Key ($\mathbf{K_{GS}}$) can be produced by hashing **T** using $\mathbf{K_{CTRL}}$ All messages contain a timestamp to prevent replay attack.

The Round Start Message is unique because the message must first be decrypted before the integrity of the message can be validated. This is because the K_{MAC} for each round is unique, and a new K_{MAC} depends on K_N which is part of the Round Start Message.

Cluster Setup Stage

Once the network synchronizes using the Round Start Message, nodes enter the Cluster Setup Stage.

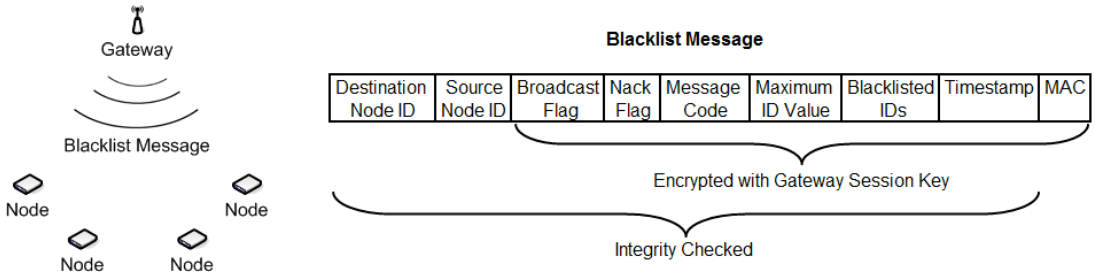

Figure 4 Blacklist Message Frame

The gateway first distributes a blacklist to warn nodes of malicious or faulty nodes. The blacklist is used to reject malicious nodes from becoming Cluster Heads (**CH**). The blacklist message contains a Maximum ID Value and a list of blacklisted nodes. Nodes entering into the network have sequential IDs. The Maximum ID Value is used by nodes to reject any ID with a greater value which allow nodes to reject invalid IDs. The black list message is encrypted with the $\mathbf{K_{GS}}$ which prevents older blacklists from being replayed. It is important to note the advertisement message is encrypted with the $\mathbf{K_{GS}}$ to prevent nodes outside of the network from spoofing as CHs. The blacklist exists to prevent compromised nodes from becoming a CH based the nodes reputation. The blacklist does not prevent a compromised node from spoofing another node.

Once nodes receive the blacklist, nodes self elect to become cluster head. Nodes elect by generating random numbers and following the same formula outlined in LEACH. Nodes elected as cluster head advertise their candidacy to neighboring nodes.

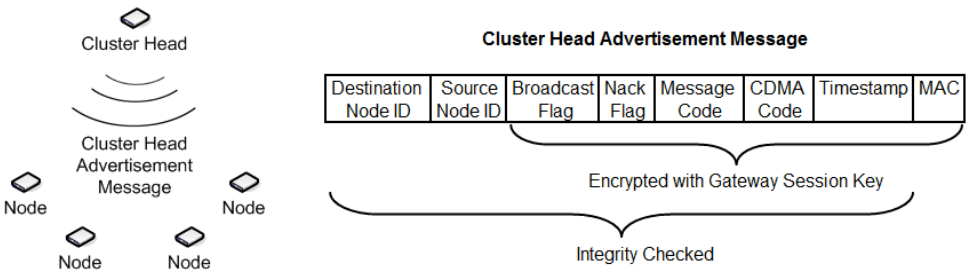

Figure 5 Cluster Head Advertisement Message Frame

The advertisement message contains the CH's preset Code Division Multiple Access (**CDMA**) code which enables clusters to communicate without interfering with neighboring clusters. The remaining nodes select a CH based on a CH candidates signal strength, and reject nodes listed by the blacklist.

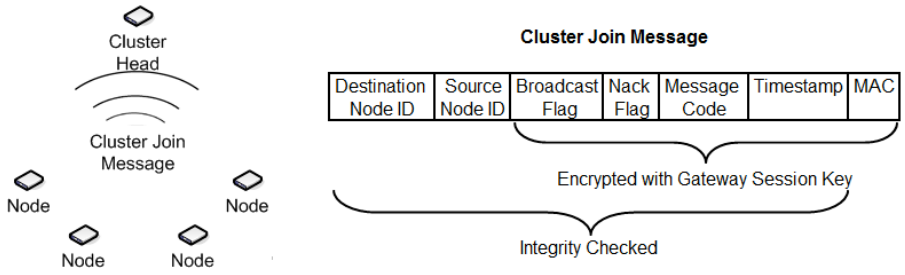

Figure 6 Cluster Join Message Frame

Nodes become CM by responding to a chosen cluster head with a join message.

Key Acquisition

Before a CM can begin transmitting data to a CH, the CH must acquire $\mathbf{K_S}$ for its members to ensure data confidentiality. To obtain each CM's $\mathbf{K_S}$, the CH compiles a list of CM IDs in a Member Session Key Request Message.

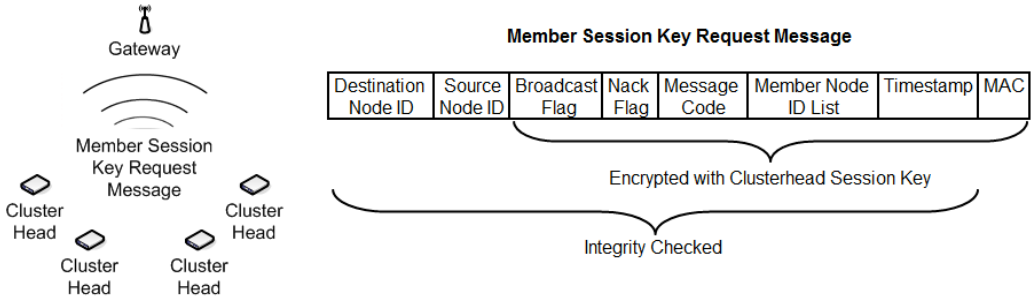

Figure 7 Round Start Message Frame

During this stage, the gateway can associate CMs with CHs. The associations allow the gateway to scan for duplicate IDs and to select CM for Trust Checks from each cluster. If a CM has insufficient trust, the gateway can withhold supplying a $\mathbf{K_S}$ to prevent the CM from communicating with the CH.

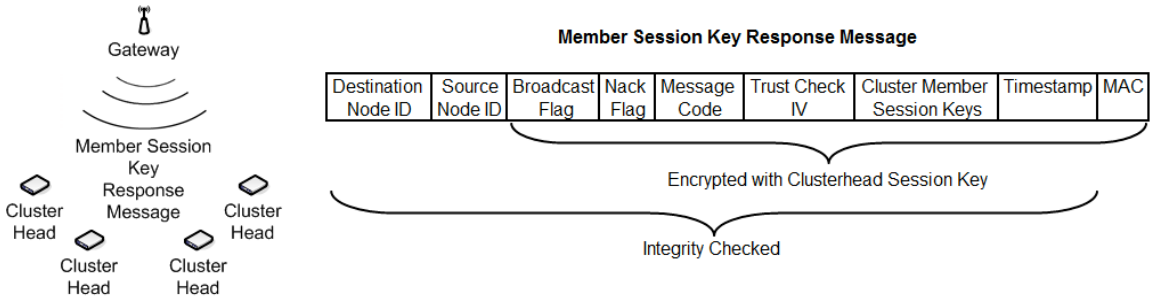

Figure 8 Member Session Key Response Message Frame

Once the keys are compiled, a response message is sent containing a list of session ids, session keys, and a Trust Check Initialization Vector (TCIV). The TCIV will be used produce a MAC called the Trust Check (TC). The entire key response message is encrypted using the CHs private key.

Time Schedule Dissemination

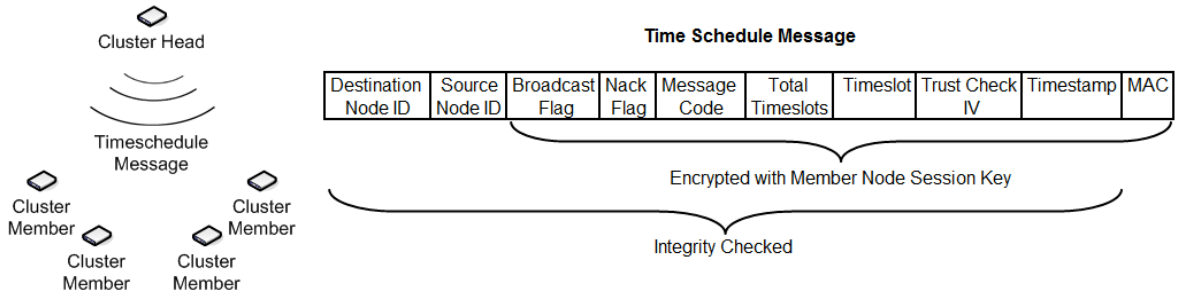

Figure 9 Time Schedule Message Frame

At this point in the protocol, a CH will have a list of session keys and TCIVs. The CH is responsible for coordinating the Time Division Multiple Access (**TDMA**) section of the protocol. The CH is responsible for assigning timeslots for CM to transmit their sensor data to the CH. For every node the gateway does not provide a $\mathbf{K_S}$, the CH will not be able to transmit a time schedule to that node. Nodes that do not receive a time schedule will no longer participate in the protocol and must wait for the next round. Accepted members are assigned timeslots designating when a node can transmit.

Transmission Stage

CSLEACH partitions transmission periods into turns, as seen in TLEACH. Each CM will transmit one timeslot each turn. After a turn is complete, the next turn begins until a preset number of turns are reached. Greater turns equates to smaller round setup overhead per transmission and conversely more memory required by the CH. In addition to transmitting the sensor data, it is the responsibility of the CM to maintain a MAC of all of its transmission for a given round. This MAC is called a Trust Check (**TC**). The MAC produced uses the TCIV given by the gateway. The TCIV must be unique each round

because a nodes private key is used to encode the TC. If nodes transmit predictable patterns of data, and the same TCIV is used each round, then the encrypted data could fall victim to known plaintext attacks.

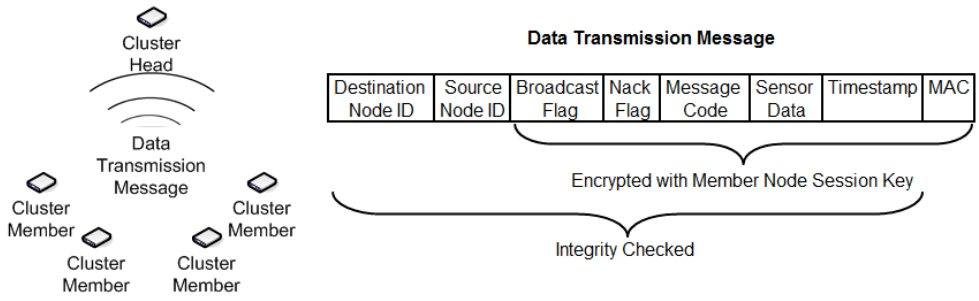

Figure 10 Data Transmission Message Frame

Once the transmissions are complete, the CH will aggregate data and send the data to the gateway. The data aggregation must be lossless to ensure the gateway is able to retrace the source node ID of sensor data. This is important for the gateway to produce a MAC to compare to the TC produced by a CM.

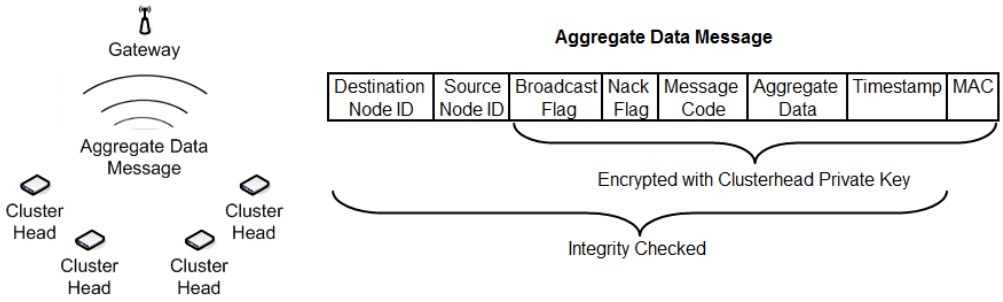

Figure 11 Aggregate Data Message Frame

Trust Update

After a round, the gateway must evaluate the performance of each node. The gateway is able to reproduce a TC for each CM based on the data received by each CH.

To verify if the TC is correct, the gateway selects CMs to sample TCs. The trust selection probability determines how many CMs are selected by the gateway.

 # CM selected per cluster = trust selection probability/CH election probability

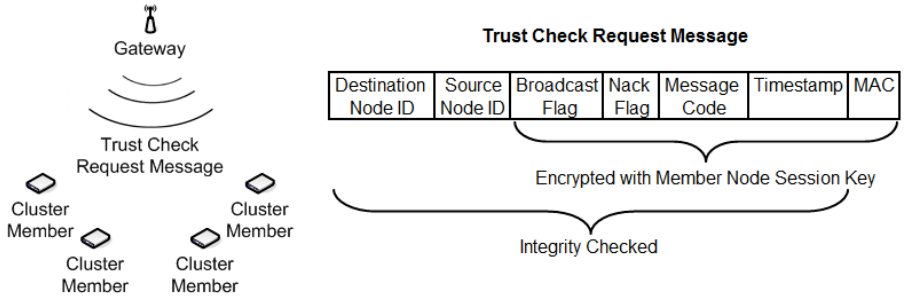

Figure 12 Trust Check Request Message Frame

The gateway sends a TC request message to randomly selected CMs. The gateway also computes a TC value from the aggregate data for the selected node. If the TC from the node mismatches the TC from the gateway, both the CH and the selected CM are punished.

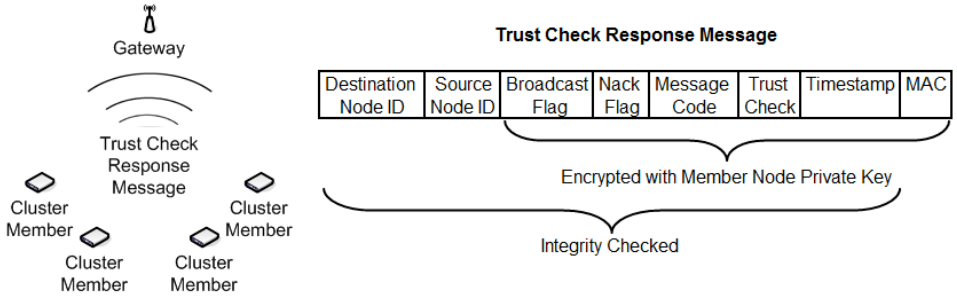

Figure 13 Trust Check Response Message Frame

Keys	Owner	Origins	Users	Purpose	Usage
Gateway Private Key	Gateway	Randomly Generated	Gateway, Registered Nodes	Prevent nodes not registered with the gateway from participating in communications.	To Encrypt the Round Start Message.
Gateway Session Key	Gateway	HMAC (**T**, Gateway Private Key)	Gateway, Registered Nodes	Prevent nodes not registered with the gateway from participating in communications and to prevent overexposure of the Gateway Private Key	To Encrypt the Blacklist Message, Cluster Head Advertisement Message, Cluster Join Message.
Node Private Key	Node	Randomly Generated	Node, Gateway	Protects communications between a node and the gateway from the attacks from a compromised Cluster Head.	To Encrypt Trust Check Response Message, Aggregate Data Message.
Node Session Key	Node	HMAC (**T**, Node Private Key)	Node, Gateway, Cluster Head, Cluster-Member	Protects communications between a Cluster Head and Cluster Members, from unregistered nodes and compromised nodes.	To Encrypt Member Session Key Request Message, Member Session Key Response Message, Time Schedule Message,. Trust Check Request Message
Mac Keys	Node	<Encryption Key> $\oplus$ $\mathbf{K_N}$	Node, Gateway, Cluster Head, Cluster-Member	Integrity protect messages.	To Integrity check the Member Session Key Request Message, Member Session Key Response Message, Time Schedule Message, Trust Check Request Message.

Note:

Network Key ($\mathbf{K_N}$) and Session Template (**T**) are components of the Round Start Message.
Node Keys are designated Cluster Head Keys and Member Keys dependant on the current role of the node.

Table 1. CSLEACH key table.

Trust mechanism

CSLEACH employs a trust mechanism specifically catered to the unique relationship between CHs and CMs. As an intruder, the role of CH is very salient because CHs are responsible for routing data from CMs. CSLEACH uses two thresholds termed Clusterhead Trust Threshold (CTT), and the Member Trust Threshold (MTT). Trust is scaled between 0 and 100 and nodes begin with a trust value of 100. Nodes with trust above the CTT are privileged to become a CH. Nodes with trust above the MTT are allowed to participate as CMs. Nodes with trust below CTT have likely experienced communication problems and are at risk of dropping packets. These nodes are blacklisted from becoming clusterhead. Nodes below the MTT are absolutely untrustworthy nodes that are blacklisted and rejected from any session key requests. For nodes that cross the MTT into the lowest trust region, their trust is automatically assigned zero trust. The CTT must be much greater than the MTT to ensure the CH is able to reliably forward sensor data. The CTT must be set strictly based on the noise level of the environment, whereas the MTT can be more freely set based on how strict the network should scrutinize suspicious transmission behaviors.

Trust Punishment

When TC validation fails, there are three possibilities. The first possibility is the CH is omitting or modifying data. The second possibility is the CM lied on the trust check. In the first two cases both the CH and the CM must both be punished because it is impossible to determine who the offender is. The reasoning behind the punishment scheme is the assumption that attackers are likely repeat offenders. A CH should be

punished more severely because of the low probability of becoming CH, and the greater potential for harm as a CH.

The third possible case is a faulty transmission. Wireless communications are subject to interference causing bit errors. Any errors during the transmission period will cause a mismatch in the TC. There are a few things that should be done by a media access control protocol to remedy this problem. The protocol must provide a robust retransmission scheme**.** Acknowledgments can become security risks as Wagner points out **[19]**. CSLEACH helps faulty nodes by gradually recovering trust between rounds if a node is accidentally punished. Redemption protects faulty nodes from becoming exiled from the network for temporary interference.

CSLEACH's trust protocol is configurable by adjusting the CTT value, MTT value, CM and CH punishment values, and recovery value. All values range from 0 to 100. The recovery value should be much smaller than the CM and CH punishment otherwise trust punishments will have no effect. The default configurations are CH punishment of 15, a CM punishment of 10, and recover value of 1. The CTT is set to 60 and MTT is set to 30.

4.2 Protocol Comparison

	LEACH [13]	**TLEACH [31]**	**TLEACH (Simulated)**	**CSLEACH**
Integrity	None	None	MAC	MAC
Authentication	None	None	Pre-distributed keys	Pre-distributed keys
Confidentiality	None	None	Symmetric Key Encryption	Symmetric Key Encryption
Trust	None	NSTT / Monitoring Neighbors	NSTT / Monitoring Neighbors	TC

Table 2. Protocol security comparison.

LEACH is not a security protocol, but it serves as a performance standard for both TLEACH and CSLEACH. LEACH has the least amount of overhead and memory requirements, but lacks in any security.

TLEACH is a purely trust based protocol intended to be coupled with other security protocols designed to provide integrity, authentication and confidentiality. TLEACH relies on message passing to distribute trust information amongst nodes which translates to transmission overhead. For the purpose of comparison, TLEACH is modified to adopt CSLEACH's key distribution mechanism which provides integrity, authentication, confidentiality and freshness. The modified protocol is used as a comparison against the efficiency of CSLEACH's trust protocol.

CSLEACH relies on the gateway as its TTP (trusted third party). CSLEACH communicates keys through encrypted messages between a CH and the gateway. Since LEACH is a two hop protocol, CSLEACH can use the gateway to detect errors and attacks against forwarded data. The gateway needs greater memory capacity to store and maintain trust table and key information. The gateway may cause scaling problems especially during blacklisting and key requests.

Chapter 4 : Performance Evaluation

Simulations were performed on the protocols LEACH, TLEACH, and CSLEACH. A custom event base simulator was built to support memory, energy and performance analysis. For fair analysis and comparison, LEACH and CSLEACH were modified to adopt TLEACH's multi-turn transmission stage. Additionally a similar key scheme used in CSLEACH is adapted to TLEACH. Initial transmission rate is set to 20,000 b/s. Each round consists of 3 transmission turns, .6s transmission timeslot per node and 1024 bytes of data per packet transmitted. The cluster head percentage is set to 5 percent with a maximum simulation time of 10 hours. Nodes are enclosed in a 100m by 100m region. The battery is set to 100 Watt-hours or 3600 Joules. Encryption and decryption are both set to 3 micro joules per bit as data is encrypted using XTEA **[35].** Transmission and reception is simulated based on the first order radio model as seen in the LEACH paper. Transmission and reception components consume 50 nJ per bit and 100 pJ/bit/m^2 of transmission amplification. TLEACH requires knowledge to determine what is considered legal data in the application layer. The simulation assumes the sensed data is legal if the data blocks form 32 bit blocks representing integer values less than 100. Simulations output results in terms of good data, bad data, lost data and total data.

4.1 Security Evaluation

External Attack Analysis

External attacks were graded based on the ability of the attack to introduce bad data into the network. Bad data is data from any attacker accepted by the gateway. If any bad data is received by the gateway, the attack has succeeded. The first attack simulates an attempt for the node to enter a cluster and transmit random data. LEACH failed as the bad data was received by the gateway without incident. Both CSLEACH and TLEACH prevented the attacking node from becoming a CH or CM because the bogus id provided by the attacker was outside the maximum blacklist range. If the attacker opted to become a CM, the gateway was unable to find a valid session key for the unknown id. The Sybil attack would also fail because messages are encrypted with keys registered with the gateway. An invalid key would prevent a CH from communicating with the attacking node, or the attacking node from communicating with the gateway to obtain CM keys. The failures of these attacks to join a cluster indicated subsequent attacks attempting similar feats would fail as well.

Clock skew

Next we turn our focus to replay attacks against the start message. Clock management and synchronization is a tricky issue. The clock skew should last no longer than the time it takes to perform 1 round. Even with a 1 round clock skew, it is possible to replay data transmissions between two turns during the transmission stage. The problem is exacerbated by the possibility some clusters may contain only one CM which reduces the time between turns.

The replay attack simulated steals and resends a start round message periodically. Unsynchronized nodes searching for round start messages are forced to synchronize with

the attacker. By extending the time after rounds before the controller sends a start message, it is possible for the replayed start message to force all nodes into processing clusterhead setup before the gateway sends its round start message. The replayed message causes the network to become out of synch with the gateway. If the clock skew is reduced and managed properly, the likelihood of a successful attack diminishes.

A point of interest is the how the network synchronizes. If the round start message is used to synchronize the messages, then what is used to validate the timestamp on the start message? This is a chicken and the egg problem where we choose to either protect the start message against a replay, or we use the time in the message to synchronize our clock.

Compromised Node Attacks

Various attacks were simulated against TLEACH and CSLEACH to test the effectiveness of the trust mechanisms. Simulations were run with 50 total nodes with 10 percent compromised.

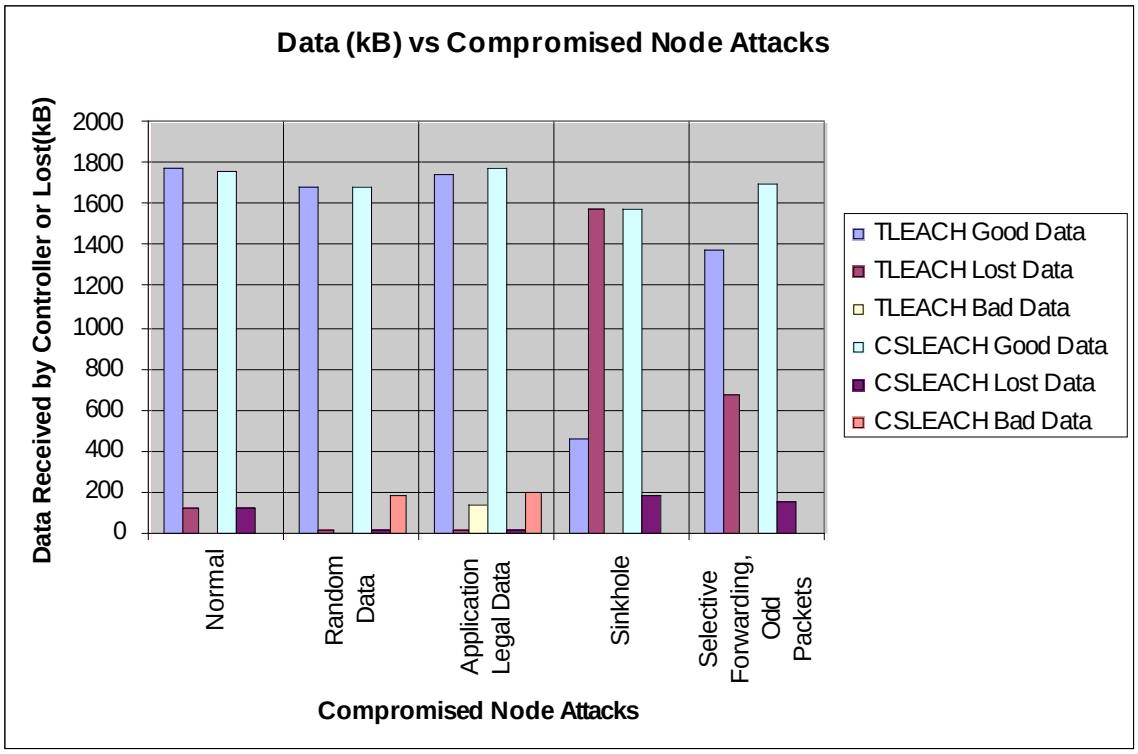

Figure 14 Compromised node attacks and the effects on data transmission.

Under normal conditions both TLEACH and CSLEACH receive high percentage good data.

Random Data

Random data is sent by compromised nodes to the CH. Compromised nodes behave normally when assuming the role of CH. TLEACH peer monitors illegal data by reporting misbehaving nodes. CH assisted monitoring enables the CH to remove data sent from compromised nodes. Unlike TLEACH, CSLEACH is a pure media access control protocol unaware of application data rules. Roughly 10 percent of the network data is bad data which means CSLEACH failed to prevent any of the falsified application data from reaching the gateway.

Random Application Legal Data

From the perspective of an intelligent attacker, the attacker could simply follow the application rules and introduce application legal random data. Application legal data is data that is indistinguishable from normal data when scanned by TLEACH's monitoring module. Both TLEACH and CSLEACH accepted bad data because neither could tell the difference between the bogus data and the actual sensor data. The overall amount of good data decreased as the network is burdened by the attacking nodes attempting to transmit

Sinkhole

For an attacker, the CH is a more attractive target than just sending bogus data as a CM. The following two attacks are variations of selective forwarding. In the sinkhole attack, attackers assume the role of CH every round. Compromised nodes drop all data received. TLEACH performs poorly against sinkhole attack because no monitoring is performed on CH transmissions. Conversely, CSLEACH is almost unaffected by attackers. Each time the sinkhole attack is performed, the CH fails a TC validation. Since the CMs outnumber the CH, the CH is punished more harshly causing the CH to quickly lose trust.

Selective Forwarding: Odd Packets

In the last attack, we attempt to forward odd packets received by the CH in an attempt to confuse the trust protocols. TLEACH losses less data because half of all data sent by CMs is received by the gateway. TLEACH is however unable to stop the constant

loss of data because of the lack of monitoring on the CH. CSLEACH performs almost as well against selectively forwarding odd packets as against a sinkhole. The TC selection process forces the CH to guess which nodes the gateway will select. If half of the packets are dropped, the gateway has a 50% chance to punish a CH for every CM selected. If two nodes are selected from each cluster, the CH has a 25% chance of escaping TC validation.

4.2 Throughput Evaluation

There are a few problems when comparing throughput for the LEACH based protocols. The goal is to obtain a throughput representing optimal conditions. In order to optimize throughput, the maximum allowed time for each stage must be minimized. The problem lies in the randomness of CH election. The random nature of CH election does not guarantee a constant number of CHs per round and therefore some rounds have fewer CHs resulting in more CMs per cluster. In order to prevent large clusters from surpassing stage limits, the maximum number of CMs is limited to twice the expected number of CMs per cluster. A portion of CM candidates are dropped from a cluster if the cluster reaches maximum capacity. Since each node can possibly reach twice the expected cluster size, extra time must be allocated to the transmission stages incase a cluster of maximum capacity exists. This means nodes are sleeping for long durations for smaller clusters. This explains the high variability in results between each of the three protocols.

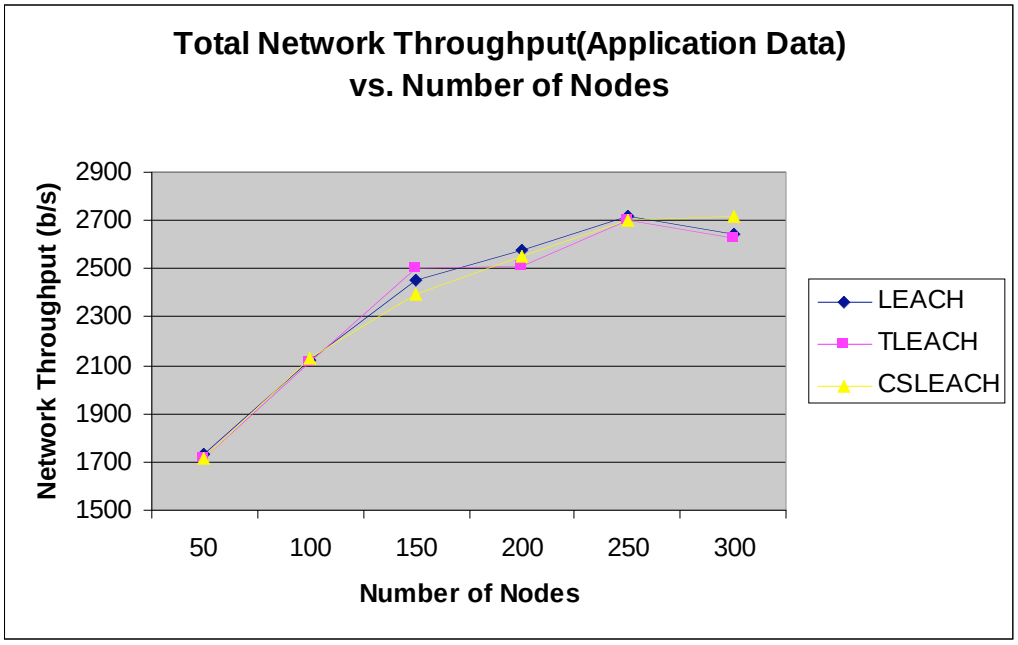

Figure 15 Throughput versus number of nodes.

Figure 3 shows the network throughput peeks around 2500 b/s. This is only a fraction of the 20,000 b/s throughput available for the network. Majority of the time is spent setting up clusters, key management, and other coordinating tasks. The network begins to peek at 250 nodes where the network throughput begins to drop indicating the difficulty LEACH based protocols have when scaling.

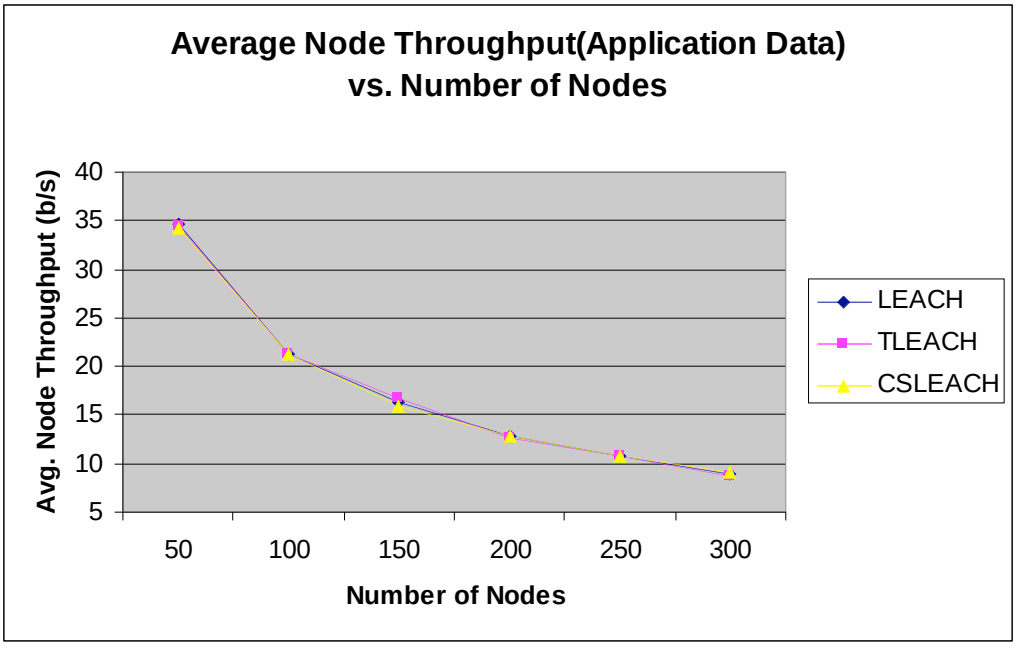

Figure 16 Average node throughput versus number of nodes.

Figure 4 shows despite the increase in network throughput, the number of CM per cluster increases at a much higher rate. As a result nodes are assigned shorter transmission times. LEACH, TLEACH, CSLEACH produced similar throughput because the election processes produces variable throughputs each round.

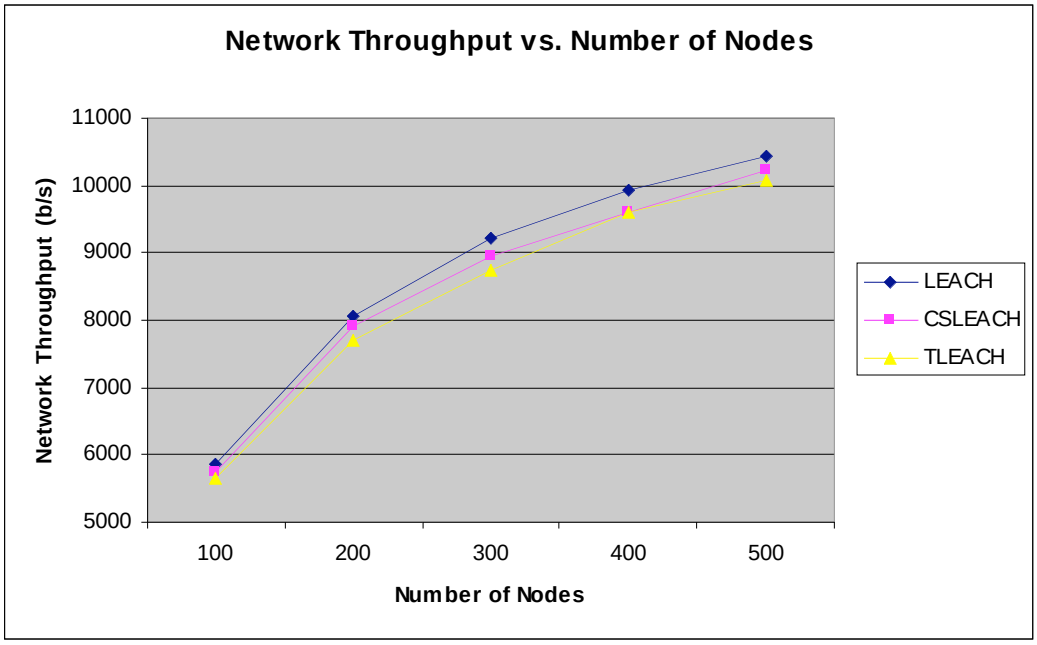

Figure 17 Network throughput vs. number of nodes under optimal conditions.

Under optimal conditions, the CH election percentage elects the same number of CHs each round. With optimal conditions, the maximum duration for each stage can be more accurately bound. In figure 5, LEACH transmits over 100 b/s more than TLEACH and CSLEACH for networks sized 300 and more. The increased throughput is a significant portion of the 20,000 b/s maximum transmission rate. CSLEACH performs marginally better than TLEACH.

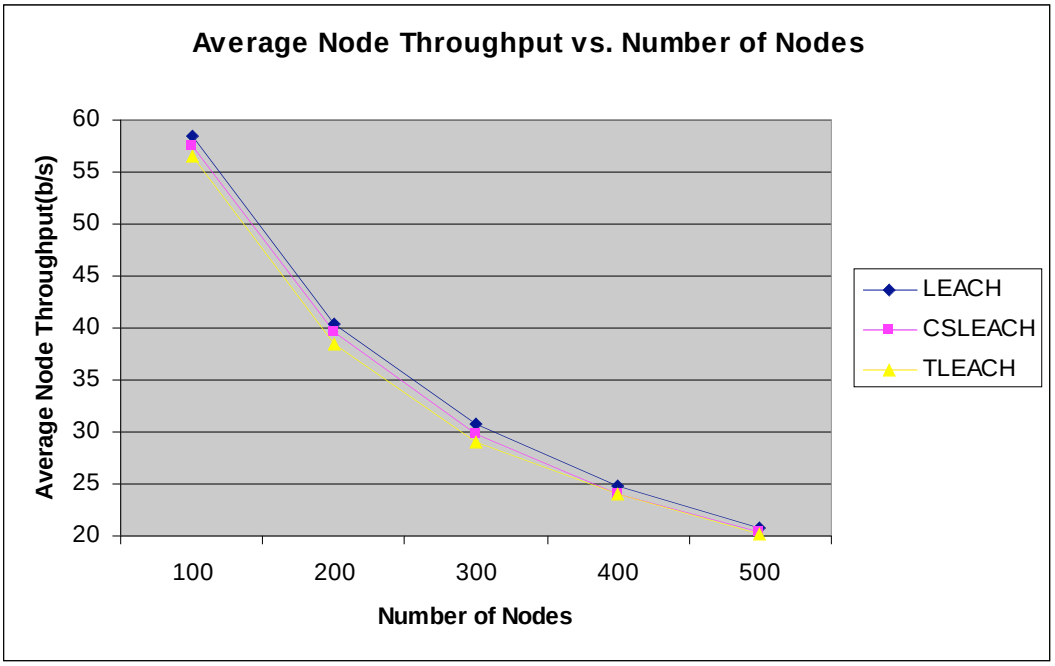

Figure 18 Average node throughput vs. number of nodes under optimal conditions.

Similarly, CSLEACH has a higher average node throughput than TLEACH. The differences are so minute that variations in real performance may be unnoticed.

4.3 Memory Evaluation

WSN is a cheap solution to automated monitoring. Simulations record the maximum memory needed at a sensor node and gateway. The simulator conservatively approximates the number of bytes required by each protocol. ROM and memory for encryption are excluded. Note the same maximum cluster size is limited to twice the expected cluster size.

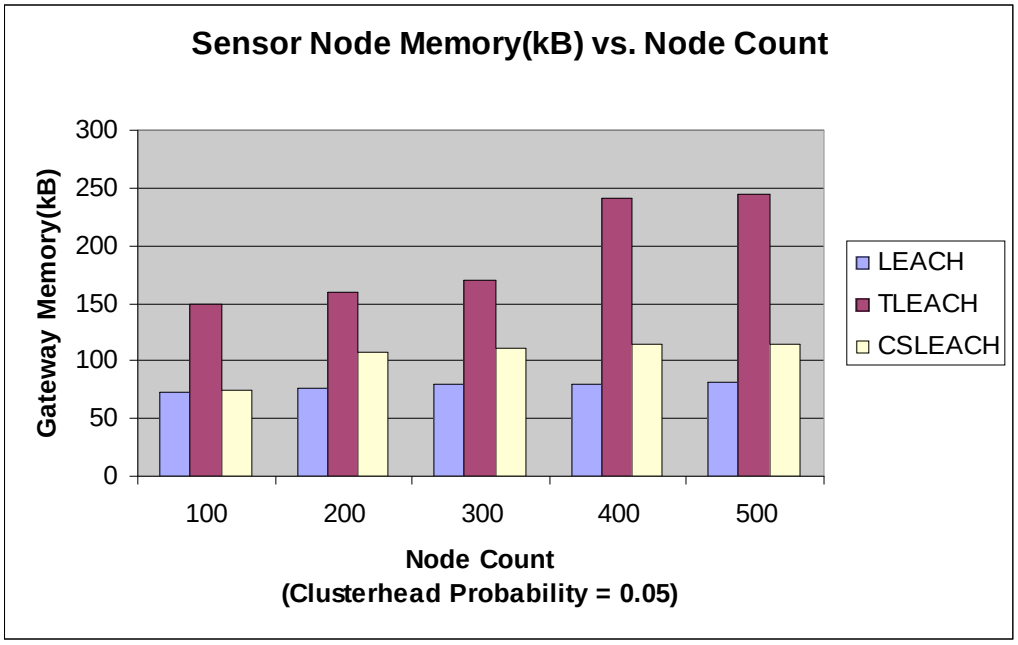

Figure 19 Minimum memory required on sensor node.

In figure 7, TLEACH requires roughly twice as much memory as LEACH and CSLEACH. The extra memory is used in the NSTT to store trust values. As the node sizes increase, memory required increases. The NSTT not only stores trust for neighboring nodes, but also nodes from second hand trust updates. CSLEACH requires slightly more memory than LEACH to store keys.

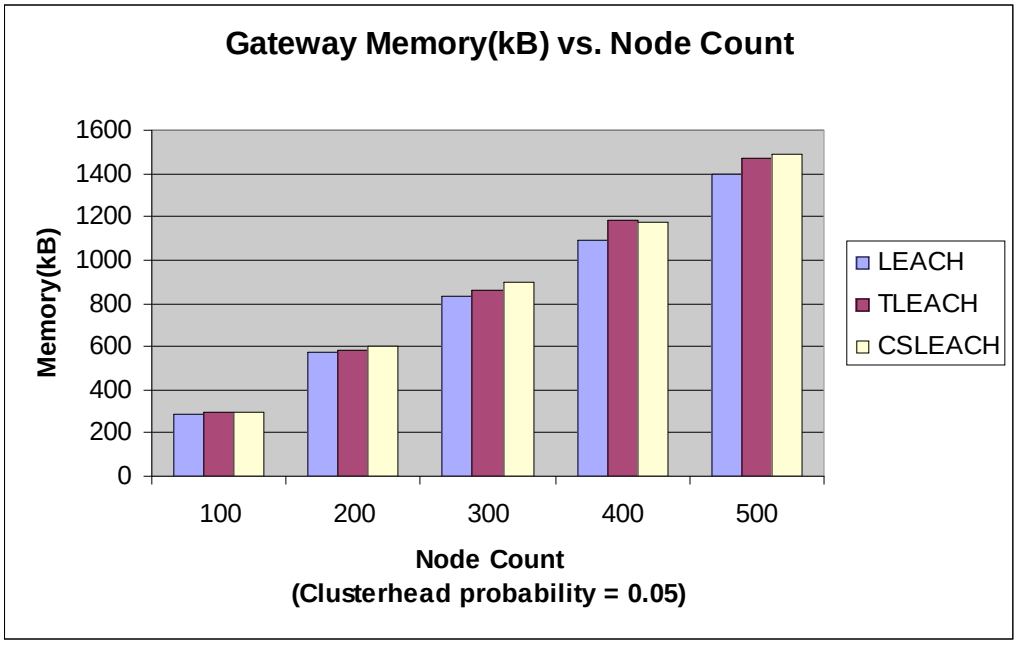

Figure 20 Minimum memory required on gateway.

Figure 8 shows CSLEACH with the highest required memory capacity on a gateway node. CSLEACH uses extra memory to store trust values for each node. TLEACH and MYLEACH both store keys at the clusterhead accounting for the extra memory over LEACH. The CSLEACH approach reduces the overall cost to WSN compared to TLEACH because only the gateway is required to store trust values whereas TLEACH reproduces trust tables for each node. From a cost perspective, CSLEACH is far cheaper to implement because of the lower memory requirements.

4.4 Energy Evaluation

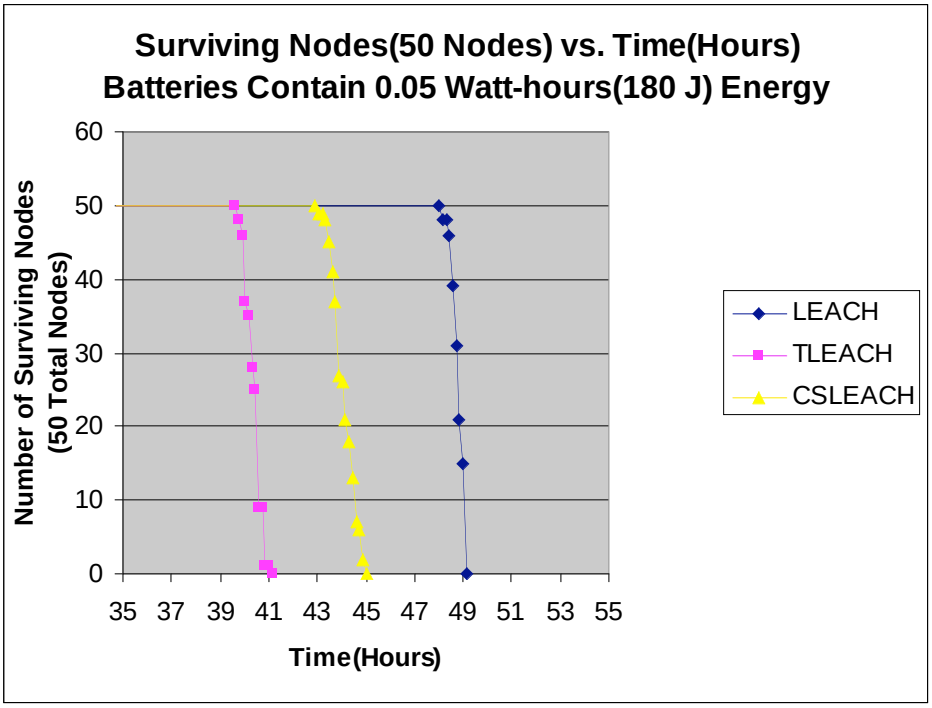

Figure 21 Number of Surviving nodes vs. time.

Energy consumption is the most important gauge to determine if a protocol is suitable for WSN. Figure 9 is the result of simulating 50 nodes with 0.05 watt-hour or 180 Joules of initial battery energy. Nodes experience energy drain during transmission, reception, encryption and decryption. Figure 9 shows the number of total dead nodes in the network with the passage of time. The two security protocols consume energy at a much higher rate than LEACH because of the added energy drain from encryption, decryption and frame overhead. CSLEACH and TLEACH implementations essentially use the same key distribution and encryption methods, therefore the difference in power efficiency is purely due to the trust management protocols. The simulation does not address the internal processing energy of each protocol so actual performance may vary. With that said, TLEACH is expected to expend more energy with the extra overhead

from updating SHT. CSLEACH takes a different approach by spending more time performing internal processing to produce a TC.

4.5 Error Tolerance Evaluation

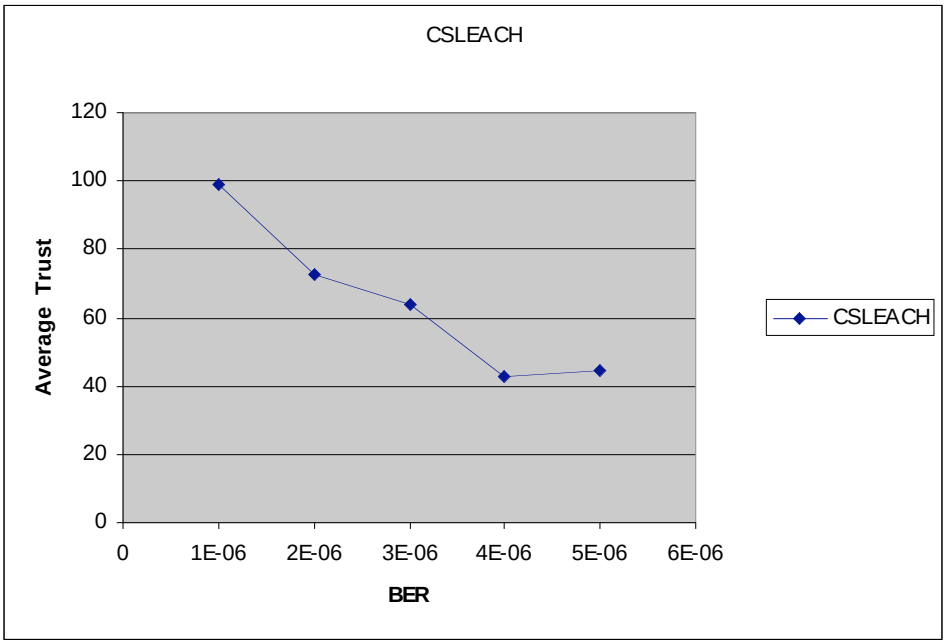

Figure 22 Effects of increasing BER on CSLEACH.

CSLEACH is not without its problems. Wireless communications is subject to bit errors caused by collisions and interference. Not all data is likely to reach a CH without alteration. If a robust retransmission scheme is in place, many distorted messages can be fixed, however there are many cases where this is not possible. The simulator uses negative acknowledgments (NACK) to retransmit data. There are risks to negative acknowledgements. If the source address in the frame header is lost, retransmission is impossible. Another possibility is losing data between stages. If a transmitting node sends a corrupted message during the end of the stage, there will not be enough time for a retransmission. The benefit of a NACK is the reduced volume of ACK transmissions.

Figure 10 shows the effects of increasing bit error rate (BER) has on the overall trust of the network. CH often lose trust quickly, but only until trust falls below the CTT. Once trust is below the CTT, the node is restricted to the role of a CM. CMs are far less likely to be punished, giving the node time to recover. For a high BER, the protocol protects nodes from falling below the MTT. For an attacker, it is nearly impossible to fall under the CTT with only attacking as the CH.

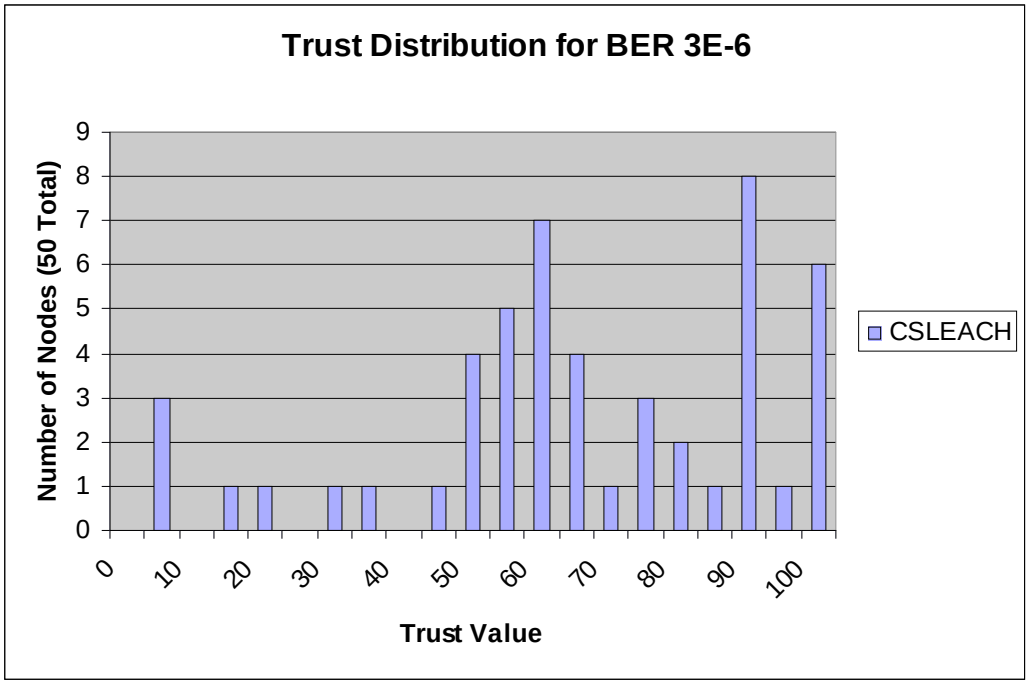

Figure 23 Trust distribution for BER 3E-6.

The simulations uses the settings 60 for CTT and 30 for MTT. The CH punishment is set to 20 while the CM punishment is set to 15. Figure 11 represents a network with BER of 3E-6. The network still functions because majority of the nodes have trust above the CTT. As errors increase, figure 11 shows the majority of nodes dip below the CTT. Nodes begin dropping because too few CHs are elected causing clusters to overfill.

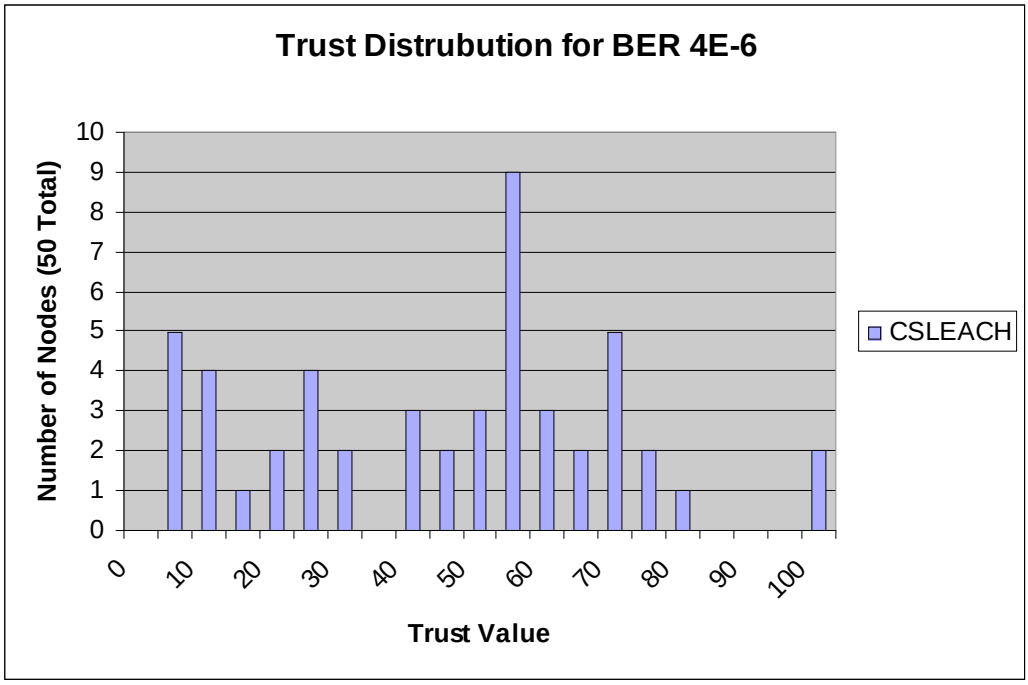

Figure 24 Trust distribution for BER 4E-6.

Under these conditions there are a few remedies. The trust punishment values for CH and CM can be reduced. Another approach is to adjust the CTT to 50 so majority of nodes qualify as CH. There is however another underlying issue which is the pool of possible CH is much smaller. The CH election percentage now only represents a fraction of the entire set of nodes. For future modifications, the election percentage should be based on an adjustable CH election percentage broadcast by the gateway with the round start message.

Chapter 5 : Moving Forward

CSLEACH accomplishes its goals to conserve energy and provide security, however there are many areas left for improvement.

5.1 LEACH Enhancements

The most difficult problem when optimizing the performance of a LEACH based protocol is dealing with the random election process. The random election process elects random numbers of CHs each round. Rounds with few CHs result in larger cluster sizes which means more time required each stage to disseminate timeslot schedules, and transmission timeslots. The maximum time allotted each stage must be buffered with enough time to account for larger clusters. As a result, time is wasted when cluster sizes are small. A deterministic election processes would reduce variability in CH election and allow for better optimization of maximum round durations **[12]**. It would also be beneficial to include the gateway in the CH election process so that the gateway could incorporate trust information to select CHs. Incorporating the gateway could eliminate the need for a blacklist stage in CSLEACH.

LEACH creates a unique traffic pattern when forming clusters. At the beginning of each round, every Non-CH must select a CH. This creates a spike in traffic during the beginning of stages where bandwidth is shared amongst large groups of nodes. This is especially noticeable during simulation because the simulator spends a majority of its time calculating collisions and backoffs. Random sleep durations were assigned before each round to improve the runtime of the simulator. In a real WSN, the increased volume

of collisions could cause increased corrupted packets at the start of a stage and implementing a random sleep strategy could be beneficial.

5.2 Security Enhancements

After Thoughts on Attacks

One of the major flaws in CSLEACH is the lack of broadcast authentication once a node is compromised. A compromised node has all the information necessary to forge a start message. The gateway private key is the critical component preventing an external attacker from sending a forged round start message. Compromised attackers have access to all the necessary components to spoof a start message. A single compromised node can perform a DOS (Denial of Service) attack against CSLEACH by attacking the synchronization of the network. This weakness does point out the need for asynchronous broadcast authentication. Possibilities include using Lamport's and Merkle's one-time signatures **[24]**.

To reach the true performance potentials of each algorithm, an efficient data aggregation or compression algorithm is necessary. Since CSLEACH requires a lossless aggregation schemes, the true gauge of how well LEACH, TLEACH, and CSLEACH may depend on the aggregation algorithms allowed.

Chapter 6 : Conclusion

WSN of the future will be energy efficient and secure. Developing such a protocol is about tradeoffs. Often to fix a security risk, the protocol sacrifices its energy efficiency. To make sensor nodes cheaper, we sacrifice security. CSLEACH attempts to interlace security and energy efficient methodologies into a single protocol. CSLEACH is more energy efficient, requires less memory per node than TLEACH and adapts a strategy for evaluating trust independent of application data knowledge. CSLEACH relies on the gateways superior resources to manage key distribution and trust management. By increasing packet sizes, increasing transmission turns, and using energy efficient block ciphers, SCLEACH can reduce the overhead from encryption and key distribution. As a result of research, we have shown how difficult LEACH is to protect. While SCLEACH has much to improve upon, it is a small step towards a necessary goal.

References

[1] Abdul, D.S., Kader, H.M.A. , Hadhoud, M.M., *Energy Efficiency of Encryption Schemes for Wireless Devices*, International Journal of Computer Theory and Engineering, Vol. 1, No. 3, August, 2009.

[2] Abuhelaleh, M.A.; Mismar, T.M.; Abuzneid, A.A.; , "Armor-LEACH - Energy Efficient, Secure Wireless Networks Communication," *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on* ,

[3] Aydos, M., Yantk, T., and Koç, Ç. K. 2000. A high-speed ECC-based wireless authentication on an ARM microprocessor. In *Proceedings of the 16th Annual Computer Security Applications Conference* (December 11 - 15, 2000). ACSAC. IEEE Computer Society, Washington, DC, 401.

[4] Boyle, D. , Newe, T. 2007. Security Protocols for Use with Wireless Sensor Networks: A Survey of Security Architectures. In *Proceedings of the Third international Conference on Wireless and Mobile Communications* (March 04 - 09, 2007). ICWMC. IEEE Computer Society, Washington, DC, 54.

[5] Chan, H. & Perrig, A. PIKE: peer intermediaries for key establishment in sensor networks *INFOCOM, IEEE*, **2005**, 524-535

[6] Culpepper, B. J., Dung, L., and Moh, M. 2004. Design and analysis of Hybrid Indirect Transmissions (HIT) for data gathering in wireless micro sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.* 8, 1 (Jan. 2004), 61-83.

[7] Di Pietro, R., Mancini, L. V., and Mei, A. 2003. Random key-assignment for secure Wireless Sensor Networks. In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks* (Fairfax, Virginia). SASN '03. ACM, New York, NY, 62-71.

[8] Du, J., Peng, S. , "Choice of Secure Routing Protocol for Applications in Wireless Sensor Networks," *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on* , vol.2, no., pp.470-473, 18-20 Nov. 2009

[9] Gaubatz, G., Kaps, J., Ozturk, E., and Sunar, B. 2005. State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks. In *Proceedings of the Third IEEE international Conference on Pervasive Computing and Communications Workshops* (March 08 - 12, 2005). PERCOMW. IEEE Computer Society, Washington, DC, 146-150.

[10] Großschädl, J., Tillich, S., Rechberger, C., Hofmann, M., and Medwed, M. 2007. Energy evaluation of software implementations of block ciphers under memory constraints. In *Proceedings of the Conference on Design, Automation and Test in Europe* (Nice, France, April 16 - 20, 2007). Design, Automation, and Test in Europe. EDA Consortium, San Jose, CA, 1110-1115.

[11] He, Q., Wu, D., Khosla, P.; , "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE* , March 2004

[12] Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H.; , "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on* , vol.1, no.4, pp. 660- 670, Oct 2002

[13] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. 2000. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33rd Hawaii international Conference on System Sciences-Volume 8 - Volume 8* (January 04 - 07, 2000). HICSS. IEEE Computer Society, Washington, DC, 8020.

[14] Heinzelman, W. R., Kulik, J., Balakrishnan, H. 1999. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE international Conference on Mobile Computing and Networking* (Seattle, Washington, United States, August 15 - 19, 1999). MobiCom '99. ACM, New York, NY, 174-185.

[15] Hu, J., Burmester, M. 2006. LARS: a locally aware reputation system for mobile ad hoc networks. In *Proceedings of the 44th Annual Southeast Regional Conference*

[16] Jiangtao Wang; Geng Yang; Shengshou Chen; Yanfei Sun; , "Secure LEACH routing protocol based on low-power cluster-head selection algorithm for wireless sensor networks," *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on* , vol., no., pp.341-344, Nov. 28 2007-Dec. 1 2007

[17] Jinwala, D.C., Patel, D.R., SkDasgupta, K., Investigating and Analyzing the Light-weight ciphers for Wireless Sensor Networks
2009,

[18] Karlof, C., Sastry, N., Wagner, D., "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 162-175.

[19] Karlof, C., Wagner, D., Secure, *Routing in Wireless Sensor Networks: Attacks and Countermeasures* , First IEEE International Workshop on Sensor Network Protocols and Applications, 2002,

[20] Law, Y. W., Doumen, J., and Hartel, P. 2006. Survey and benchmark of block ciphers for wireless sensor networks. *ACM Trans. Sen. Netw.* 2, 1 (Feb. 2006), 65-93.

[21] Liu, D., Ning, P., Zhu, S., and Jajodia, S. 2005. Practical Broadcast Authentication in Sensor Networks. In *Proceedings of the the Second Annual international Conference on Mobile and Ubiquitous Systems: Networking and Services* (July 17 - 21, 2005). International Conference on Mobile and Ubiquitous Systems: Networking and Services. IEEE Computer Society, Washington, DC, 118-132.

[22] Luk, M., Perrig, A., Whillock, B. 2006. Seven cardinal properties of sensor network broadcast authentication. In *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks* (Alexandria, Virginia, USA, October 30 - 30, 2006). SASN '06. ACM, New York, NY, 147-156.

[23] Marti, S., Giuli, T. J., Lai, K., and Baker, M. 2000. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual international Conference on Mobile Computing and Networking*

[24] Merkle, R. C. 1989. A certified digital signature. In *Proceedings on Advances in Cryptology* (Santa Barbara, California, United States). G. Brassard, Ed. Springer-Verlag New York, New York, NY, 218-238.

[25] Moore, T., Clulow, J., Secure Path-Key Revocation in Sensor Networks,
Available [Online] http://people.seas.harvard.edu/~tmoore/ifipsec-pres.pdf

[26] Oliveira, L.B.; Wong, H.C.; Bern, M.; Dahab, R.; Loureiro, A.A.F.; , "SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks," *Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on* , July 2006

[27] Perrig, A., Stankovic, J., and Wagner, D. 2004. Security in wireless sensor networks. *Commun. ACM* 47, 6 (Jun. 2004), 53-57.

[28] Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E. 2002. SPINS: security protocols for sensor networks. *Wirel. Netw.* 8, 5 (Sep. 2002), 521-534.

[29] Raymond, D.R., Midkiff, S.F., "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses," *Pervasive Computing, IEEE* , 2008

[30] Sami, S., Al-Wakeel , S., Al-Swailem, S.A., PRSA: A Path Redundancy Based Security Algorithm for Wireless Sensor Networks, IEE WNC 2007 Proceedings, 2007

[31] Song, F. and Zhao, B. 2008. Trust-Based LEACH Protocol for Wireless Sensor Networks. In *Proceedings of the 2008 Second international Conference on Future Generation Communication and Networking - Volume 01* (December 13 - 15, 2008). FGCN. IEEE Computer Society, Washington, DC, 202-207.

[32] Wander, A.S.; Gura, N.; Eberle, H.; Gupta, V.; Shantz, S.C.; , "Energy analysis of public-key cryptography for wireless sensor networks," *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on* , vol., no., pp. 324- 328, 8-12 March 2005

[33] Wang, H.D., Sheng, B.; Tan, C.C.; Li, Q.; , "Comparing Symmetric-key and Public-key Based Security Schemes in Sensor Networks: A Case Study of User Access Control," *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on* , June 2008

[34] Xiao, D., Wei, M., Zhou, Y. , "Secure-SPIN: Secure Sensor Protocol for Information via Negotiation for Wireless Sensor Networks," *Industrial Electronics and Applications, 2006 1ST IEEE Conference on* , vol., no., pp.1-4, 24-26 May 2006

[35] Available [Online] http://en.wikipedia.org/wiki/XTEA